YTM32B1HA0x Reference Manual

Support: YTM32B1HA01G0MLUT, YTM32B1HA01G0MLQT, YTM32B1HA01G0MLLT

Yuntu Microelectronics

Document Number: YTM32B1HA0x RM

Rev.1.3, 2026/4/21

YUNTU reserves the right to change the production detail specifications as may be required to permit improvements in the design of its products.

Contents

I Introduction

1 Introduction

1.1 Overview . . . 2
1.2 Features Summary . . . 2
1.3 Block Diagram . . . 4
1.4 Glossary . . . 4

II Memory Map and Memories

2 Memory Map

2.1 Introduction . . . 7
2.2 System Memory Map . . . 7
2.3 Peripheral Bus (APB) Memory Map . . . 7
2.4 Glossary . . . 10

3 Embedded FLASH Module (EFM)

3.1 Introduction . . . 12
3.1.1 EFM Memory Map . . . 12
3.1.2 Features . . . 13
3.1.3 Modes of Operation . . . 13
3.2 Register Definition . . . 13
3.2.1 EFM Memory Map . . . 13
3.2.1.1 EFM CTRL Register . . . 14
3.2.1.2 EFM STS Register . . . 17
3.2.1.3 EFM CMD Register . . . 19
3.2.1.4 EFM CMD_PROT Register . . . 20
3.2.1.5 EFM TIMING1 Register . . . 21
3.2.1.6 EFM TIMING2 Register . . . 22
3.2.1.7 EFM ECCERR_INJECT_MSK_0 Register . . . 23
3.2.1.8 EFM ECCERR_INJECT_MSK_1 Register . . . 24
3.2.1.9 EFM ECCERR_INJECT_ADDR Register . . . 25
3.2.1.10 EFM ADDR_PROT Register . . . 26
3.2.1.11 EFM ECC_ERR_ADDR Register . . . 27
3.2.1.12 EFM CMD_UNLOCK Register . . . 28
3.2.1.13 EFM CUS_KEY Register . . . 29
3.2.1.14 EFM CTRL_OVRD Register . . . 30

Yuntu Microelectronics

3.3 Functional Description . . . 31
3.3.1 EFM Initial . . . 31
3.3.2 Setting Timing Control Register . . . 31
3.3.3 Command Operation Flow . . . 31
3.3.4 Read While Write . . . 33
3.3.5 Boot Swap Operation . . . 33
3.3.6 Disable JTAG and SWD . . . 33
3.3.7 Address Protection . . . 33
3.3.8 Manual Erase Retry Operation . . . 34
3.3.9 Flash Abnormal Operation . . . 34
3.4 Glossary . . . 35

4 Read-Only Memory (ROM)
4.1 Introduction . . . 36
4.1.1 Features . . . 36
4.2 ROM Boot Code Functions and Flowchart . . . 37
4.2.1 Use the CUS_KEY As Root of Trust . . . 37
4.2.2 The ROM Boot Work Flowchart . . . 37
4.3 Functional Descriptions . . . 39
4.3.1 Structure of BVT (Boot Vector Table) . . . 39
4.3.2 Location of BVT . . . 39
4.3.3 Boot Vector Table . . . 39
4.3.4 Boot Configuration Word Description . . . 39
4.3.5 Secure Boot . . . 41
4.3.5.1 Secure Boot Mode . . . 41
4.3.5.2 Secure Boot Configuration . . . 41
4.3.5.2.1 Secure Boot Config Group Structure . . . 43
4.3.5.2.2 Secure Boot Section Configuration Structure . . . 44
4.3.6 Fast Wakeup from Powerdown Mode . . . 44
4.4 Glossary . . . 44

5 On-Chip RAM (OCRAM)
5.1 Chip-specific OCRAM information . . . 46
5.1.1 Instantiation Information . . . 46
5.2 Introduction . . . 46
5.2.1 Overview . . . 46
5.2.2 Features . . . 46
5.2.3 Block diagram . . . 47
5.3 Functional Description . . . 47
5.3.1 RAM Initialization . . . 47
5.3.2 Reads . . . 47
5.3.3 Writes . . . 47
5.3.3.1 64 Bits Write . . . 47
5.3.3.2 Less than 64-bit writes . . . 47
5.3.3.3 Unaligned writes . . . 48

6 Register File (REGFILE)
6.1 Introduction . . . 49
6.1.1 Features . . . 49

Yuntu Microelectronics

6.2 Register Definition . . . 49
6.2.1 REGFILE Memory Map . . . 49
6.2.1.1 REGFILE DR Register . . . 49

III Signal Multiplexing and Pin Assignment

7 Pinmux
7.1 Pinmux Table . . . 52
7.2 Glossary . . . 57

8 General Purpose Input/Output (GPIO)
8.1 Introduction . . . 58
8.1.1 Features . . . 58
8.1.2 Modes of Operation . . . 58
8.1.2.1 Active Mode . . . 58
8.1.2.2 Sleep Mode . . . 58
8.1.2.3 Deepsleep Mode . . . 58
8.1.2.4 Standby Mode . . . 58
8.1.2.5 Powerdown Mode . . . 59
8.2 Register Definition . . . 59
8.2.1 GPIO Memory Map . . . 59
8.2.1.1 GPIO PDOR Register . . . 60
8.2.1.2 GPIO PSOR Register . . . 60
8.2.1.3 GPIO PCOR Register . . . 61
8.2.1.4 GPIO PTOR Register . . . 62
8.2.1.5 GPIO PDIR Register . . . 62
8.2.1.6 GPIO POER Register . . . 63
8.2.1.7 GPIO PIER Register . . . 63
8.2.1.8 GPIO PIFR Register . . . 64
8.2.1.9 GPIO PCR Register . . . 64
8.3 Functional description . . . 65
8.3.1 General-purpose input . . . 65
8.3.2 General-purpose output . . . 65
8.3.3 Port interrupt and DMA request . . . 65

9 Port Control (PCTRL)
9.1 Chip-specific Information . . . 66
9.1.1 Pin Property . . . 66
9.2 Introduction . . . 68
9.2.1 Overview . . . 68
9.2.2 Features . . . 68
9.2.3 Modes of Operation . . . 69
9.2.3.1 Active Mode . . . 69
9.2.3.2 Sleep Mode . . . 69
9.2.3.3 Deepsleep Mode . . . 69
9.2.3.4 Standby Mode . . . 69
9.2.3.5 Powerdown Mode . . . 69
9.3 Register Definition . . . 69
9.3.1 PCTRL Memory Map . . . 69
9.3.1.1 PCTRL PCR Register . . . 70

Yuntu Microelectronics

9.4 Functional description . . . 72
9.4.1 Port Control . . . 72

IV Core and System Modules

10 CM7 Overview
10.1 ARM Cortex-M7 Introduction . . . 74
10.1.1 Features . . . 74
10.1.2 Buses, Interconnects, and Interfaces . . . 75
10.1.3 Core Configuration . . . 75
10.2 Nested Vectored Interrupt Controller (NVIC) . . . 75
10.3 Wakeup Interrupt Controller (WIC) . . . 81
10.4 FPU Configuration . . . 82
10.5 JTAG Controller Configuration . . . 82
10.6 Glossary . . . 82

11 Direct Memory Access (DMA)
11.1 Chip Specific Information . . . 83
11.1.1 DMA Channel MUX table . . . 83
11.1.2 Power Mode Description . . . 86
11.2 Introduction . . . 86
11.2.1 Features . . . 86
11.2.2 DMA Channel Transfer Structure (CTS) . . . 87
11.2.3 DMA Trigger MUX . . . 88
11.3 Register Definition . . . 88
11.3.1 DMA Memory Map . . . 88
11.3.1.1 DMA CTRL Register . . . 102
11.3.1.2 DMA CHTLDIF Register . . . 104
11.3.1.3 DMA CHTLHDIF Register . . . 104
11.3.1.4 DMA CHEIF Register . . . 105
11.3.1.5 DMA CHEIE Register . . . 106
11.3.1.6 DMA ERS Register . . . 106
11.3.1.7 DMA REQEN Register . . . 108
11.3.1.8 DMA REQS Register . . . 109
11.3.1.9 DMA START Register . . . 109
11.3.1.10 DMA DONE Register . . . 109
11.3.1.11 DMA CHMUX Register . . . 110
11.3.1.12 DMA CTS_SADDR Register . . . 110
11.3.1.13 DMA CTS_SOFF Register . . . 111
11.3.1.14 DMA CTS_TCR Register . . . 111
11.3.1.15 DMA CTS_BCNT_BCNT Register . . . 112
11.3.1.16 DMA CTS_BCNT_LODIS Register . . . 113
11.3.1.17 DMA CTS_BCNT_LOEN Register . . . 113
11.3.1.18 DMA CTS_STO Register . . . 114
11.3.1.19 DMA CTS_DADDR Register . . . 115
11.3.1.20 DMA CTS_DOFF Register . . . 115
11.3.1.21 DMA CTS_TCNT_LKDIS Register . . . 116
11.3.1.22 DMA CTS_TCNT_LKEN Register . . . 116
11.3.1.23 DMA CTS_DTO_RLD_DTO Register . . . 117
11.3.1.24 DMA CTS_DTO_RLD_RLD Register . . . 118

Yuntu Microelectronics

11.3.1.25 DMA CTS_CSR Register . . . 118
11.3.1.26 DMA CTS_TCNTRV Register . . . 120
11.4 Functional Description . . . 120
11.4.1 Transfer Loop . . . 120
11.4.2 Trigger Loop . . . 121
11.4.3 Channel Arbitration . . . 121
11.4.4 DMA Channel Initialize . . . 121
11.5 Glossary . . . 121

12 Chip Integration Module (CIM)
12.1 Introduction . . . 122
12.1.1 Features . . . 122
12.2 Register Definition . . . 122
12.2.1 CIM Memory Map . . . 122
12.2.1.1 CIM CTRL0 Register . . . 123
12.2.1.2 CIM CTRL1 Register . . . 124
12.2.1.3 CIM FPUIE Register . . . 125
12.2.1.4 CIM SWTRIG Register . . . 126
12.2.1.5 CIM ETMROPT0 Register . . . 127
12.2.1.6 CIM ETMROPT1 Register . . . 129
12.2.1.7 CIM ETMROPT2 Register . . . 131
12.2.1.8 CIM ETMROPT3 Register . . . 133
12.2.1.9 CIM ETMROPT4 Register . . . 135
12.2.1.10 CIM ETMROPT5 Register . . . 136
12.2.1.11 CIM ETMR_GLB_OPT Register . . . 138
12.2.1.12 CIM SWFAULT Register . . . 139
12.2.1.13 CIM DCCM_EI Register . . . 140
12.2.1.14 CIM UID0 Register . . . 141
12.2.1.15 CIM UID1 Register . . . 141
12.2.1.16 CIM UID2 Register . . . 142
12.2.1.17 CIM UID3 Register . . . 143
12.2.1.18 CIM DIEINFO Register . . . 143
12.2.1.19 CIM CHIPINFO0 Register . . . 145
12.2.1.20 CIM CHIPINFO1 Register . . . 147
12.2.1.21 CIM CPUINFO Register . . . 148
12.2.1.22 CIM BOOT_CFG Register . . . 150

13 Trigger Multiplexer Module (TMU)
13.1 Introduction . . . 153
13.2 Trigger Interconnectivity Table . . . 153
13.3 Register Definition . . . 156
13.3.1 TMU Memory Map . . . 156
13.3.1.1 TMU MUX Register . . . 157
13.4 Glossary . . . 158

V Clock, Reset and Power Management

14 System Clock Unit (SCU)
14.1 Introduction . . . 160
14.1.1 Features . . . 160

Yuntu Microelectronics

14.1.2 Block Diagram . . . 161
14.2 Register Definition . . . 161
14.2.1 SCU Memory Map . . . 161
14.2.1.1 SCU CLKS Register . . . 162
14.2.1.2 SCU DIV Register . . . 163
14.2.1.3 SCU STS Register . . . 165
14.2.1.4 SCU DIVSTS Register . . . 166
14.2.1.5 SCU FIRC_CTRL Register . . . 168
14.2.1.6 SCU SIRC_CTRL Register . . . 169
14.2.1.7 SCU FXOSC_CTRL Register . . . 170
14.2.1.8 SCU SXOSC_CTRL Register . . . 172
14.2.1.9 SCU PLL_CTRL Register . . . 174
14.2.1.10 SCU PLL_CFG Register . . . 175
14.2.1.11 SCU PLL_FM Register . . . 176
14.2.1.12 SCU PLL_FD Register . . . 176
14.2.1.13 SCU CLKO Register . . . 177
14.2.1.14 SCU CMU_CTRL Register . . . 178
14.2.1.15 SCU CMUSTS Register . . . 180
14.2.1.16 SCU CMUCMP_HIGH Register . . . 182
14.2.1.17 SCU CMUCMP_LOW Register . . . 183
14.3 Functional Description . . . 184
14.3.1 Clock Switcher . . . 184
14.3.2 Phase-Locked Loop Clock(PLL) . . . 184
14.3.2.1 Integer Mode . . . 185
14.3.2.2 Fractional Mode . . . 185
14.3.2.3 Spread Spectrum Mode . . . 185
14.3.3 System Clock Divider . . . 185
14.3.4 Clock Monitor Unit (CMU) . . . 186
14.3.5 Clocks After Reset . . . 186
14.3.6 Operation Modes . . . 186
14.4 Glossary . . . 186

15 IP Control (IPC)
15.1 Introduction . . . 187
15.2 Features . . . 187
15.3 Block diagram . . . 187
15.4 Peripheral Clock Table . . . 188
15.5 Register Definition . . . 191
15.5.1 IPC Memory Map . . . 191
15.5.1.1 IPC CTRL Register . . . 194
15.6 Functional Description . . . 195

16 Reset Control Unit (RCU)
16.1 Introduction . . . 196
16.1.1 Features . . . 196
16.2 Register Definition . . . 196
16.2.1 RCU Memory Map . . . 196
16.2.1.1 RCU RSSR Register . . . 197
16.2.1.2 RCU RPFR Register . . . 198

Yuntu Microelectronics

16.3 Functional Description . . . 200
16.3.1 External Pin Reset . . . 200
16.3.2 POR and LVD . . . 200
16.3.3 High Voltage Detect Reset . . . 200
16.3.4 Clock Monitor Reset . . . 200
16.3.5 Watchdog Reset . . . 200
16.3.6 Low Power Mode Entry Acknowledge Timeout Reset . . . 200
16.3.7 Software Reset . . . 200
16.3.8 Core Lockup Reset . . . 200
16.3.9 Debug Reset . . . 200
16.3.10 FMU Functional Reset . . . 201
16.3.11 FMU Fail React Reset . . . 201

17 Power Control Unit (PCU)
17.1 Introduction . . . 202
17.1.1 Block Diagram . . . 202
17.1.2 Features . . . 204
17.2 Register Definition . . . 204
17.2.1 PCU Memory Map . . . 204
17.2.1.1 PCU SSTS Register . . . 204
17.2.1.2 PCU STS Register . . . 205
17.2.1.3 PCU INTE Register . . . 206
17.2.1.4 PCU CTRL Register . . . 207
17.3 Functional Description . . . 210
17.3.1 Reset . . . 210
17.3.2 Interrupt . . . 211
17.4 Glossary . . . 211

18 Power Mode
18.1 Introduction . . . 212
18.1.1 Active Mode . . . 212
18.1.2 Sleep Mode . . . 212
18.1.3 Deepsleep Mode . . . 212
18.1.4 Standby Mode . . . 212
18.1.5 Powerdown Mode . . . 212
18.2 Power Modes Table . . . 213
18.3 Wake-up Sources . . . 214

19 Wake-up Unit (WKU)
19.1 Introduction . . . 219
19.1.1 Features . . . 219
19.1.2 Wake-up pins introduction . . . 219
19.2 Register Definition . . . 221
19.2.1 WKU Memory Map . . . 221
19.2.1.1 WKU PCR Register . . . 222
19.2.1.2 WKU PFR Register . . . 223
19.2.1.3 WKU RPCR Register . . . 224
19.2.1.4 WKU MER Register . . . 225

Yuntu Microelectronics

VI Communication Interfaces

20 Flexible Controller Area Network (FlexCAN)

20.1 Chip-specific FlexCAN Information . . . 228
20.1.1 Instantiation Information . . . 228
20.2 Introduction . . . 228
20.2.1 Block Diagram . . . 229
20.2.2 Overview . . . 230
20.2.3 Features . . . 230
20.2.4 Modes of Operation . . . 232
20.3 Register Definition . . . 233
20.3.1 Register Access and Reset Information . . . 234
20.3.2 Memory Mapping . . . 237
20.3.3 CAN Memory Map . . . 237
20.3.3.1 CAN MCR Register . . . 263
20.3.3.2 CAN CTRL1 Register . . . 270
20.3.3.3 CAN TIMER Register . . . 274
20.3.3.4 CAN RXMGMASK Register . . . 275
20.3.3.5 CAN RX14MASK Register . . . 277
20.3.3.6 CAN RX15MASK Register . . . 278
20.3.3.7 CAN ECR Register . . . 279
20.3.3.8 CAN ESR1 Register . . . 281
20.3.3.9 CAN IMASK2 Register . . . 288
20.3.3.10 CAN IMASK1 Register . . . 289
20.3.3.11 CAN IFLAG2 Register . . . 290
20.3.3.12 CAN IFLAG1 Register . . . 290
20.3.3.13 CAN CTRL2 Register . . . 294
20.3.3.14 CAN ESR2 Register . . . 299
20.3.3.15 CAN CRCR Register . . . 301
20.3.3.16 CAN RXFGMASK Register . . . 302
20.3.3.17 CAN RXFIR Register . . . 304
20.3.3.18 CAN CBT Register . . . 304
20.3.3.19 CAN IMASK4 Register . . . 306
20.3.3.20 CAN IMASK3 Register . . . 307
20.3.3.21 CAN IFLAG4 Register . . . 308
20.3.3.22 CAN IFLAG3 Register . . . 309
20.3.3.23 CAN RAM Register . . . 310
20.3.3.24 CAN RXIMR Register . . . 311
20.3.3.25 CAN MECR Register . . . 312
20.3.3.26 CAN ERRIAR Register . . . 315
20.3.3.27 CAN ERRIDPR Register . . . 316
20.3.3.28 CAN ERRIPPR Register . . . 317
20.3.3.29 CAN RERRAR Register . . . 318
20.3.3.30 CAN RERRDR Register . . . 319
20.3.3.31 CAN RERRSYNR Register . . . 320
20.3.3.32 CAN ERRSR Register . . . 322
20.3.3.33 CAN CTRL1_PN Register . . . 323
20.3.3.34 CAN CTRL2_PN Register . . . 325
20.3.3.35 CAN WU_MTC Register . . . 326
20.3.3.36 CAN FLT_ID1 Register . . . 327

Yuntu Microelectronics

20.3.3.37 CAN FLT_DLC Register . . . 328
20.3.3.38 CAN PL1_LO Register . . . 329
20.3.3.39 CAN PL1_HI Register . . . 330
20.3.3.40 CAN FLT_ID2_IDMASK Register . . . 332
20.3.3.41 CAN PL2_PLMASK_LO Register . . . 333
20.3.3.42 CAN PL2_PLMASK_HI Register . . . 334
20.3.3.43 CAN WMB_WMBn_CS Register . . . 335
20.3.3.44 CAN WMB_WMBn_ID Register . . . 337
20.3.3.45 CAN WMB_WMBn_D03 Register . . . 337
20.3.3.46 CAN WMB_WMBn_D47 Register . . . 338
20.3.3.47 CAN EPRS Register . . . 340
20.3.3.48 CAN ENCBT Register . . . 341
20.3.3.49 CAN EDCBT Register . . . 342
20.3.3.50 CAN ETDC Register . . . 344
20.3.3.51 CAN FDCTRL Register . . . 345
20.3.3.52 CAN FDCBT Register . . . 349
20.3.3.53 CAN FDCRC Register . . . 351
20.3.3.54 CAN ERFCR Register . . . 352
20.3.3.55 CAN ERFIER Register . . . 357
20.3.3.56 CAN ERFSR Register . . . 358
20.3.3.57 CAN HR_TIME_STAMPn Register . . . 360
20.3.3.58 CAN ERFFELn Register . . . 361
20.3.4 Message Buffer Structure . . . 362
20.3.5 FlexCAN Memory Partition for CAN FD . . . 368
20.3.6 FlexCAN Message Buffer Memory Map . . . 370
20.3.7 Legacy Rx FIFO Structure . . . 376
20.3.8 Enhanced Rx FIFO Structure . . . 379
20.4 Functional Description . . . 381
20.4.1 Transmit Process . . . 382
20.4.2 Arbitration Process . . . 383
20.4.2.1 Lowest-Number Mailbox First . . . 383
20.4.2.2 Highest-Priority Mailbox First . . . 384
20.4.2.2.1 Local Priority Disabled . . . 384
20.4.2.2.2 Local Priority Enabled . . . 384
20.4.2.3 Arbitration Completion . . . 385
20.4.2.4 Arbitration Start and Stop Conditions . . . 385
20.4.3 Receive Process . . . 386
20.4.3.1 Receiving CAN Frames through Enhanced Rx FIFO . . . 387
20.4.3.1.1 CPU Read Triggered by Data Available . . . 387
20.4.3.1.2 CPU Read Triggered by Watermark Indication . . . 388
20.4.3.1.3 Using DMA to Service the Enhanced Rx FIFO . . . 389
20.4.4 Matching Process . . . 390
20.4.5 Receive Process under Pretended Networking Mode . . . 394
20.4.6 Move Process . . . 397
20.4.6.1 Move-in . . . 397
20.4.6.2 Move-out . . . 398
20.4.7 Data Coherence . . . 398
20.4.7.1 Transmission Abort Mechanism . . . 398
20.4.7.2 Mailbox Inactivation . . . 399
20.4.7.3 Mailbox Lock Mechanism . . . 400

Yuntu Microelectronics

20.4.8 Enhanced Rx FIFO . . . 400
20.4.8.1 Enhanced Rx FIFO Filter Elements . . . 401
20.4.8.2 Enhanced Rx FIFO Matching Process . . . 402
20.4.8.3 Enhanced Rx FIFO Under DMA Operation . . . 405
20.4.8.4 Enhanced Rx FIFO Clear Operation . . . 405
20.4.9 Legacy Rx FIFO . . . 406
20.4.9.1 Rx FIFO Under DMA Operation . . . 407
20.4.9.2 Clear Legacy Rx FIFO Operation . . . 407
20.4.10 DMA Operation . . . 407
20.4.11 CAN Protocol Related Features . . . 408
20.4.11.1 CAN FD Frames . . . 408
20.4.11.2 Transceiver Delay Compensation . . . 413
20.4.11.3 Remote Frame Transmission . . . 416
20.4.11.4 Remote Frame Reception . . . 416
20.4.11.5 Overload Frames . . . 417
20.4.11.6 Message Buffer Time Stamp . . . 417
20.4.11.7 High-Resolution Time Stamp . . . 417
20.4.11.8 Protocol Timing . . . 418
20.4.11.9 Arbitration and Matching Timing . . . 423
20.4.11.10 Tx Arbitration Start Delay . . . 424
20.4.12 Clock Domains and Restrictions . . . 427
20.4.13 Modes of Operation Details . . . 431
20.4.13.1 Freeze Mode . . . 431
20.4.13.2 Module Disable Mode . . . 432
20.4.13.3 Doze Mode . . . 432
20.4.13.4 Deepsleep Mode . . . 433
20.4.13.5 Pretended Networking Mode . . . 435
20.4.14 Interrupts . . . 435
20.4.15 Bus Interface . . . 436
20.4.16 Detection and Correction of Memory Errors . . . 437
20.4.16.1 Sources of Memory Access . . . 437
20.4.16.2 Error Indication . . . 438
20.4.16.3 Error Reporting . . . 438
20.4.16.4 Response to Errors . . . 439
20.4.16.5 Error Injection . . . 440
20.5 Initialization / Application Information . . . 440
20.5.1 RAM Initialization . . . 440
20.5.2 FlexCAN Initialization Sequence . . . 441
20.6 FlexCAN Ram Usage and Initialization . . . 441
20.6.1 RAM Size . . . 442
20.6.2 RAM Usage . . . 442
20.6.3 RAM Initialization Addresses . . . 449
20.6.4 RAM Addresses for Error Injection and Reporting . . . 451
20.6.4.1 RAM Addresses for FlexCAN Standard Features . . . 451
20.6.4.2 RAM Addresses for FlexCAN Advanced Features . . . 453
20.6.4.2.1 Enhanced MB Memory Region . . . 453
20.6.4.2.2 High-Resolution Time Stamp Region . . . 454
20.6.4.2.3 Enhanced Rx FIFO Region . . . 455
20.6.4.2.4 Enhanced Rx FIFO Filter Elements Region . . . 459

Yuntu Microelectronics

21 Local Interconnect Network (LINFlexD)
21.1 Chip-specific LINFlexD Information . . . 461
21.1.1 Instantiation Information . . . 461
21.1.2 Signal Description . . . 461
21.1.3 Power Mode Discription . . . 461
21.2 Introduction . . . 461
21.2.1 General . . . 461
21.2.2 Features . . . 462
21.3 Register Definition . . . 463
21.3.1 LINFlexD Memory Map . . . 463
21.3.1.1 LINFlexD LINCR1 Register . . . 465
21.3.1.2 LINFlexD LINIER Register . . . 468
21.3.1.3 LINFlexD LINSR Register . . . 470
21.3.1.4 LINFlexD LINESR Register . . . 473
21.3.1.5 LINFlexD UARTCR Register . . . 475
21.3.1.6 LINFlexD UARTSR Register . . . 481
21.3.1.7 LINFlexD LINTCSR Register . . . 484
21.3.1.8 LINFlexD LINOCR Register . . . 486
21.3.1.9 LINFlexD LINTOCR Register . . . 486
21.3.1.10 LINFlexD LINFBRR Register . . . 487
21.3.1.11 LINFlexD LINIBRR Register . . . 488
21.3.1.12 LINFlexD LINCFR Register . . . 489
21.3.1.13 LINFlexD LINCR2 Register . . . 490
21.3.1.14 LINFlexD BIDR Register . . . 492
21.3.1.15 LINFlexD DATA_DATA8 Register . . . 493
21.3.1.16 LINFlexD DATA_DATA16 Register . . . 494
21.3.1.17 LINFlexD DATA_DATA32 Register . . . 494
21.3.1.18 LINFlexD IFER Register . . . 495
21.3.1.19 LINFlexD IFMI Register . . . 496
21.3.1.20 LINFlexD IFMR Register . . . 497
21.3.1.21 LINFlexD IFCR Register . . . 498
21.3.1.22 LINFlexD GCR Register . . . 499
21.3.1.23 LINFlexD UARTPTO Register . . . 501
21.3.1.24 LINFlexD UARTCTO Register . . . 502
21.3.1.25 LINFlexD DMATXE Register . . . 503
21.3.1.26 LINFlexD DMARXE Register . . . 504
21.4 Functional Description . . . 505
21.4.1 LIN Protocol . . . 505
21.4.1.1 Frames . . . 505
21.4.1.2 Data Field . . . 506
21.4.1.3 Break . . . 506
21.4.1.4 Sync Byte . . . 507
21.4.1.5 Identifier . . . 507
21.4.1.6 Checksum . . . 507
21.4.2 LINFlexD Features . . . 507
21.4.2.1 Operating Modes . . . 507
21.4.2.2 Loop Back Mode . . . 509
21.4.2.3 Master Mode . . . 509
21.4.2.4 Slave Mode . . . 510
21.4.2.5 Errors . . . 511

Yuntu Microelectronics

21.4.2.6 Identifier Filtering . . . 515
21.4.2.7 Start Detection and Break Delimiter Detection in Receiver . . . 516
21.4.2.8 Baud Rate Generation . . . 520
21.4.2.9 Automatic Resynchronization . . . 521
21.4.2.10 Wakeup Management . . . 521
21.4.3 Timer . . . 521
21.4.4 UART Mode . . . 522
21.4.4.1 8-Bit Data Frames . . . 522
21.4.4.2 9-Bit Frames . . . 522
21.4.4.3 13-Bit Frames . . . 522
21.4.4.4 Buffer in UART Mode . . . 523
21.4.4.5 UART Transmitter . . . 523
21.4.4.6 UART Receiver . . . 524
21.4.5 Interrupts . . . 526
21.4.6 DMA Operations . . . 527
21.4.6.1 LIN Master Node: Transmit DMA Channel . . . 527
21.4.6.2 LIN Master Node: Receive DMA Channel . . . 528
21.4.6.3 LIN Slave Node: Transmit DMA Channel . . . 529
21.4.6.4 LIN Slave Node: Receive DMA Channel . . . 529
21.4.6.5 UART Mode: Transmit DMA Channel . . . 530
21.4.6.6 UART Mode: Receive DMA Channel . . . 530
21.4.6.7 DMA Usage Notes and Limitations . . . 530
21.4.7 LINFlexD Clock Tolerance . . . 531
21.4.7.1 Faster Receiver Tolerance . . . 531
21.4.7.2 Slower Receiver Tolerance . . . 531
21.5 Glossary . . . 532

22 Serial Port Interface (SPI)
22.1 Chip-specific SPI information . . . 533
22.1.1 Instantiation Information . . . 533
22.2 Introduction . . . 533
22.2.1 Features . . . 533
22.2.2 Signal Description . . . 533
22.3 Register Definition . . . 534
22.3.1 SPI Memory Map . . . 534
22.3.1.1 SPI CTRL Register . . . 534
22.3.1.2 SPI TXCFG Register . . . 537
22.3.1.3 SPI CLK Register . . . 539
22.3.1.4 SPI STS Register . . . 540
22.3.1.5 SPI INTE Register . . . 542
22.3.1.6 SPI MATCH0 Register . . . 543
22.3.1.7 SPI MATCH1 Register . . . 543
22.3.1.8 SPI TXFIFO Register . . . 544
22.3.1.9 SPI RXFIFO Register . . . 545
22.3.1.10 SPI DATA Register . . . 545
22.4 Functional description . . . 546
22.4.1 SPI Functional Clock . . . 546
22.4.2 Master Mode . . . 546
22.4.2.1 Tx and CMD FIFO . . . 546
22.4.2.2 Rx FIFO . . . 546

Yuntu Microelectronics

22.4.2.3 Data Match . . . 547
22.4.2.4 Pin Configuration . . . 547
22.4.2.5 Clock LoopBack . . . 547
22.4.2.6 Continuous Transfer . . . 547
22.4.2.6.1 Continuous Command . . . 547
22.4.3 Slave Mode . . . 547
22.4.3.1 Tx and CMD FIFO . . . 547
22.4.3.2 Receive FIFO . . . 548
22.4.3.3 Data Match . . . 548
22.4.3.4 Internal PCS . . . 548
22.4.3.5 Continuous Transfer . . . 548
22.4.3.5.1 Continuous Mode . . . 548
22.5 Glossary . . . 548

23 Quad Serial Peripheral Interface (QSPI)
23.1 Chip-specific Information . . . 549
23.1.1 Instantiation Information . . . 549
23.1.2 Signal Description . . . 549
23.1.3 Clock Description . . . 549
23.1.4 DMA Description . . . 549
23.1.5 Interrupt Description . . . 550
23.2 Introduction . . . 550
23.2.1 Overview . . . 550
23.2.2 Features . . . 550
23.2.3 Block Diagram . . . 550
23.3 Register Definition . . . 551
23.3.1 QSPI Memory Map . . . 552
23.3.1.1 QSPI MEN Register . . . 554
23.3.1.2 QSPI MCFG Register . . . 554
23.3.1.3 QSPI SMP Register . . . 557
23.3.1.4 QSPI ST Register . . . 559
23.3.1.5 QSPI IF Register . . . 562
23.3.1.6 QSPI IE Register . . . 564
23.3.1.7 QSPI SPNDST Register . . . 566
23.3.1.8 QSPI RBCS Register . . . 566
23.3.1.9 QSPI TBCS Register . . . 567
23.3.1.10 QSPI APBCFG Register . . . 568
23.3.1.11 QSPI AHBCFG Register . . . 569
23.3.1.12 QSPI BUF0CFG Register . . . 570
23.3.1.13 QSPI BUF1CFG Register . . . 571
23.3.1.14 QSPI BUF2CFG Register . . . 572
23.3.1.15 QSPI BUF3CFG Register . . . 573
23.3.1.16 QSPI BUF0IND Register . . . 574
23.3.1.17 QSPI BUF1IND Register . . . 575
23.3.1.18 QSPI BUF2IND Register . . . 576
23.3.1.19 QSPI FACFG Register . . . 577
23.3.1.20 QSPI SFA Register . . . 578
23.3.1.21 QSPI SFACFG Register . . . 579
23.3.1.22 QSPI SFATA Register . . . 580
23.3.1.23 QSPI SFA1TA Register . . . 581

Yuntu Microelectronics

23.3.1.24 QSPI SFBTA Register . . . 582
23.3.1.25 QSPI SFB1TA Register . . . 583
23.3.1.26 QSPI DTPR Register . . . 584
23.3.1.27 QSPI TXDAT Register . . . 585
23.3.1.28 QSPI RXDAT Register . . . 586
23.3.1.29 QSPI LUTKEY Register . . . 587
23.3.1.30 QSPI LOCK Register . . . 588
23.3.1.31 QSPI LUT Register . . . 589
23.4 Function . . . 591
23.4.1 Flash Connection . . . 591
23.4.2 Flash Access Mode . . . 592
23.4.2.1 Single/Dual/Quad/Octal Mode . . . 592
23.4.2.2 Individual Mode and Parallel Mode . . . 593
23.4.2.3 SDR Mode and DDR Mode . . . 594
23.4.2.4 HyperBus Mode . . . 595
23.4.3 Operation Mode . . . 595
23.4.3.1 Busy Mode . . . 595
23.4.3.2 Disable Mode . . . 596
23.4.3.3 Deepsleep Mode . . . 596
23.4.3.4 Debug Mode . . . 596
23.4.4 Flash Memory Map . . . 596
23.4.4.1 Addressing Mode . . . 597
23.4.5 Look Up Table . . . 597
23.4.5.1 LUT Lock . . . 598
23.4.5.2 Programmable Sequence Engine . . . 598
23.4.6 Device Accessing . . . 599
23.4.6.1 APB Command . . . 599
23.4.6.2 AHB Command . . . 599
23.4.7 AHB Buffers . . . 600
23.4.7.1 Prefetch . . . 600
23.4.7.1.1 Prefetch Abort Mechanism . . . 600
23.4.7.2 AHB Burst Support . . . 601
23.4.8 Flash Device Programming . . . 601
23.4.9 Flash Device Reading . . . 601
23.4.10 Error . . . 601
23.4.11 Input Sample Configuration . . . 602
23.4.11.1 Loopback Sample Mode . . . 602
23.4.11.2 Normal Sample Mode . . . 603
23.4.12 Data-test Mechanism . . . 603
23.4.13 Communication Example . . . 603
23.5 Glossary . . . 604

24 Inter-Integrated Circuit (I2C)
24.1 Chip-specific I2C Information . . . 605
24.1.1 Instantiation Information . . . 605
24.1.2 Signal Description . . . 605
24.1.3 Clock Description . . . 605
24.1.4 Reset Description . . . 605
24.1.5 Power Mode Description . . . 606

Yuntu Microelectronics

24.2 Introduction . . . 606
24.2.1 Features . . . 606
24.3 Register Definition . . . 606
24.3.1 I2C Memory Map . . . 606
24.3.1.1 I2C MCTRL Register . . . 607
24.3.1.2 I2C MCLKCFG Register . . . 609
24.3.1.3 I2C MFLTCFG Register . . . 609
24.3.1.4 I2C MSTS Register . . . 611
24.3.1.5 I2C MIE Register . . . 612
24.3.1.6 I2C TXFIFO Register . . . 614
24.3.1.7 I2C RXFIFO Register . . . 614
24.3.1.8 I2C MDATA Register . . . 615
24.3.1.9 I2C MATCH Register . . . 616
24.3.1.10 I2C TOCFG Register . . . 617
24.3.1.11 I2C HSCLK Register . . . 618
24.3.1.12 I2C SCTRL Register . . . 619
24.3.1.13 I2C SCLKCFG Register . . . 622
24.3.1.14 I2C SFLTCFG Register . . . 623
24.3.1.15 I2C SSTS Register . . . 624
24.3.1.16 I2C SIE Register . . . 627
24.3.1.17 I2C SDATA Register . . . 628
24.3.1.18 I2C SACK Register . . . 629
24.3.1.19 I2C SADDR Register . . . 629
24.3.1.20 I2C SRADDR Register . . . 630
24.4 Functional Description . . . 631
24.4.1 I2C Functional Clock . . . 631
24.4.2 Master Mode . . . 631
24.4.2.1 Tx and CMD FIFO . . . 631
24.4.2.2 Master Operations . . . 632
24.4.2.3 Rx FIFO and Data Matching . . . 632
24.4.3 Slave Mode . . . 632
24.4.3.1 Address Matching . . . 632
24.4.3.2 Data Processing . . . 633
24.4.3.3 Clock Stretching . . . 633
24.5 Glossary . . . 633

25 Single Edge Nibble Transmission (SENT)
25.1 Chip-specific SENT Information . . . 634
25.1.1 Instantiation Information . . . 634
25.1.2 Signal Description . . . 634
25.1.3 Clock Description . . . 634
25.1.4 DMA Description . . . 634
25.1.5 Interrupt Description . . . 635
25.2 Introduction . . . 635
25.2.1 Overview . . . 635
25.2.2 Features . . . 635
25.2.3 Block diagram . . . 636
25.3 Register definition . . . 636
25.3.1 SENT Memory Map . . . 637
25.3.1.1 SENT CTRL Register . . . 638

Yuntu Microelectronics

25.3.1.2 SENT CHCTRL Register . . . 639
25.3.1.3 SENT GSTS Register . . . 640
25.3.1.4 SENT GINTE Register . . . 642
25.3.1.5 SENT FRDY Register . . . 642
25.3.1.6 SENT SRDY Register . . . 643
25.3.1.7 SENT FINTE Register . . . 644
25.3.1.8 SENT SINTE Register . . . 645
25.3.1.9 SENT DMACTRL Register . . . 646
25.3.1.10 SENT CFG Register . . . 648
25.3.1.11 SENT NIBCFG Register . . . 649
25.3.1.12 SENT FMDATA Register . . . 650
25.3.1.13 SENT FMCHNUM Register . . . 652
25.3.1.14 SENT FMTS Register . . . 652
25.3.1.15 SENT SMBIT3 Register . . . 653
25.3.1.16 SENT SMBIT2 Register . . . 655
25.3.1.17 SENT SMTS Register . . . 655
25.3.1.18 SENT CH_CLK Register . . . 656
25.3.1.19 SENT CH_STS Register . . . 657
25.3.1.20 SENT CH_INTE Register . . . 659
25.3.1.21 SENT CH_CFG Register . . . 661
25.3.1.22 SENT CH_FMDATA Register . . . 663
25.3.1.23 SENT CH_FMCHNUM Register . . . 665
25.3.1.24 SENT CH_FMTS Register . . . 665
25.3.1.25 SENT CH_SMBIT3 Register . . . 666
25.3.1.26 SENT CH_SMBIT2 Register . . . 668
25.3.1.27 SENT CH_SMTS Register . . . 668
25.4 Functional Description . . . 669
25.4.1 Data Receive . . . 669
25.4.1.1 Input Filter . . . 669
25.4.1.2 Rx Clock Generation and Adjustment . . . 670
25.4.1.3 Received Message Diagnostics . . . 670
25.4.1.3.1 Calibration Pulse Length Check . . . 670
25.4.1.3.2 Number of Falling Edge Check . . . 670
25.4.1.3.3 Checksum Check . . . 671
25.4.1.3.4 Nibble Value Check . . . 671
25.4.1.3.5 Successive Calibration Pulse Check . . . 671
25.4.1.3.6 Pause Pulse Diagnostic . . . 671
25.4.1.4 Time Stamp Logic . . . 671
25.4.1.4.1 Limitation . . . 672
25.4.1.5 Data Store and Access . . . 672
25.4.2 Modes of Operation . . . 672
25.4.2.1 Debug Mode . . . 672
25.4.2.2 Low Power Mode . . . 672
25.4.3 Interrupts . . . 673
25.4.4 DMA Usage . . . 673
25.4.4.1 FIFO Access . . . 673
25.4.4.2 Buffer Access . . . 673
25.4.4.3 Error Access . . . 673
25.4.5 Initialization Flow . . . 674
25.5 Glossary . . . 674

Yuntu Microelectronics

26 Serial Audio Interface (SAI)
26.1 Chip-specific Information . . . 675
26.1.1 Instantiation Information . . . 675
26.1.2 MCLK Sources . . . 675
26.1.3 DMA Description . . . 675
26.1.4 Interrupt Description . . . 676
26.1.5 Power Mode Description . . . 676
26.2 Introduction . . . 676
26.2.1 Overview . . . 676
26.2.2 Features . . . 676
26.2.3 Block Diagram . . . 677
26.2.4 Signal Description . . . 677
26.3 Register Definition . . . 677
26.3.1 SAI Memory Map . . . 677
26.3.1.1 SAI TCTRL Register . . . 678
26.3.1.2 SAI TINTE Register . . . 679
26.3.1.3 SAI TSTS Register . . . 681
26.3.1.4 SAI TFCR Register . . . 681
26.3.1.5 SAI TCFG0 Register . . . 683
26.3.1.6 SAI TCFG1 Register . . . 685
26.3.1.7 SAI TCFG2 Register . . . 686
26.3.1.8 SAI TCFG3 Register . . . 687
26.3.1.9 SAI TDATA Register . . . 688
26.3.1.10 SAI TFR Register . . . 688
26.3.1.11 SAI TMR Register . . . 689
26.3.1.12 SAI RCTRL Register . . . 690
26.3.1.13 SAI RINTE Register . . . 691
26.3.1.14 SAI RSTS Register . . . 693
26.3.1.15 SAI RFCR Register . . . 693
26.3.1.16 SAI RCFG0 Register . . . 695
26.3.1.17 SAI RCFG1 Register . . . 697
26.3.1.18 SAI RCFG2 Register . . . 698
26.3.1.19 SAI RCFG3 Register . . . 699
26.3.1.20 SAI RDATA Register . . . 700
26.3.1.21 SAI RFR Register . . . 700
26.3.1.22 SAI RMR Register . . . 701
26.3.1.23 SAI MCLK_CTRL Register . . . 702
26.3.1.24 SAI MCLK_FRAC_DIV Register . . . 703
26.4 Function . . . 704
26.4.1 Master Clock . . . 704
26.4.1.1 Internal Master Clock . . . 704
26.4.1.2 External Master Clock . . . 705
26.4.2 Synchronous Mode . . . 705
26.4.2.1 Synchronous Internal . . . 705
26.4.2.2 Synchronous External . . . 705
26.4.3 Bit Clock . . . 705
26.4.4 Frame Sync . . . 705
26.4.5 Data Lines . . . 706
26.4.6 Data FIFO . . . 706
26.4.6.1 Data Alignment . . . 706

Yuntu Microelectronics

26.4.6.2 FIFO Packing . . . 706
26.4.6.3 FIFO Combine . . . 706
26.4.6.4 FIFO Request . . . 707
26.4.6.5 FIFO Warning . . . 707
26.4.6.6 FIFO Error . . . 707
26.4.7 Interrupt . . . 707
26.4.8 DMA . . . 708
26.5 Glossary . . . 708

VII Timer

27 Timer (TMR)
27.1 Chip-specific Information . . . 710
27.1.1 Instantiation Information . . . 710
27.2 Introduction . . . 710
27.2.1 Features . . . 710
27.2.2 Block Diagram . . . 711
27.3 Register Definition . . . 711
27.3.1 TMR Memory Map . . . 711
27.3.1.1 TMR CTRL Register . . . 712
27.3.1.2 TMR PRS Register . . . 712
27.3.1.3 TMR CNT Register . . . 713
27.3.1.4 TMR CH_CTRL Register . . . 714
27.3.1.5 TMR CH_INT Register . . . 715
27.3.1.6 TMR CH_CMP Register . . . 716
27.4 Functional Description . . . 717
27.4.1 Count-up Timer . . . 717
27.4.2 Compare Channels . . . 717
27.4.3 Operation Modes . . . 717
27.4.4 Configure the Timer . . . 718
27.4.5 Configure the Compare Channels . . . 718
27.4.6 Respond to Compare Channel Events . . . 718

28 Periodic Timer (pTMR)
28.1 Chip-specific pTMR Information . . . 719
28.1.1 Instantiation Information . . . 719
28.2 Introduction . . . 719
28.2.1 Block Diagram . . . 719
28.2.2 Features . . . 719
28.3 Register Definition . . . 720
28.3.1 pTMR Memory Map . . . 720
28.3.1.1 pTMR MCR Register . . . 720
28.3.1.2 pTMR CH_TSV Register . . . 721
28.3.1.3 pTMR CH_TCV Register . . . 722
28.3.1.4 pTMR CH_TCR Register . . . 722
28.3.1.5 pTMR CH_TFR Register . . . 723
28.4 Functional Description . . . 724
28.4.1 Timers . . . 724
28.4.2 Chain Mode . . . 724
28.4.3 Debug Mode . . . 724

Yuntu Microelectronics

28.4.4 Interrupt . . . 724

29 Low Power Timer (lpTMR)

29.1 Introduction . . . 725
29.1.1 Features . . . 725
29.2 Register Definition . . . 725
29.2.1 lpTMR Memory Map . . . 725
29.2.1.1 lpTMR CTRL Register . . . 726
29.2.1.2 lpTMR PRS Register . . . 727
29.2.1.3 lpTMR DIE Register . . . 729
29.2.1.4 lpTMR STS Register . . . 729
29.2.1.5 lpTMR CMP Register . . . 730
29.2.1.6 lpTMR LCNT Register . . . 731
29.2.1.7 lpTMR CNT Register . . . 732
29.3 Functional Description . . . 733
29.3.1 lpTMR Clocking . . . 733
29.3.2 lpTMR Prescaler/Glitch Filter . . . 733
29.3.2.1 Time Counter . . . 734
29.3.2.2 Pulse Counter . . . 734
29.3.3 lpTMR Compare . . . 734
29.3.4 lpTMR Counter . . . 734
29.3.5 lpTMR Hardware Trigger . . . 734
29.3.6 lpTMR Interrupt . . . 734
29.3.7 Operation Modes . . . 735
29.4 Glossary . . . 735

30 Enhanced Timer (eTMR)

30.1 Chip-specific eTMR Information . . . 736
30.1.1 Instantiation Information . . . 736
30.1.2 Signal Description . . . 736
30.1.3 Clock Description . . . 737
30.1.4 DMA Description . . . 737
30.1.5 Interrupt Description . . . 738
30.1.6 Power mode Description . . . 739
30.2 Introduction . . . 739
30.2.1 Overview . . . 739
30.2.2 Features . . . 740
30.2.2.1 Common features . . . 740
30.2.2.2 Individual features . . . 741
30.2.3 Block diagram . . . 741
30.3 Register Definition . . . 742
30.3.1 eTMR Memory Map . . . 742
30.3.1.1 eTMR STS Register . . . 746
30.3.1.2 eTMR IOSTS Register . . . 749
30.3.1.3 eTMR CNT Register . . . 750
30.3.1.4 eTMR QDCNT Register . . . 751
30.3.1.5 eTMR CTRL Register . . . 752
30.3.1.6 eTMR SYNC Register . . . 754
30.3.1.7 eTMR OTRIG Register . . . 757
30.3.1.8 eTMR INTE Register . . . 758

Yuntu Microelectronics

30.3.1.9 eTMR CHINIT Register . . . 760
30.3.1.10 eTMR CHSWV Register . . . 763
30.3.1.11 eTMR CHMASK Register . . . 764
30.3.1.12 eTMR CHFV Register . . . 766
30.3.1.13 eTMR INIT Register . . . 768
30.3.1.14 eTMR MOD Register . . . 769
30.3.1.15 eTMR DITHMOD Register . . . 770
30.3.1.16 eTMR MID Register . . . 771
30.3.1.17 eTMR QDCTRL Register . . . 772
30.3.1.18 eTMR QDINIT Register . . . 774
30.3.1.19 eTMR QDMOD Register . . . 774
30.3.1.20 eTMR FAULT Register . . . 775
30.3.1.21 eTMR CH_CTRL Register . . . 777
30.3.1.22 eTMR CH_VAL0 Register . . . 779
30.3.1.23 eTMR CH_VAL1 Register . . . 780
30.3.1.24 eTMR CH_DITHVAL1 Register . . . 780
30.3.1.25 eTMR CH_DT Register . . . 781
30.3.1.26 eTMR CH_CVAL Register . . . 781
30.3.1.27 eTMR CH_PPCNTN Register . . . 782
30.3.1.28 eTMR CH_PPCNTV Register . . . 782
30.3.1.29 eTMR CH_NPCNTN Register . . . 783
30.3.1.30 eTMR CH_NPCNTV Register . . . 783
30.4 Functional Description . . . 784
30.4.1 Clock . . . 784
30.4.2 Counter . . . 784
30.4.2.1 Global Time Base . . . 785
30.4.3 Common timer . . . 785
30.4.4 PWM mode . . . 785
30.4.4.1 Center Aligned PWM . . . 786
30.4.4.2 Edge Aligned PWM . . . 786
30.4.4.3 Asymmetrical PWM . . . 787
30.4.4.4 Software-generated PWM . . . 788
30.4.4.5 Double-switch PWM . . . 788
30.4.4.6 PWM Source . . . 788
30.4.4.7 Independent Mode . . . 788
30.4.4.8 Complementary Mode . . . 789
30.4.4.9 Complementary Mode with Dead-time Insertion . . . 789
30.4.4.10 Fault Detection and Protection . . . 790
30.4.4.10.1 Safe Value . . . 790
30.4.4.10.2 Fault Source Selection . . . 791
30.4.4.10.3 Fault Input Polarity . . . 791
30.4.4.10.4 Fault Filter . . . 791
30.4.4.10.5 Input Stretch . . . 791
30.4.4.10.6 Combinational fault . . . 791
30.4.4.10.7 Fault Recovery . . . 792
30.4.4.10.8 Fault Flag Clear . . . 792
30.4.4.11 Mask Control . . . 792
30.4.5 Output Compare Mode . . . 792
30.4.6 Input Capture Mode . . . 792
30.4.6.1 Channel Filter . . . 793
30.4.6.2 Capture Pulse Width . . . 793

Yuntu Microelectronics

30.4.6.3 Capture Test . . . 794
30.4.7 Output Polarity . . . 794
30.4.8 Channels Initialization . . . 794
30.4.9 Loading Mechanism . . . 794
30.4.9.1 General Registers Loading . . . 795
30.4.9.2 Special Registers Loading . . . 795
30.4.9.3 Counter Loading . . . 796
30.4.9.4 Loading trigger . . . 796
30.4.9.5 Loading Frequency . . . 796
30.4.10 Dithering . . . 796
30.4.11 Quadrature Decoder . . . 797
30.4.11.1 Decoder Prescaler . . . 797
30.4.11.2 Count and Direction Decoder Mode . . . 797
30.4.11.3 Phase A and phase B Decoder Mode . . . 797
30.4.11.4 Decoder Input Polarity . . . 798
30.4.11.5 Decoder Filter . . . 798
30.4.11.6 Decoder Counter Overflow . . . 798
30.4.12 Filter . . . 799
30.4.13 Output Trigger . . . 799
30.4.13.1 Width Fixed Pulse . . . 799
30.4.13.2 Width Adjustable Pulse . . . 800
30.4.13.3 Output Frequency . . . 800
30.4.14 Input Capture Source . . . 800
30.4.15 Hall Sensor Support . . . 800
30.4.16 Modulation Implementation . . . 800
30.4.17 Power Mode . . . 800
30.4.18 Interrupts . . . 801
30.4.19 DMA . . . 801
30.4.20 Reset . . . 801
30.5 Glossary . . . 801

31 Multiple Pulse Width Modulation (MPWM)

31.1 Chip-specific Information . . . 802
31.1.1 Instantiation Information . . . 802
31.1.2 Signal Description . . . 802
31.1.3 Clock Description . . . 803
31.1.4 DMA Description . . . 803
31.1.5 Interrupt Description . . . 803
31.1.6 Power mode Description . . . 804
31.2 Introduction . . . 804
31.2.1 Overview . . . 804
31.2.2 Features . . . 804
31.2.3 Block diagram . . . 805
31.3 Register Definition . . . 805
31.3.1 MPWM Memory Map . . . 805
31.3.1.1 MPWM IOSTS Register . . . 808
31.3.1.2 MPWM STS0 Register . . . 810
31.3.1.3 MPWM STS1 Register . . . 812
31.3.1.4 MPWM INTE0 Register . . . 815
31.3.1.5 MPWM INTE1 Register . . . 818

Yuntu Microelectronics

31.3.1.6 MPWM CTRL Register . . . 820
31.3.1.7 MPWM SWT Register . . . 822
31.3.1.8 MPWM INIT Register . . . 824
31.3.1.9 MPWM POL Register . . . 826
31.3.1.10 MPWM CH_CFG Register . . . 829
31.3.1.11 MPWM CH_CNT Register . . . 831
31.3.1.12 MPWM CH_CAP Register . . . 832
31.3.1.13 MPWM CH_PERIOD Register . . . 832
31.3.1.14 MPWM CH_CMP Register . . . 833
31.4 Function . . . 834
31.4.1 Clock . . . 834
31.4.2 Common Timer . . . 834
31.4.3 PWM Mode . . . 835
31.4.3.1 Edge Aligned PWM . . . 836
31.4.4 Output Compare Mode . . . 837
31.4.5 Input Capture Mode . . . 837
31.4.6 Pulse Counter Mode . . . 838
31.4.7 Output Polarity . . . 838
31.4.8 Channels Initialization . . . 838
31.4.9 Loading Mechanism . . . 838
31.4.9.1 Loading Trigger . . . 840
31.4.10 Output Trigger . . . 840
31.4.11 Power Mode . . . 841
31.4.12 Interrupts . . . 841
31.4.13 DMA . . . 841
31.4.14 Reset . . . 841
31.5 Glossary . . . 841

32 Real Time Clock (RTC)
32.1 Introduction . . . 842
32.1.1 Features . . . 842
32.2 Register Definition . . . 842
32.2.1 RTC Memory Map . . . 842
32.2.1.1 RTC SEC Register . . . 842
32.2.1.2 RTC CMP Register . . . 843
32.2.1.3 RTC ALM Register . . . 845
32.2.1.4 RTC EN Register . . . 846
32.2.1.5 RTC CTRL Register . . . 846
32.2.1.6 RTC INTF Register . . . 848
32.2.1.7 RTC INTE Register . . . 849
32.3 Functional Description . . . 850
32.3.1 General . . . 850
32.3.1.1 Software Reset . . . 850
32.3.1.2 Time Counter . . . 850
32.3.1.3 Compensation . . . 850
32.3.1.4 Register Lock . . . 850
32.3.2 Interrupts . . . 851
32.3.3 Operation Modes . . . 851

Yuntu Microelectronics

33 Programmable Trigger Unit (PTU)
33.1 Chip-specific PTU Information . . . 852
33.1.1 Instantiation Information . . . 852
33.2 Introduction . . . 852
33.2.1 Overview . . . 852
33.2.2 Features . . . 852
33.2.3 Block Diagram . . . 852
33.3 Register Definition . . . 853
33.3.1 PTU Memory Map . . . 853
33.3.1.1 PTU CTRL Register . . . 854
33.3.1.2 PTU PRS Register . . . 856
33.3.1.3 PTU INTE Register . . . 857
33.3.1.4 PTU STS Register . . . 858
33.3.1.5 PTU MOD Register . . . 859
33.3.1.6 PTU CNT Register . . . 860
33.3.1.7 PTU INTDLY Register . . . 861
33.3.1.8 PTU CHCTRL Register . . . 862
33.3.1.9 PTU CHSTS Register . . . 863
33.3.1.10 PTU CHDLYn Register . . . 864
33.3.1.11 PTU POEN Register . . . 865
33.3.1.12 PTU PODLY_PODLY0 Register . . . 866
33.3.1.13 PTU PODLY_PODLY1 Register . . . 867
33.4 Functional Description . . . 867
33.4.1 PTU Counter . . . 867
33.4.2 PTU Trigger . . . 867
33.4.3 PTU Pulse Out . . . 868
33.4.4 Operation Modes . . . 868
33.5 Glossary . . . 868

VIII Analog

34 Analog-to-Digital Converter (ADC)
34.1 Chip-specific ADC information . . . 870
34.1.1 Instantiation Information . . . 870
34.1.2 Signal Description . . . 870
34.1.3 Clock Description . . . 872
34.1.4 DMA Description . . . 872
34.1.5 Interrupt Description . . . 872
34.1.6 Power mode Description . . . 873
34.2 Introduction . . . 873
34.2.1 Overview . . . 873
34.2.2 Features . . . 873
34.2.3 Block diagram . . . 874
34.3 Register Definition . . . 874
34.3.1 ADC Memory Map . . . 874
34.3.1.1 ADC STS Register . . . 875
34.3.1.2 ADC INTE Register . . . 877
34.3.1.3 ADC CTRL Register . . . 879
34.3.1.4 ADC CFG0 Register . . . 880
34.3.1.5 ADC CFG1 Register . . . 882

Yuntu Microelectronics

34.3.1.6 ADC SMP Register . . . 883
34.3.1.7 ADC CHSEL Register . . . 884
34.3.1.8 ADC FIFO Register . . . 885
34.3.1.9 ADC TSC Register . . . 885
34.3.1.10 ADC CAL Register . . . 886
34.3.1.11 ADC WDCTRL Register . . . 887
34.3.1.12 ADC WDTH Register . . . 889
34.4 Functional Description . . . 889
34.4.1 Basic Configuration . . . 890
34.4.1.1 ADC on-off Control . . . 890
34.4.1.2 ADC Clock . . . 891
34.4.1.3 Channel Selection . . . 891
34.4.1.4 ADC Configuration . . . 892
34.4.1.5 Starting conversions . . . 892
34.4.1.6 Stopping an ongoing conversion . . . 892
34.4.2 Timings . . . 893
34.4.3 Functional Mode . . . 894
34.4.3.1 Single conversion mode . . . 894
34.4.3.2 Continuous conversion mode . . . 894
34.4.3.3 Discontinuous Mode . . . 895
34.4.3.4 Conversion on external trigger . . . 895
34.4.3.5 Programmable resolution . . . 896
34.4.3.6 End of sampling/conversion . . . 896
34.4.3.7 End of conversion sequence . . . 896
34.4.4 Calibration . . . 896
34.4.5 Data management . . . 896
34.4.5.1 Data FIFO . . . 896
34.4.5.2 ADC overrun . . . 897
34.4.5.3 Managing a sequence of data converted without using the DMA . . . 897
34.4.5.4 Managing converted data without using the DMA and overrun . . . 897
34.4.5.5 Managing converted data using the DMA . . . 898
34.4.6 Power Saving Modes . . . 898
34.4.6.1 Wait Mode . . . 898
34.4.6.2 Auto-off Mode . . . 898
34.4.7 Analog Window Watchdog . . . 898
34.4.8 Temperature Sensor Description . . . 899
34.4.9 Interrupts . . . 899
34.5 Glossary . . . 900

35 Analog Comparator (ACMP)
35.1 Chip-specific ACMP information . . . 901
35.1.1 Instantiation Information . . . 901
35.1.2 Signal Description . . . 901
35.1.3 Clock Description . . . 902
35.1.4 DMA Description . . . 902
35.1.5 Interrupt Description . . . 902
35.1.6 Power Mode Description . . . 902
35.2 Introduction . . . 902
35.2.1 Overview . . . 902
35.2.2 Features . . . 903

Yuntu Microelectronics

35.2.3 Block diagram . . . 903
35.3 Register Definition . . . 904
35.3.1 ACMP Memory Map . . . 904
35.3.1.1 ACMP CTRL Register . . . 904
35.3.1.2 ACMP DAC Register . . . 906
35.3.1.3 ACMP MUX Register . . . 907
35.3.1.4 ACMP FILT Register . . . 908
35.3.1.5 ACMP DMACR Register . . . 909
35.3.1.6 ACMP STS Register . . . 910
35.3.1.7 ACMP INTE Register . . . 913
35.3.1.8 ACMP CONT Register . . . 914
35.3.1.9 ACMP EXP Register . . . 915
35.4 Functional Description . . . 917
35.4.1 Input Control . . . 917
35.4.1.1 Input Selection . . . 917
35.4.1.2 DAC (Digital to Analog Converter) . . . 917
35.4.2 Output Control . . . 917
35.4.2.1 Polarity . . . 917
35.4.2.2 Filter . . . 917
35.4.3 Function Mode . . . 918
35.4.3.1 Common Mode . . . 918
35.4.3.2 Sample Mode . . . 918
35.4.3.3 Window Mode . . . 918
35.4.3.4 Continuous Mode . . . 918
35.4.3.4.1 Channel Fix . . . 918
35.4.3.4.2 Channel Select . . . 919
35.4.3.4.3 Timing . . . 919
35.4.3.4.4 Scan Result . . . 919
35.4.3.4.5 Loop Mode . . . 919
35.4.3.4.6 One-shot Mode . . . 919
35.4.4 Modes of Operation . . . 920
35.4.5 Interrupts . . . 920
35.4.6 DMA Usage . . . 920
35.4.7 Reset . . . 920
35.5 Glossary . . . 920

36 Temperature Sensor (TEMPSENSOR)
36.1 Introduction . . . 921

IX Security, Integrity and Safety

37 Programmable Cyclic Redundancy Check (PCRC)
37.1 Introduction . . . 923
37.1.1 Features . . . 923
37.1.2 Block diagram . . . 924
37.2 Register Definition . . . 924
37.2.1 PCRC Memory Map . . . 924
37.2.1.1 PCRC CTRL Register . . . 924
37.2.1.2 PCRC POLY Register . . . 926
37.2.1.3 PCRC INIT Register . . . 926

Yuntu Microelectronics

37.2.1.4 PCRC DATA_IN32 Register . . . 927
37.2.1.5 PCRC DATA_IN16 Register . . . 928
37.2.1.6 PCRC DATA_IN8 Register . . . 928
37.2.1.7 PCRC RESULT Register . . . 928
37.3 Functional description . . . 929
37.3.1 CRC initialization . . . 929
37.3.2 CRC polynomials . . . 929
37.3.3 Type of transpose . . . 930
37.3.4 Bit inversion . . . 931

38 Hardware Cryptography Unit (HCU)
38.1 Chip-specific HCU information . . . 932
38.1.1 Instantiation Information . . . 932
38.2 Introduction . . . 932
38.2.1 Block diagram . . . 932
38.2.2 Features . . . 933
38.2.3 Signal Description . . . 933
38.3 Register Definition . . . 933
38.3.1 HCU Memory Map . . . 933
38.3.1.1 HCU GO Register . . . 934
38.3.1.2 HCU CR Register . . . 935
38.3.1.3 HCU MSGINFO Register . . . 938
38.3.1.4 HCU MSGADL Register . . . 938
38.3.1.5 HCU FIFOWM Register . . . 939
38.3.1.6 HCU INTE Register . . . 940
38.3.1.7 HCU SR Register . . . 942
38.3.1.8 HCU KEY Register . . . 943
38.3.1.9 HCU AESCV Register . . . 944
38.3.1.10 HCU AESIV Register . . . 945
38.3.1.11 HCU AESMAC Register . . . 946
38.3.1.12 HCU SHAICV Register . . . 947
38.3.1.13 HCU IFDAT Register . . . 948
38.3.1.14 HCU OFDAT Register . . . 949
38.4 Functional description . . . 950
38.4.1 General . . . 950
38.4.2 Use Flow . . . 950
38.4.2.1 AES Engine . . . 950
38.4.2.1.1 ECB . . . 950
38.4.2.1.2 CBC . . . 951
38.4.2.1.3 CTR . . . 951
38.4.2.1.4 CCM . . . 951
38.4.2.1.5 CMAC . . . 952
38.4.2.2 SM4 Engine . . . 952
38.4.2.2.1 ECB . . . 952
38.4.2.3 SHA Engine . . . 952
38.4.2.3.1 SHA-256/SHA-384 generate hash value . . . 953
38.4.2.3.2 SHA-256/SHA-384 verify hash value . . . 953
38.4.3 Multiple Message Blocks For AES/SHA . . . 953
38.4.4 Context Store . . . 954
38.4.5 Secure key . . . 954
38.4.6 Data swapping . . . 954

Yuntu Microelectronics

38.4.7 Interrupts . . . 955
38.4.8 DMA . . . 956
38.4.9 Clock Gating Strategy . . . 956
38.5 Glossary . . . 956

39 True Random Number Generator (TRNG)
39.1 Introduction . . . 957
39.1.1 Features . . . 957
39.1.2 Block diagram . . . 958
39.2 Register Definition . . . 958
39.2.1 TRNG Memory Map . . . 958
39.2.1.1 TRNG CTRL Register . . . 959
39.2.1.2 TRNG STAT Register . . . 960
39.2.1.3 TRNG SDCTL Register . . . 962
39.2.1.4 TRNG FRQMIN Register . . . 962
39.2.1.5 TRNG FRQMAX Register . . . 963
39.2.1.6 TRNG FRQCNT Register . . . 964
39.2.1.7 TRNG SCML Register . . . 965
39.2.1.8 TRNG SCMC Register . . . 966
39.2.1.9 TRNG ENT Register . . . 967
39.3 Functional description . . . 968
39.3.1 Ring Oscillator . . . 968
39.3.2 Random Bits Collector . . . 968
39.3.3 Clock Monitor . . . 969
39.3.4 Fault Detector . . . 969
39.3.4.1 Long Run Detector . . . 969
39.3.4.2 Monobit Limitation . . . 969
39.3.4.3 Retry Count Limitation . . . 969
39.4 Glossary . . . 969

40 Watch Dog (WDG)
40.1 Introduction . . . 970
40.1.1 Features . . . 970
40.2 Register Definition . . . 970
40.2.1 WDG Memory Map . . . 970
40.2.1.1 WDG SVCR Register . . . 971
40.2.1.2 WDG CR Register . . . 971
40.2.1.3 WDG LR Register . . . 973
40.2.1.4 WDG TOVR Register . . . 974
40.2.1.5 WDG WVR Register . . . 974
40.2.1.6 WDG CNTCVR Register . . . 975
40.2.1.7 WDG INTF Register . . . 976
40.3 Functional Description . . . 977
40.3.1 General . . . 977
40.3.1.1 Operation Modes . . . 977
40.3.2 Interrupts . . . 978

41 ECC Management Unit (EMU)
41.1 Chip-specific EMU Information . . . 979

Yuntu Microelectronics

41.2 Introduction . . . 979
41.2.1 Overview . . . 979
41.2.1.1 Error Injection . . . 979
41.2.1.2 Error Report . . . 979
41.2.2 Features . . . 980
41.3 Register Definition . . . 980
41.3.1 Programming Model Access . . . 980
41.3.2 Error Injection Registers . . . 980
41.3.3 Error Report Registers . . . 980
41.3.4 EMU Memory Map . . . 980
41.3.4.1 EMU EIGCR Register . . . 982
41.3.4.2 EMU EICHCR Register . . . 983
41.3.4.3 EMU EICHD_ADDR Register . . . 983
41.3.4.4 EMU EICHD_CHKBIT Register . . . 984
41.3.4.5 EMU EICHD_DATAL Register . . . 985
41.3.4.6 EMU EICHD_DATAH Register . . . 985
41.3.4.7 EMU ERIE Register . . . 986
41.3.4.8 EMU ERIF Register . . . 986
41.3.4.9 EMU ERINFO_ADDR Register . . . 987
41.3.4.10 EMU ERINFO_SYND Register . . . 988
41.3.4.11 EMU ERINFO_CNT Register . . . 988
41.4 Functional Description . . . 989
41.4.1 General . . . 989
41.4.2 Error Injection . . . 989
41.4.3 Error Report . . . 989

42 Interrupt Monitor (INTM)
42.1 Chip-specific INTM Information . . . 990
42.1.1 Instantiation Information . . . 990
42.2 Introduction . . . 990
42.2.1 Features . . . 990
42.3 Register Definition . . . 990
42.3.1 INTM Memory Map . . . 991
42.3.1.1 INTM GCTRL Register . . . 991
42.3.1.2 INTM GINTACK Register . . . 992
42.3.1.3 INTM MON_SEL Register . . . 993
42.3.1.4 INTM MON_THD Register . . . 994
42.3.1.5 INTM MON_TMR Register . . . 995
42.3.1.6 INTM MON_STS Register . . . 995
42.4 Functional Description . . . 996

43 Peripheral Protection Unit (PPU)
43.1 Introduction . . . 997
43.1.1 Features . . . 997
43.1.2 Block Diagram . . . 998
43.1.3 Signal Description . . . 998
43.1.4 PPU Protection IPs and Registers . . . 998
43.2 Register Definition . . . 1001
43.2.1 PPU Memory Map . . . 1002
43.2.1.1 PPU SLOCK Register . . . 1026

Yuntu Microelectronics

43.2.1.2 PPU PROTCFG Register . . . 1026
43.3 Functional Description . . . 1027
43.3.1 Soft Lock . . . 1027
43.3.2 Hard Lock . . . 1028
43.3.3 User Access Enable . . . 1028

44 Fault Management Unit(FMU)
44.1 Chip-specific Information . . . 1029
44.1.1 FMU Fault Configuration . . . 1029
44.1.2 FMU Clocking . . . 1030
44.2 Introduction . . . 1030
44.2.1 Overview . . . 1030
44.2.2 Features . . . 1030
44.2.3 Block diagram . . . 1032
44.3 Register Definition . . . 1032
44.3.1 FMU Memory Map . . . 1032
44.3.1.1 FMU LOCK Register . . . 1034
44.3.1.2 FMU OP Register . . . 1035
44.3.1.3 FMU CTRL Register . . . 1036
44.3.1.4 FMU ST Register . . . 1037
44.3.1.5 FMU CFG_CNT Register . . . 1039
44.3.1.6 FMU FAULT_ST Register . . . 1039
44.3.1.7 FMU FAULT_ST_KEY Register . . . 1040
44.3.1.8 FMU FAULT_FSM_ST Register . . . 1041
44.3.1.9 FMU ALARM_CNT_TO_TH Register . . . 1042
44.3.1.10 FMU ALARM_CNT Register . . . 1043
44.3.1.11 FMU FOUT_CFG Register . . . 1044
44.3.1.12 FMU FOUT_MIN_WIN Register . . . 1046
44.3.1.13 FMU FOUT_FRC Register . . . 1046
44.3.1.14 FMU FOUT_CNT Register . . . 1047
44.3.1.15 FMU FFTR_CTRL Register . . . 1048
44.3.1.16 FMU FFTR_CNT_TO_TH Register . . . 1049
44.3.1.17 FMU FFTR_CNT Register . . . 1050
44.3.1.18 FMU FAULT_INJ Register . . . 1051
44.3.1.19 FMU FIN_CTRL Register . . . 1052
44.3.1.20 FMU FCH_CTRL Register . . . 1054
44.4 Function . . . 1055
44.4.1 FMU Reset Source . . . 1055
44.4.2 FMU FAULT Inputs . . . 1055
44.4.2.1 Fault Inputs On-chip . . . 1055
44.4.2.2 Fault Inputs from PAD(FMU_ERR_IN) . . . 1055
44.4.2.3 Fault Recovery . . . 1055
44.4.2.3.1 Software Recovery . . . 1056
44.4.2.3.2 Hardware Recovery . . . 1056
44.4.2.4 Disable FAULT in Debug Mode . . . 1056
44.4.3 FMU FSM . . . 1056
44.4.3.1 NORMAL . . . 1056
44.4.3.2 CFG . . . 1056
44.4.3.3 ALARM . . . 1056
44.4.3.4 FAULT . . . 1057

Yuntu Microelectronics

44.4.3.5 FMU Fault Indication Output . . . 1057
44.4.3.5.1 Bi-Stable Protocol . . . 1057
44.4.3.5.2 Fault Output Counter . . . 1058
44.4.3.6 Interrupt Generation . . . 1058
44.4.3.7 FMU Reset Generation . . . 1058
44.4.4 FFTR(FMU Fail To React) . . . 1059
44.4.4.1 FFTR Counter . . . 1059
44.4.4.2 FFTR Reset Generation . . . 1059
44.5 Glossary . . . 1059

X Appendix

A

A.1 Revision History . . . 1061
A.2 Release Notes . . . 1062
A.2.1 Introduction Changes . . . 1062
A.2.2 Memory Map and Memories Changes . . . 1062
A.2.2.1 Memory Map Changes . . . 1062
A.2.2.2 Embedded FLASH Module (EFM) Changes . . . 1062
A.2.2.3 Read-Only Memory (ROM) Changes . . . 1062
A.2.2.4 On-Chip RAM (OCRAM) Changes . . . 1062
A.2.2.5 Register File (REGFILE) Changes . . . 1062
A.2.3 Signal multiplexing Changes . . . 1063
A.2.3.1 Pinmux Changes . . . 1063
A.2.3.2 General Purpose Input/Output (GPIO) Changes . . . 1063
A.2.3.3 Port Control (PCTRL) Changes . . . 1063
A.2.4 Core and System Changes . . . 1063
A.2.4.1 CM7 Overview Changes . . . 1063
A.2.4.2 Direct Memory Access (DMA) Changes . . . 1063
A.2.4.3 Chip Integration Module (CIM) Changes . . . 1063
A.2.4.4 Trigger Multiplexer Module (TMU) Changes . . . 1063
A.2.5 Clock,Reset and Power Management Changes . . . 1064
A.2.5.1 System Clock Unit (SCU) Changes . . . 1064
A.2.5.2 IP Control (IPC) Changes . . . 1064
A.2.5.3 Reset Control Unit (RCU) Changes . . . 1064
A.2.5.4 Power Control Unit (PCU) Changes . . . 1064
A.2.5.5 Power Mode Changes . . . 1064
A.2.5.6 Wake-up Unit (WKU) Changes . . . 1064
A.2.6 Communication Interfaces Changes . . . 1065
A.2.6.1 Flexible Controller Area Network (FlexCAN) Changes . . . 1065
A.2.6.2 Local Interconnect Network (LIN) Changes . . . 1065
A.2.6.3 Serial Port Interface (SPI) Changes . . . 1065
A.2.6.4 Quad Serial Peripheral Interface (QSPI) Changes . . . 1065
A.2.6.5 Inter-Integrated Circuit (I2C) Changes . . . 1065
A.2.6.6 Single Edge Nibble Transmission (SENT) Changes . . . 1065
A.2.6.7 Serial Audio Interface (SAI) Changes . . . 1066
A.2.7 Timer Changes . . . 1066
A.2.7.1 Timer (TMR) Changes . . . 1066
A.2.7.2 Periodic Timer (pTMR) Changes . . . 1066
A.2.7.3 Low Power Timer (lpTMR) Changes . . . 1066

Yuntu Microelectronics

A.2.7.4 Enhanced Timer (eTMR) Changes 1066
A.2.7.5 Multiple Pulse Width Modulation (MPWM) Changes 1066
A.2.7.6 Real Time Clock (RTC) Changes. 1066
A.2.7.7 Programmable Trigger Unit (PTU) Changes 1067
A.2.8 Analog Changes 1067
A.2.8.1 Analog-to-Digital Converter (ADC) Changes 1067
A.2.8.2 Analog Comparator (ACMP) Changes 1067
A.2.8.3 Temperature Sensor (TEMPSENSOR) Changes 1067
A.2.9 Security, Integrity and Safety Changes 1067
A.2.9.1 Programmable Cyclic Redundancy Check (PCRC) Changes 1067
A.2.9.2 Hardware Cryptography Unit (HCU) Changes 1067
A.2.9.3 Ture Random Number Generator (TRNG) Changes 1067
A.2.9.4 Watch Dog (WDG) Changes. 1068
A.2.9.5 ECC Management Unit (EMU) Changes. 1068
A.2.9.6 Interrupt Monitor (INTM) Changes 1068
A.2.9.7 Peripheral Protection Unit (PPU) Changes. 1068
A.2.9.8 Fault Management Unit (FMU) Changes 1068

Yuntu Microelectronics

Part I

Introduction

Yuntu Microelectronics

Chapter 1

Introduction

1.1 Overview

YTM32B1HA0x series provide the highly scalable portfolio of ARM® Cortex®-M7 MCUs in the automotive industry. With 2.97 ~ 5.5 V supply and focus on exceptional EMC/ESD robustness, YTM32B1HA0x series devices are well suited to a wide range of applications in electrical harsh environments, and are optimized for cost-sensitive applications offering low pin-count option.

The YTM32B1HA0x series offer a broad range of memory, peripherals and package options. They share common peripherals and pin counts allowing developers to migrate easily within an MCU family or among the MCU families to take advantage of more memory or feature integration. This scalability allows developers to standardize on the YTM32B1HA0x series for their end product platforms, maximizing hardware and software reuse and reducing time-to-market.

1.2 Features Summary

Following are the overview of device features:

- AEC-Q100 qualified
- ASIL-D compliant
- Arm Cortex-M7 with lockstep, up to 200 MHz
- Memories
  - 8 KB I-Cache, 8 KB D-Cache with ECC support
  - Up to 2 MB Program Flash memory and 256 KB Data Flash with ECC support, and support OTA
  - Up to 256 KB OCRAM, supports ECC feature, and includes 32 KB OCRAM block which supports data retention in Powerdown mode
  - Up to 32 KB ITCM and 128 KB DTCM with ECC support
  - 32 bytes register file (REGFILE) which supports data retention in Powerdown mode
  - 64 KB ROM with ECC support
- Provide multiple clock sources including:
  - 96 MHz Fast Internal RC Oscillator (FIRC)
  - 4~40 MHz Fast Crystal Oscillator (FXOSC)
  - Up to 200 MHz Phase-Locked Loop (PLL)
  - 12 MHz Slow Internal RC Oscillator (SIRC)
  - 32.768 KHz Slow Crystal Oscillator (SXOSC)

Yuntu Microelectronics

- Power Control Unit (PCU) with internal regulators capable of supporting multiple power modes
  - Active
  - Sleep
  - Deepsleep
  - Standby
  - Powerdown
- Support up to 64 WKU pins to wake up from Powerdown mode
- 32 DMA channels with up to 111 hardware trigger sources
- I/O supporting 2.97 ~ 5.5 V supply
- Wide operating voltage ranges (2.97 ~ 5.5 V) with fully functional flash memory program/erase/read operations
- Debug functionality
  - Joint Test Action Group (IEEE 1149.1 standard)
  - Serial Wire Debug (SWD)
- Human-machine interface
  - Up to 154 general-purpose input/output (GPIO)
  - External interrupt
- Analog
  - Two 12-bit, 2Msps SAR ADCs, up to 2x32 external channels and 8 internal channels
  - Two On-chip Analog Comparators (ACMPs) with 8-bit DAC, up to 2x8 channels
  - Support temperature sensor
- Timers
  - Two Timer (TMR) modules
  - Three Periodic Timer (pTMR) modules with 4 channels respectively
  - One Low Power Timer (lpTMR)
  - Six Enhanced Timer (eTMR) modules with 8 channels respectively
  - Three Multiple Pulse Width Modulation (MPWM) modules with 16 channels respectively
  - One Real-Time Clock (RTC)
  - Two Programmable Trigger Unit (PTU) modules
- Serial communication interfaces
  - Eight FlexCAN modules with FD
  - Ten LINFlexD modules
  - Eight Serial Peripheral Interface (SPI) modules
  - One Quad Serial Peripheral Interface (QSPI) module
  - Five Inter-Integrated Circuit (I2C) modules
  - Two Single Edge Nibble Transmission (SENT) modules with 4 channels respectively
  - Two Serial Audio Interface (SAI) modules
- Security, integrity and safety
  - Programmable Cyclic Redundancy Checker (PCRC)
  - Hardware Cryptography Unit (HCU) which supports AES/SM4/SHA
  - True Random Number Generator (TRNG)
  - Clock Monitor Unit (CMU)
  - Watchdog (WDG)
  - MPU for dynamic task protection (16 regions)
  - Peripheral Protection Unit (PPU)
  - Interrupt Monitor (INTM)
  - ECC Management Unit (EMU)

Yuntu Microelectronics

  - Fault Management Unit (FMU)
- Temperature range
  - Ambient operating temperature: -40 °C ~ 125 °C
  - Junction operating temperature: -40 °C ~ 150 °C
- Package options
  - 176-pin LQFP
  - 144-pin LQFP
  - 100-pin LQFP

1.3 Block Diagram

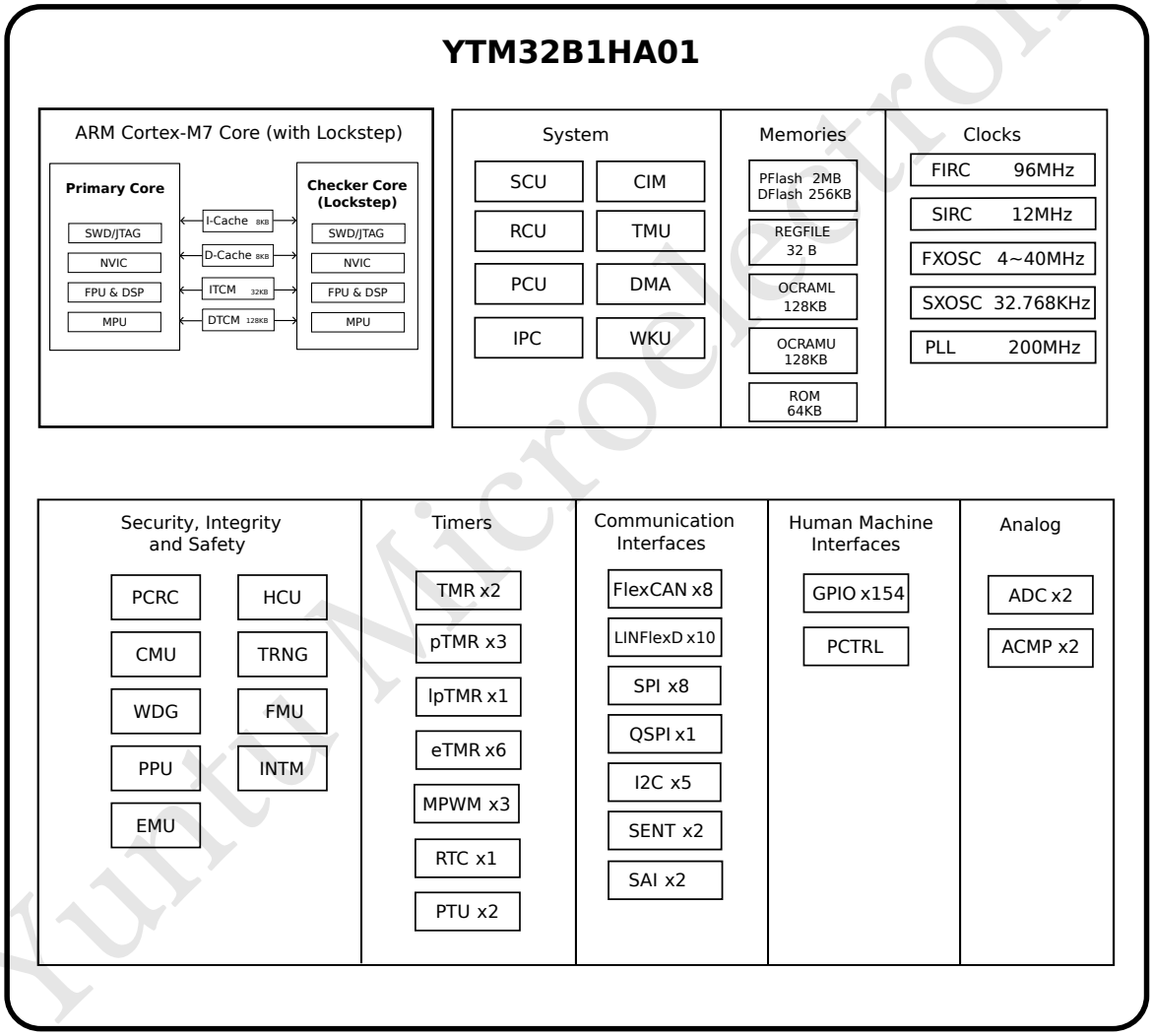


Figure 1.1: YTM32B1HA01 Block Diagram

1.4 Glossary

Yuntu Microelectronics

ACMP	Analog Comparator
ADC	Analog-to-Digital Converter
AES	Advanced Encryption Standard
CMU	Clock Monitor Unit
PCRC	Programmable Cyclic Redundancy Checker
DMA	Direct Memory Access
ECC	Error Correction Code
EMC	Electromagnetic Compatibility
EMU	ECC Management Unit
ESD	Electrostatic discharge
eTMR	Enhanced Timer
FIRC	Fast Internal RC Oscillator
FMU	Fault Management Unit
FXOSC	Fast Crystal Oscillator
GPIO	General-Purpose Input/Output
HCU	Hardware Cryptography Unit
I2C	Inter-Integrated Circuit
INTM	Interrupt Monitor
lpTMR	Low Power Timer
MPU	Memory Protection Unit
MPWM	Multiple Pulse Width Modulation
OCRAM	On-Chip RAM
OTA	Over-The-Air
PCU	Power Control Unit
PLL	Phase-Locked Loop
PPU	Peripheral Protection Unit
pTMR	Periodic Timer
PTU	Programmable Trigger Unit
QSPI	Quad Serial Peripheral Interface
REGFILE	Register File
RTC	Real-Time Clock
SAI	Serial Audio Interface
SENT	Single Edge Nibble Transmission
SHA	Secure Hash Algorithms
SIRC	Slow Internal RC Oscillator
SM4	Shang Mi 4
SPI	Serial Peripheral Interface
SWD	Serial Wire Debug
SXOSC	Slow Crystal Oscillator
TCM	Tightly Coupled Memory
TMR	Timer
TRNG	True Random Number Generator
WDG	Watchdog
WKU	Wake-Up Unit

Yuntu Microelectronics

Part II

Memory Map and Memories

Yuntu Microelectronics

Chapter 2

Memory Map

2.1 Introduction

This chip contains various memories and memory-mapped peripherals that are located in one 32-bit contiguous memory space. This chapter describes the memory and peripheral locations within memory space.

2.2 System Memory Map

Table 2.1: System Memory Map

Module	Start address	End address	Size
ITCM	0x0000_0000	0x0000_7FFF	32KB
ROM	0x0100_0000	0x0100_FFFF	64KB
PFlash0	0x0200_0000	0x020F_FFFF	1MB
PFlash1	0x0210_0000	0x021F_FFFF	1MB
DFlash	0x0600_0000	0x0603_FFFF	256KB
NVR0	0x1000_0000	0x1000_3FFF	16KB
NVR1	0x1002_0000	0x1002_3FFF	16KB
DTCM	0x2000_0000	0x2001_FFFF	128KB
OCRAML	0x2002_0000	0x2003_FFFF	128KB
OCRAMU	0x2004_0000	0x2005_FFFF	128KB
DTCM Backdoor	0x2006_0000	0x2007_FFFF	128KB
ITCM Backdoor	0x2008_0000	0x2008_7FFF	32KB
APB	0x4000_0000	0x401F_FFFF	2MB
QSPI RX buffer	0x6700_0000	0x6700_007F	128B
QSPI AHB	0x6800_0000	0x6FFF_FFFF	128MB
Private Peripheral Bus	0xE000_0000	0xE00F_FFFF	1MB

2.3 Peripheral Bus (APB) Memory Map

Yuntu Microelectronics

Table 2.2: APB Module Table

Slot	Start Address	End Address	Module
0	0x4000_0000	0x4000_3FFF	
1	0x4000_4000	0x4000_7FFF	
2	0x4000_8000	0x4000_BFFF	
3	0x4000_C000	0x4000_FFFF	
4	0x4001_0000	0x4001_3FFF	
5	0x4001_4000	0x4001_7FFF	
6	0x4001_8000	0x4001_BFFF	
7	0x4001_C000	0x4001_FFFF	
8	0x4002_0000	0x4002_3FFF	DMA
9	0x4002_4000	0x4002_7FFF	
10	0x4002_8000	0x4002_BFFF	
11	0x4002_C000	0x4002_FFFF	
12	0x4003_0000	0x4003_3FFF	
13	0x4003_4000	0x4003_7FFF	
14	0x4003_8000	0x4003_BFFF	
15	0x4003_C000	0x4003_FFFF	
16	0x4004_0000	0x4004_3FFF	EFM
17	0x4004_4000	0x4004_7FFF	GPIO
18	0x4004_8000	0x4004_BFFF	PCTRLA
19	0x4004_C000	0x4004_FFFF	PCTRLB
20	0x4005_0000	0x4005_3FFF	PCTRLC
21	0x4005_4000	0x4005_7FFF	PCTRLD
22	0x4005_8000	0x4005_BFFF	PCTRLE
23	0x4005_C000	0x4005_FFFF	
24	0x4006_0000	0x4006_3FFF	
25	0x4006_4000	0x4006_7FFF	
26	0x4006_8000	0x4006_BFFF	
27	0x4006_C000	0x4006_FFFF	LINFlexD0
28	0x4007_0000	0x4007_3FFF	LINFlexD1
29	0x4007_4000	0x4007_7FFF	LINFlexD2
30	0x4007_8000	0x4007_BFFF	LINFlexD3
31	0x4007_C000	0x4007_FFFF	LINFlexD4
32	0x4008_0000	0x4008_3FFF	LINFlexD5
33	0x4008_4000	0x4008_7FFF	LINFlexD6
34	0x4008_8000	0x4008_BFFF	LINFlexD7
35	0x4008_C000	0x4008_FFFF	LINFlexD8
36	0x4009_0000	0x4009_3FFF	LINFlexD9
37	0x4009_4000	0x4009_7FFF	I2C0
38	0x4009_8000	0x4009_BFFF	I2C1
39	0x4009_C000	0x4009_FFFF	I2C2
40	0x400A_0000	0x400A_3FFF	I2C3
41	0x400A_4000	0x400A_7FFF	I2C4
42	0x400A_8000	0x400A_BFFF	SPI0
43	0x400A_C000	0x400A_FFFF	SPI1

Yuntu Microelectronics

Table 2.2 continued from previous page

Slot	Start Address	End Address	Module
44	0x400B_0000	0x400B_3FFF	SPI2
45	0x400B_4000	0x400B_7FFF	SPI3
46	0x400B_8000	0x400B_BFFF	SPI4
47	0x400B_C000	0x400B_FFFF	SPI5
48	0x400C_0000	0x400C_3FFF	SPI6
49	0x400C_4000	0x400C_7FFF	SPI7
50	0x400C_8000	0x400C_BFFF	
51	0x400C_C000	0x400C_FFFF	
52	0x400D_0000	0x400D_3FFF	FlexCAN0
53	0x400D_4000	0x400D_7FFF	FlexCAN1
54	0x400D_8000	0x400D_BFFF	FlexCAN2
55	0x400D_C000	0x400D_FFFF	FlexCAN3
56	0x400E_0000	0x400E_3FFF	FlexCAN4
57	0x400E_4000	0x400E_7FFF	FlexCAN5
58	0x400E_8000	0x400E_BFFF	FlexCAN6
59	0x400E_C000	0x400E_FFFF	FlexCAN7
60	0x400F_0000	0x400F_3FFF	
61	0x400F_4000	0x400F_7FFF	
62	0x400F_8000	0x400F_BFFF	
63	0x400F_C000	0x400F_FFFF	
64	0x4010_0000	0x4010_3FFF	ADC0
65	0x4010_4000	0x4010_7FFF	ADC1
66	0x4010_8000	0x4010_BFFF	
67	0x4010_C000	0x4010_FFFF	ACMP0
68	0x4011_0000	0x4011_3FFF	ACMP1
69	0x4011_4000	0x4011_7FFF	
70	0x4011_8000	0x4011_BFFF	PTU0
71	0x4011_C000	0x4011_FFFF	PTU1
72	0x4012_0000	0x4012_3FFF	TMU
73	0x4012_4000	0x4012_7FFF	eTMR0
74	0x4012_8000	0x4012_BFFF	eTMR1
75	0x4012_C000	0x4012_FFFF	eTMR2
76	0x4013_0000	0x4013_3FFF	eTMR3
77	0x4013_4000	0x4013_7FFF	eTMR4
78	0x4013_8000	0x4013_BFFF	eTMR5
79	0x4013_C000	0x4013_FFFF	
80	0x4014_0000	0x4014_3FFF	
81	0x4014_4000	0x4014_7FFF	MPWM0
82	0x4014_8000	0x4014_BFFF	MPWM1
83	0x4014_C000	0x4014_FFFF	MPWM2
84	0x4015_0000	0x4015_3FFF	TMR0
85	0x4015_4000	0x4015_7FFF	TMR1
86	0x4015_8000	0x4015_BFFF	
87	0x4015_C000	0x4015_FFFF	
88	0x4016_0000	0x4016_3FFF	SENT0

Yuntu Microelectronics

Table 2.2 continued from previous page

Slot	Start Address	End Address	Module
89	0x4016_4000	0x4016_7FFF	SENT1
90	0x4016_8000	0x4016_BFFF	pTMR0
91	0x4016_C000	0x4016_FFFF	pTMR1
92	0x4017_0000	0x4017_3FFF	pTMR2
93	0x4017_4000	0x4017_7FFF	lpTMR0
94	0x4017_8000	0x4017_BFFF	
95	0x4017_C000	0x4017_FFFF	
96	0x4018_0000	0x4018_3FFF	RTC
97	0x4018_4000	0x4018_7FFF	REGFILE
98	0x4018_8000	0x4018_BFFF	WKU
99	0x4018_C000	0x4018_FFFF	
100	0x4019_0000	0x4019_3FFF	
101	0x4019_4000	0x4019_7FFF	PCRC
102	0x4019_8000	0x4019_BFFF	TRNG
103	0x4019_C000	0x4019_FFFF	HCU
104	0x401A_0000	0x401A_3FFF	QSPI
105	0x401A_4000	0x401A_7FFF	
106	0x401A_8000	0x401A_BFFF	WDG
107	0x401A_C000	0x401A_FFFF	
108	0x401B_0000	0x401B_3FFF	
109	0x401B_4000	0x401B_7FFF	
110	0x401B_8000	0x401B_BFFF	
111	0x401B_C000	0x401B_FFFF	
112	0x401C_0000	0x401C_3FFF	FMU
113	0x401C_4000	0x401C_7FFF	INTM
114	0x401C_8000	0x401C_BFFF	EMU
115	0x401C_C000	0x401C_FFFF	SAI0
116	0x401D_0000	0x401D_3FFF	SAI1
117	0x401D_4000	0x401D_7FFF	
118	0x401D_8000	0x401D_BFFF	
119	0x401D_C000	0x401D_FFFF	
120	0x401E_0000	0x401E_3FFF	
121	0x401E_4000	0x401E_7FFF	-
122	0x401E_8000	0x401E_BFFF	
123	0x401E_C000	0x401E_FFFF	CIM
124	0x401F_0000	0x401F_3FFF	SCU
125	0x401F_4000	0x401F_7FFF	IPC
126	0x401F_8000	0x401F_BFFF	PCU
127	0x401F_C000	0x401F_FFFF	RCU

2.4 Glossary

Yuntu Microelectronics

AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
DFlash	Data Flash
DTCM	Data Tightly Couple Memory
ITCM	Instruction Tightly Couple Memory
NVR	Nonvolatile Register
PFlash	Program Flash
ROM	Read-only Memory

Yuntu Microelectronics

Chapter 3

Embedded FLASH Module (EFM)

3.1 Introduction

The Embedded Flash Module (EFM) is a Flash memory block with sector/chip Erase and Program capability. It consists of two Program Flash (PFlash0 and PFlash1) and one Data Flash (DFlash).

A single power supply is used for the Flash array and no need for any external high-voltage power source while EFM is permitted program and erase operation.

The EFM includes a controller that executes commands to program or erase Flash array. Flash array must be erased before any program operation. Access to the Flash array is not allowed while it is under programming or erasing.

NVR arrays are used to store MCU information. Detailed usage of the NVR array is listed below:

- CUS_NVR (Customized NVR) stores customized information.
- HCU_NVR (Hardware Cryptography Unit NVR) stores user-defined private encryption keys and can store up to 128 sets of 128-bit or 192-bit or 256-bit keys.
- SHA_NVR (SHA-384 NVR) stores the SHA-384 (Secure Hash Algorithm) result of image code which programmed into PFlash.
- BOOT_NVR can only be used by Boot Swap command.
- OTP_NVR (One-time programmable NVR) stores information that can only be programmed once.

3.1.1 EFM Memory Map

Table 3.1: EFM Memory Map

Memory	Start address	End address	Size	Note
PFlash0	0x0200_0000	0x020F_FFFF	1 MB	Block 0
PFlash1	0x0210_0000	0x021F_FFFF	1 MB	Block 1
DFlash	0x0600_0000	0x0603_FFFF	256 KB	Block 2
CUS_NVR	0x1000_0000	0x1000_07FF	2 KB	Read/Erase/Program protected by CUS_KEY
HCU_NVR	0x1000_0800	0x1000_17FF	4 KB	Erase protected by CUS_KEY
USER_NVR	0x1000_1800	0x1000_2FFF	6 KB	

Yuntu Microelectronics

Table 3.1 continued from previous page

Memory	Start address	End address	Size	Note
SHA_NVR	0x1000_3000	0x1000_37FF	2 KB	Erase protected by CUS_KEY
BOOT_NVR	0x1000_3800	0x1000_3FFF	2 KB	Used for BOOT_SWAP command
OTP_NVR	0x1002_0000	0x1002_3FFF	16 KB	Erase invalid

3.1.2 Features

The EFM includes the following features:

- Two Program-Flash (PFlash0 and PFlash1) with ECC:
  - Each PFlash includes 1 MB main array
  - Each PFlash block includes 512 sectors, and the valid data size of each sector is 2KB
  - Program page size is 32 Byte
  - 16KB OTP (One-time Programmable)
  - 4KB AES key storage
  - Support RWW (read-while-write) and OTA (Over-The-Air) technology
- One Data-Flash (DFlash) with ECC:
  - 256 KB main array
  - Include 256 sectors, and the valid data size of each sector is 1KB
  - Program page size is 8 Byte
- Protection scheme against accidental program or erase operations
- Command protection function for authorization protection
- Optional interruptions on command completion and status update

3.1.3 Modes of Operation

- If CTRL[DPD_EN]=1, Flash arrays enter Deep Powerdown mode when MCU goes to Deepsleep mode. Power consumption of Flash could be further reduced in Deep Powerdown mode. Flash wake-up time takes at least 15μs.
- If CTRL[DPD_EN]=0, Flash arrays do not enter Deep Powerdown mode when MCU goes to Deepsleep mode. The Flash power consumption is high, but don't need wake-up time.

3.2 Register Definition

3.2.1 EFM Memory Map

EFM base address 40040000h

Table 3.2: EFM Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	800C0100h
4h	Status Register (STS)	32	RW	00000080h
8h	Command Register (CMD)	32	RW	00000000h
ch	Command Protect Register (CMD_PROT)	32	RW	00000000h
10h	Timing 1 Control Register (TIMING1)	32	RW	00050010h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
14h	Timing 2 Control Register (TIMING2)	32	RW	06408000h
18h	ECC Error Inject Mask Register 0 (ECCERR_INJECT_MSK_0)	32	RW	00000000h
1ch	ECC Error Inject Mask Register 1 (ECCERR_INJECT_MSK_1)	32	RW	00000000h
20h	ECC Error Inject Address Register (ECCERR_INJECT_ADDR)	32	RW	00000000h
24h	Address Protect Register (ADDR_PROT0)	32	RW	see description
28h	Address Protect Register (ADDR_PROT1)	32	RW	see description
2ch	Address Protect Register (ADDR_PROT2)	32	RW	see description
30h	EFM ECC Error Address (ECC_ERR_ADDR)	32	RW	FFFFFFFFh
200h	EFM Command Unlock (CMD_UNLOCK)	32	RW	00000000h
204h	EFM Customer Key (CUS_KEY)	32	RW	00000000h
300h	Override Control Register (CTRL_OVRD)	32	RW	00010000h

3.2.1.1 EFM CTRL Register

Offset: 0h

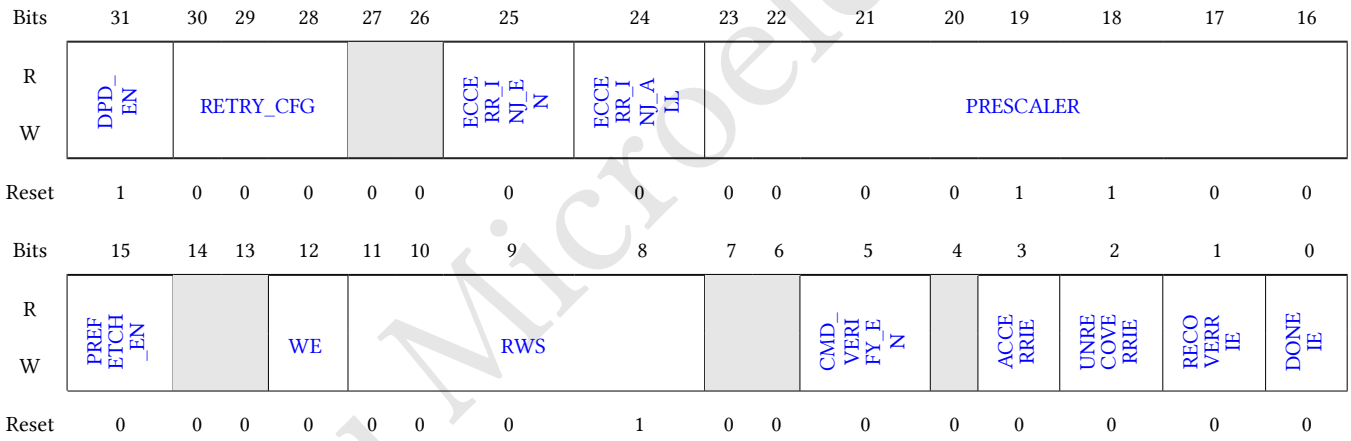


Table 3.3: EFM CTRL Register Description

Field	Function
31 DPD_EN	**Deep Powerdown Enable** This DPD_EN bit determines whether the Flash array will enter deep Powerdown mode when MCU is entering Deepsleep mode. 0b - Flash will keep active when MCU enters Deepsleep mode 1b - Flash will enter deep power-down mode when MCU enters Deepsleep mode
30 - 28	**Erase Retry Configuration**

Yuntu Microelectronics

Field	Function
RETRY_CFG	These RETRY_CFG bits set erase-retry order in manual-erase operation. 001b - For 1st pre-erase pulse 010b - For 1st pre-erase pulse 100b - For 1st erase-retry pulse 101b - For 2nd erase-retry pulse 110b - For 3rd erase-retry pulse 111b - For 4th to 20th erase-retry pulse
25 ECCERR_INJ_EN	**ECC Error Inject Enable** This ECCERR_INJ_EN bit determines whether to inject ECC error when reading Flash. 0b - Disable ECC error inject when reading Flash. 1b - Enable ECC error inject when reading Flash.
24 ECCERR_INJ_ALL	**ECC Error Inject For All Address** This ECCERR_INJ_ALL bit determines if ECC error inject happens on all Flash read. 0b - ECC error inject only when reading address matches ECCERR_INJ_ADDR. 1b - ECC error inject when reading any Flash address.
23 - 16 PRESCALER	**Prescaler Configuration** These PRESCALER bits are used for EFM operation time control. Please use the following equation to calculate the PRESCALER value: PRESCALER = slow bus clock frequency / 2 Round up if not divisible. The unit of slow bus clock frequency is Mhz. For example, if slow bus clock frequency is 50Mhz, PRESCALER value should be 25. Please re-configure PRESCALER once the frequency is changed.
15 PREFETCH_EN	**Prefetch Enable** This PREFETCH_EN bit determines whether enable Flash data preftech. 0b - Disable prefetch. 1b - Enable prefetch.
12 WE	**Write Enable** This WE bit determines whether Flash is writable. Write Flash when WE is 0 causes bus error and EFM_STATUS[CI_WEERR] flag is set at the same time. 0b - Disable Flash write. 1b - Enable Flash write.
11 - 8	**Read Wait State**

Yuntu Microelectronics

Field	Function
RWS	These RWS bits determine clock cycles number that needs to be waited to get the Flash data out after Flash receives the read access signal. Please use the following equation to calculate the RWS value : (fast bus clock frequency / (RWS + 1)) <= 50MHz The unit of fast bus clock frequency in the equation is Mhz. For example when fast bus clock is 100Mhz, software can set RWS=0x1 to get best performance. But if read read Flash failed, you can set RWS=0x2 and read Flash again. Please re-configure RWS value once the fast bus clock frequency is changed. **NOTE**. RWS should be reconfigured before fast bus clock switching to a higher frequency.
5 CMD_VERIFY_EN	**Command Verify Read Enable** This CMD_VERIFY_EN bit determines whether to do verify read after erase or program operation during executing a command. - After each program cycle, if CMD_VERIFY_EN=1, EFM should read the programmed address and compare the read data with program data, if not equal, set EFM_STATUS[FAIL] flag to 1. - After each erase cycle, if CMD_VERIFY_EN=1, EFM should read the erased addresses and compare read data with FF, if read data are not all FF, set EFM_STATUS[FAIL] flag to 1. 0b - Disable verify read after erase or program operation. 1b - Enable verify read after erase or program operation
3 ACCERRIE	**Access Error Interrupt Enable** This ACCERRIE bit determines whether the interrupt will be generated if ACCERR flag is asserted. 0b - Interrupt is not generated if ACCERR flag is asserted 1b - Interrupt is generated if ACCERR flag is asserted
2 UNRECOVERRIE	**Unrecovery Error Interrupt Enable** This UNRECOVERRIE bit determines whether the interrupt will be generated if UNRECOVERR flag is asserted. 0b - Interrupt is not generated if UNRECOVERR flag is asserted 1b - Interrupt is generated if UNRECOVERR flag is asserted
1 RECOVERRIE	**Recovery Error Interrupt Enable** This RECOVERRIE bit determines whether the interrupt will be generated if RECOVERR flag is asserted. 0b - Interrupt is not generated if RECOVERR flag is asserted 1b - Interrupt is generated if RECOVERR flag is asserted
0	**Command Done Interrupt Enable**

Yuntu Microelectronics

Field	Function
DONEIE	This DONEIE bit determines whether the interrupt will be generated if DONE flag is asserted. 0b - Interrupt is not generated if DONE flag is asserted 1b - Interrupt is generated if DONE flag is asserted

3.2.1.2 EFM STS Register

Offset: 4h

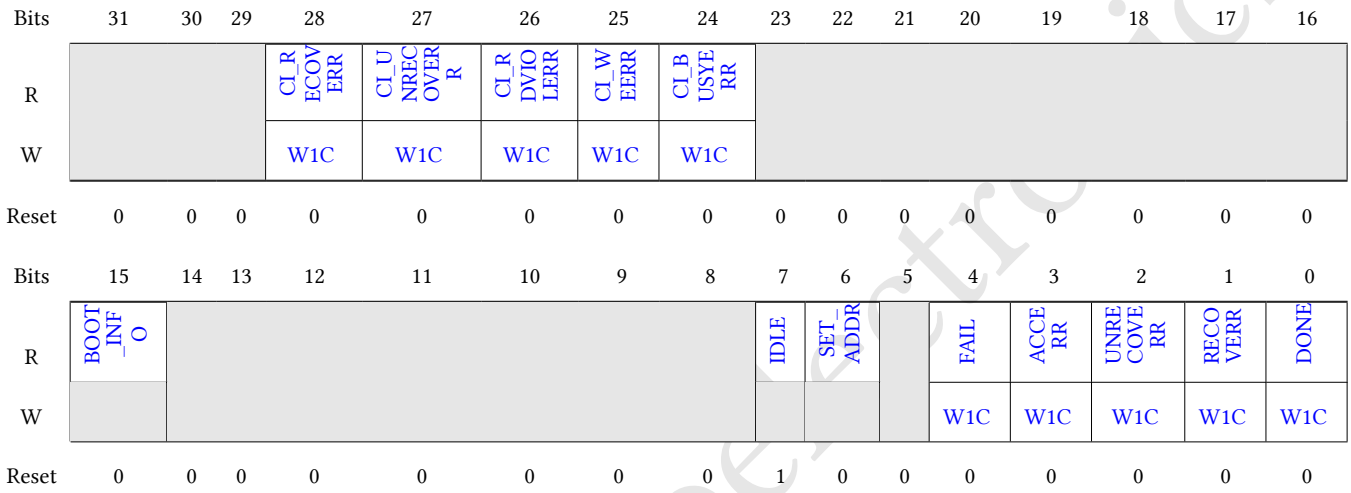


Table 3.4: EFM STS Register Description

Field	Function
28 CI_RECOVERR	**Core Interface Read Recovery Error Status** 1-bit ECC error is detected during bus read will set CI_RECOVERR flag and generate a ECC error interrupt when RECOVERRIE is enabled. 0b - Recovery ecc error is not detected during read access. 1b - 1-bit ECC error is detected during read access.
27 CI_UNRECOVERR	**Core Interface Read Unrecovery Error Status** More than 2-bits ECC error is detected during read access, which causes bus error and sets CI_UNRECOVERR flag. 0b - No unrecovery ECC error during read access. 1b - 2-bits ECC error or multi-bits ECC error is detected during read access.
26	**Bus Read Violation Error Status**

Yuntu Microelectronics

Field	Function
CI_RDVIOLERR	A bus read violation error will set this flag and send a bus error . Read violation error includes below scenarios: - Read CUS_NVR sector when CUS_KEY is not set. - Read HCU_NVR sector when CFG_KEY is not set. - Read BOOT_NVR sector when FAC_KEY is not set. - Read FAC_NVR sector when FAC_KEY is not set. - Read NVR_CFG sectors when CFG_KEY is not set.
25 CI_WEERR	**Bus Write Enable Error Status** Try to write Flash array when EFM_CTRL[WE] is disabled will cause bus error and set CI_WEERR flag.
24 CI_BUSYERR	**Bus Busy Error Status** Try to access Flash array when EFM_CTRL[IDLE] is 0 will cause bus error and set CI_BUSYERR flag.
15 BOOT_INFO	**Boot Information Status** This BOOT_INFO bit indicates whether Flash SoC address remap is enabled. This register bit is reset by System Reset. 0b - Flash SoC address remap is disabled, SoC boots from flash A side. 1b - Flash SoC address remap is enabled, SoC boots from flash B side.
7 IDLE	**Command IDLE Status** This IDLE bit indicates whether Flash is busy. IDLE is 0 in following scenarios: - EFM is executing a command. - EFM is reading NVR array (dummy read command).
6 SET_ADDR	**Set Target Address Status** A write access to target address is needed before launching a command. The write access operation will set SET_ADDR flag. **NOTE**. Launching command when SET_ADDR=1 avoid access error caused by non-sequential scheduling of instructions.
4 FAIL	**Command FAIL Status** This FAIL bit indicates whether the current command is failed. Command failure includes below scenarios: - program verify check failed. - erase verify check failed. - Manual erase-retry Verify-Read check failed
3	**Command Access Error Status**

Yuntu Microelectronics

Field	Function
ACCERR	This ACCERR bit indicates whether there is an access error is detected. Set this bit will generate a access error interrupt when ACCERRIE is enabled. Access error scenarios refer to Flash Abnormal Operation. 0b - Access error is not detected 1b - Access error is detected
2 UNRECOVERR	**Command Un-recovery Error Status** This UNRECOVERR bit indicates whether more than 2-bit ECC error is detected during command verify read stage. Set this bit will generate a ECC error interrupt when UNRECOVERRIE is enabled. 0b - No un-recovery ECC error is detected in command verify read stage. 1b - 2-bit or multi-bit ECC error find in command verify read stage.
1 RECOVERR	**Command Recovery Error Status** This RECOVERR bit indicates whether there is a recovery ecc error is detected during command verify read stage.Set this bit will generate a ECC error interrupt when RECOVERRIE is enabled. 0b - No recovery ECC error is detected in command verify read stage. 1b - 1-bit ECC error is detected in command verify read stage.
0 DONE	**Command Done Status** This DONE bit indicates whether the current command is completed.Set this bit will generate a ECC error interrupt when DONEIE is enabled. 0b - Current command is in progress. 1b - Current command is completed.

Yuntu Microelectronics

3.2.1.3 EFM CMD Register

Offset: 8h

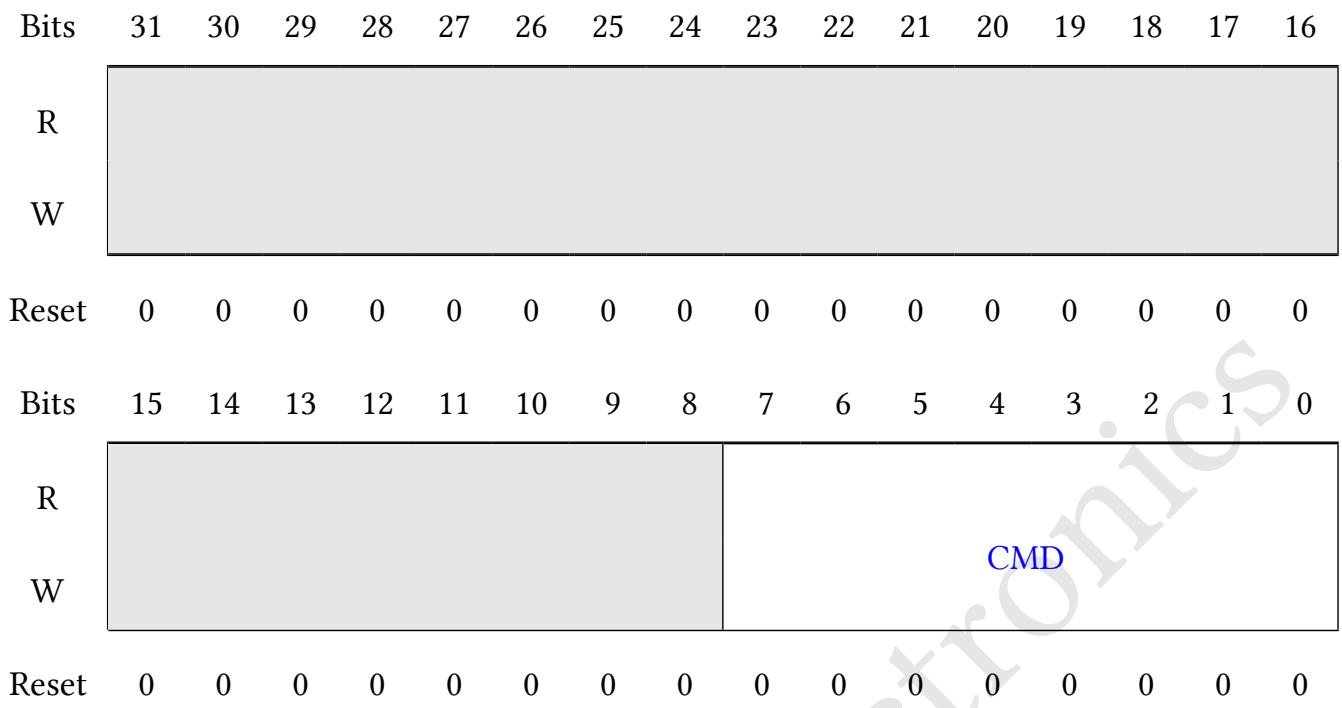


Table 3.5: EFM CMD Register Description

Field	Function
7 - 0 CMD	**Command Control** These CMD bits determine which commands are available.

Command	Code	Description	CMD_VERIFY_EN=1
PROG64	0x01	Program 64bits data	Do verify check
PROG256	0x02	Program 256bits data	Do verify check
ERASE_SECTOR	0x10	Erase Sector	Do verify check
ERASE_BLOCK	0x11	Erase Block	Do verify check
ERASE_RETRY_AUTO	0x13	Auto Erase-Retry	Ignore
ERASE_RETRY_MAN	0x14	Manual Erase-Retry	Ignore
ERASE_RETRY_VREAI	0x15	Manual Verify-Read.	Ignore
LOAD_AES_KEY	0x20	Load AES Key	Ignore
BOOT_SWAP	0x30	Boot Swap	Ignore

NOTE. ERASE_RETRY_VREAD command must be launched after a ERASE_RETRY_MAN command.

NOTE. Each LOAD_AES_KEY command loads 256-bit key data from Flash HCU_NVR array to HCU module. HCUcan be configured to use 128-bit or 192-bit or 256-bit key data.

3.2.1.4 EFM CMD_PROT Register

Offset: ch

Yuntu Microelectronics

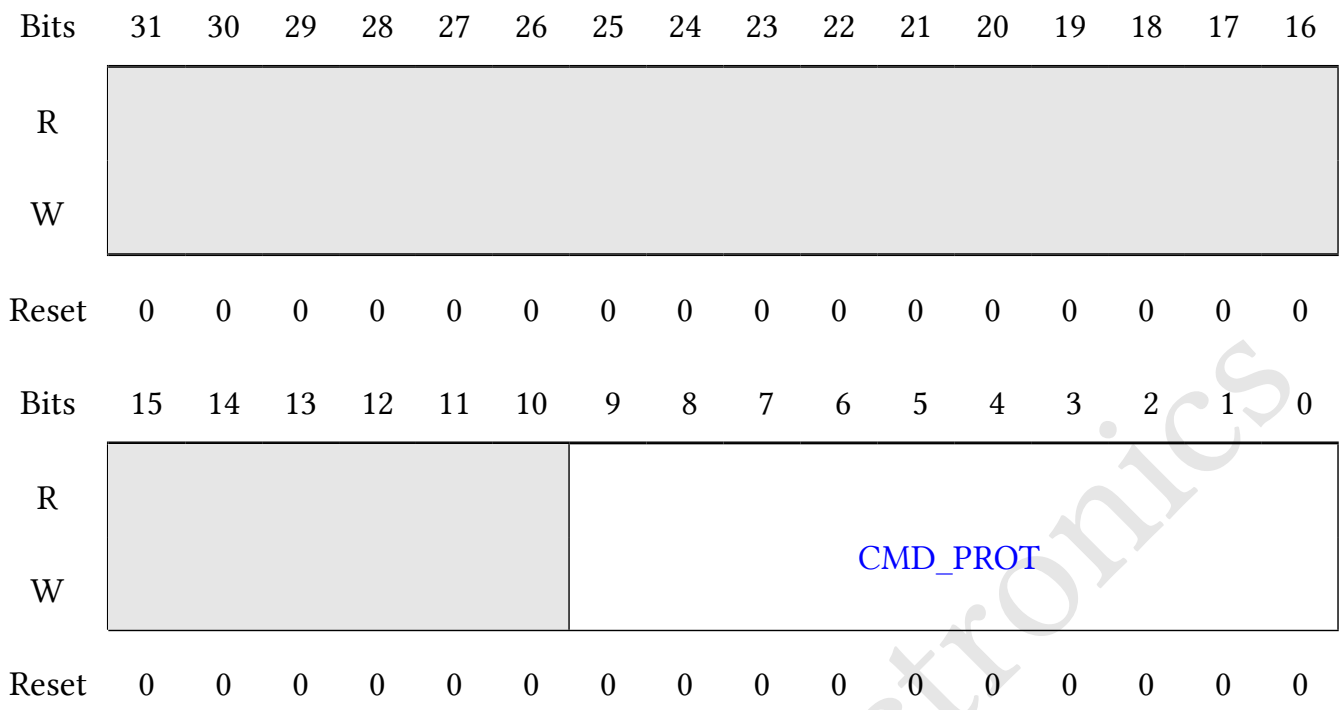


Table 3.6: EFM CMD_PROT Register Description

Field	Function
9 - 0 CMD_PROT	**Command Protect** These CMD_PROT is used to disable commands. Software can write 1 to value 0 bit of CMD_PROT, but can not write 1 to a value 0 bit. The following table gives the map between commands and CMD_PROT bits: Command / CMD_PROT bit PROG64 / 0 PROG256 / 1 ERASE_SECTOR / 2 ERASE_BLOCK / 3 ERASE_RETRY_AUTO / 5 ERASE_RETRY_MAN / 6 ERASE_RETRY_VREA / 7 LOAD_AES_KEY / 8 BOOT_SWAP / 9 For example, if CMD_PROT[8]=1, EFM can not launch LOAD_AES_KEY command.

3.2.1.5 EFM TIMING1 Register

Offset: 10h

Yuntu Microelectronics

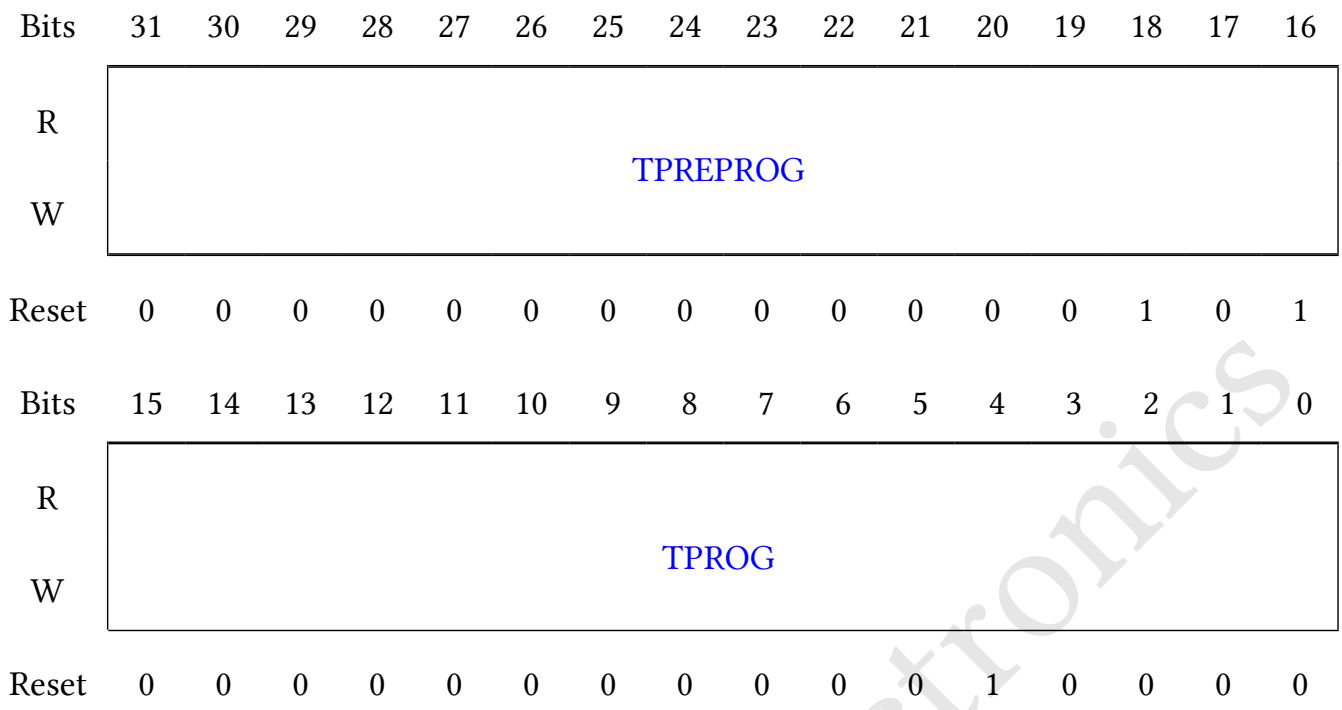


Table 3.7: EFM TIMING1 Register Description

Field	Function
31 - 16 TPREPROG	**Timing of Pre-program** These TPREPROG bits indicate the pre-program pulse time. The valid pre-program pulse time is 2.5us to 3.5us. The time unit of TPREPROG is 0.5us. So that the TPREPROG value range is 5 to 7. For example, if TPREPROG value is 6, the Flash array pre-program pulse time is 6*0.5us=3us.
15 - 0 TPROG	**Timing of Program** These TPROG bits indicate the program pulse time. The valid program pulse time is 8us to 10us. The time unit of TPROG is 0.5us. So that the TPROG value range is 16 to 20. For example, if TPROG value is 16, the Flash array program pulse time is 16*0.5us=8us.

Yuntu Microelectronics

3.2.1.6 EFM TIMING2 Register

Offset: 14h

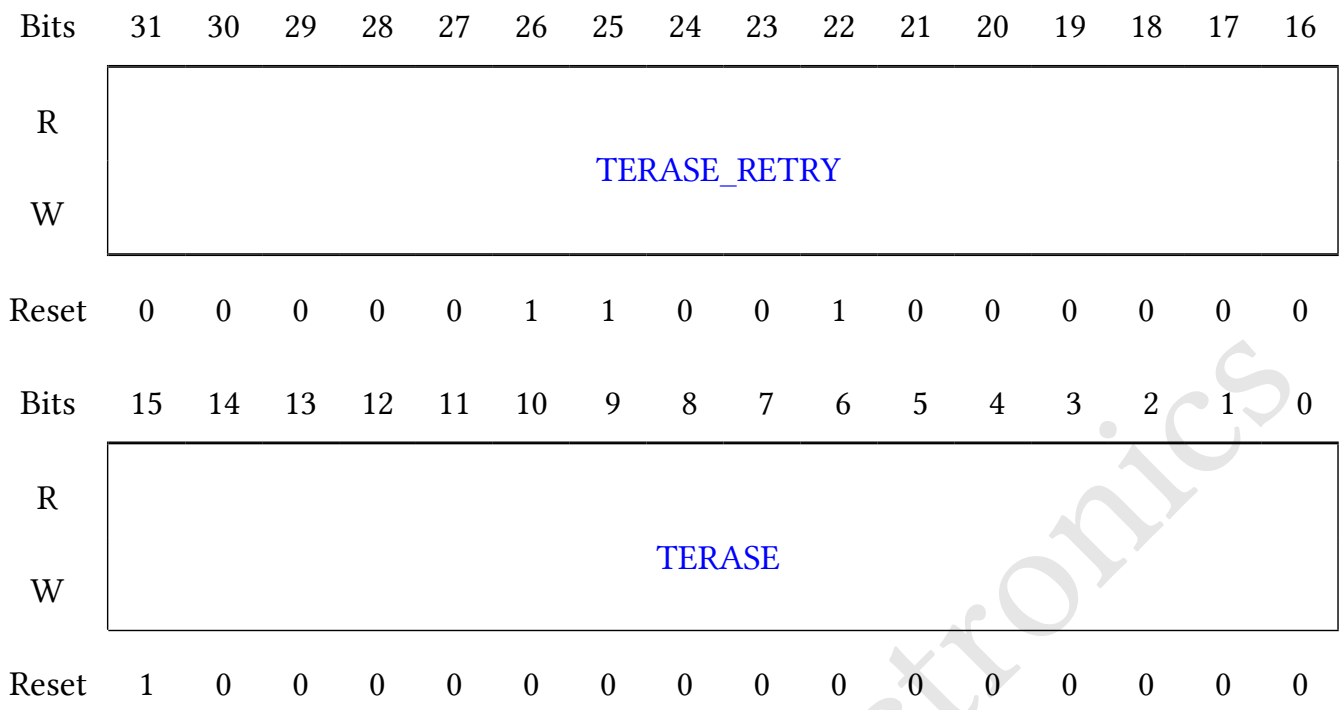


Table 3.8: EFM TIMING2 Register Description

Field	Function
31 - 16 TERASE_RETRY	**Timing of Erase Retry** These TERASE_RETRY bits indicate the erase-retry pulse time. The valid erase-retry pulse time is 0.8ms to 1ms. The time unit of TERASE_RETRY is 0.5us. So that the TERASE_RETRY value range is 1600 to 2000. For example, if TERASE_RETRY value is 1600, the Flash array erase-retry pulse time is 1600*0.5us=800us =0.8ms. **NOTE**. Both Manual and Auto erase-retry are sector erase for Main array.
15 - 0 TERASE	**Timing of Erase** These TERASE bits indicate the single erase times. The valid single erase pulse time is 16ms to 20ms. The time unit of TERASE is 0.5us. So that the TERASE value range is 32000 to 40000. For example, if TERASE value is 32000, the Flash array single erase pulse time is 32000*0.5us=16000us =16ms. **NOTE**. Both ERASE_SECTOR and ERASE_BLOCK command use this single erase time.

3.2.1.7 EFM ECCERR_INJECT_MSK_0 Register

Offset: 18h

Yuntu Microelectronics

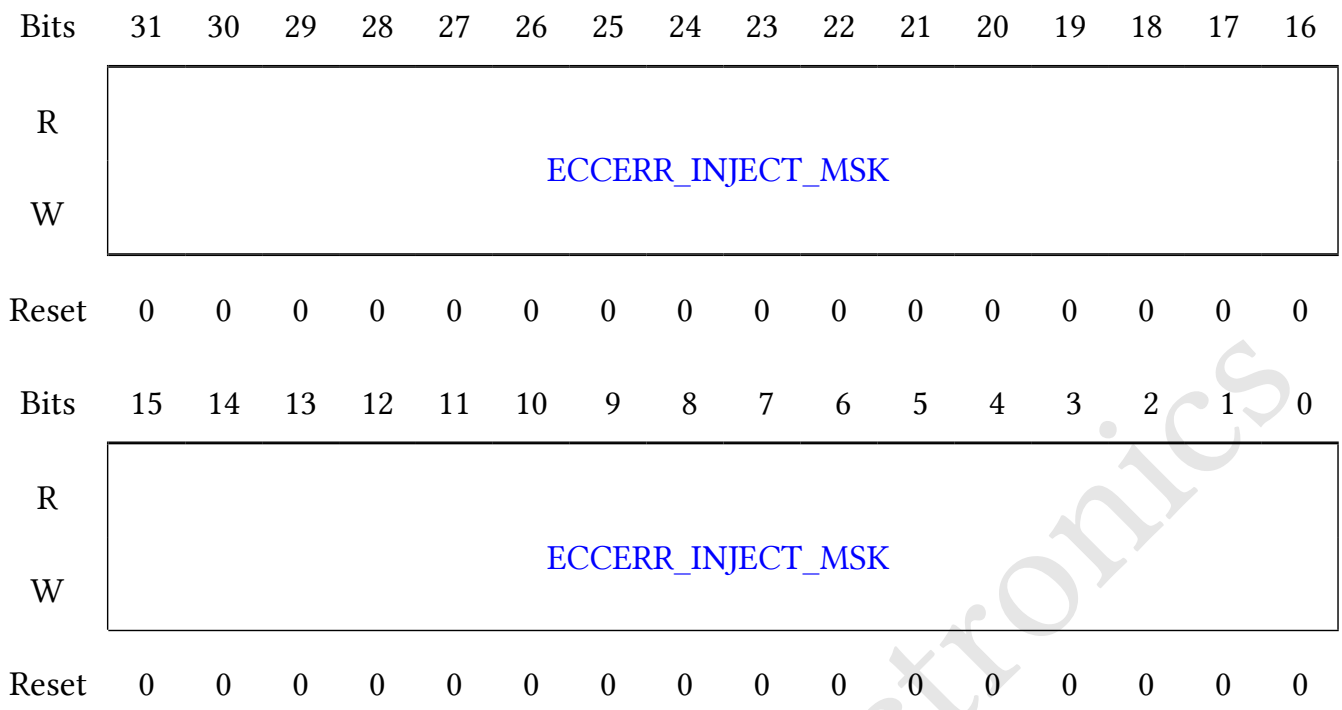


Table 3.9: EFM ECCERR_INJECT_MSK_0 Register Description

Field	Function
31 - 0 ECCERR_INJECT_MSK	**ECC Error Inject Mask Low 32-bits** This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data[31:0] from the target Flash should be inverted or remain unmodified. For example, 0x00000100 means Flash read data bit[12] is inverted. This register is valid only when ECCERR_INJ_EN is 1.

Yuntu Microelectronics

3.2.1.8 EFM ECCERR_INJECT_MSK_1 Register

Offset: 1ch

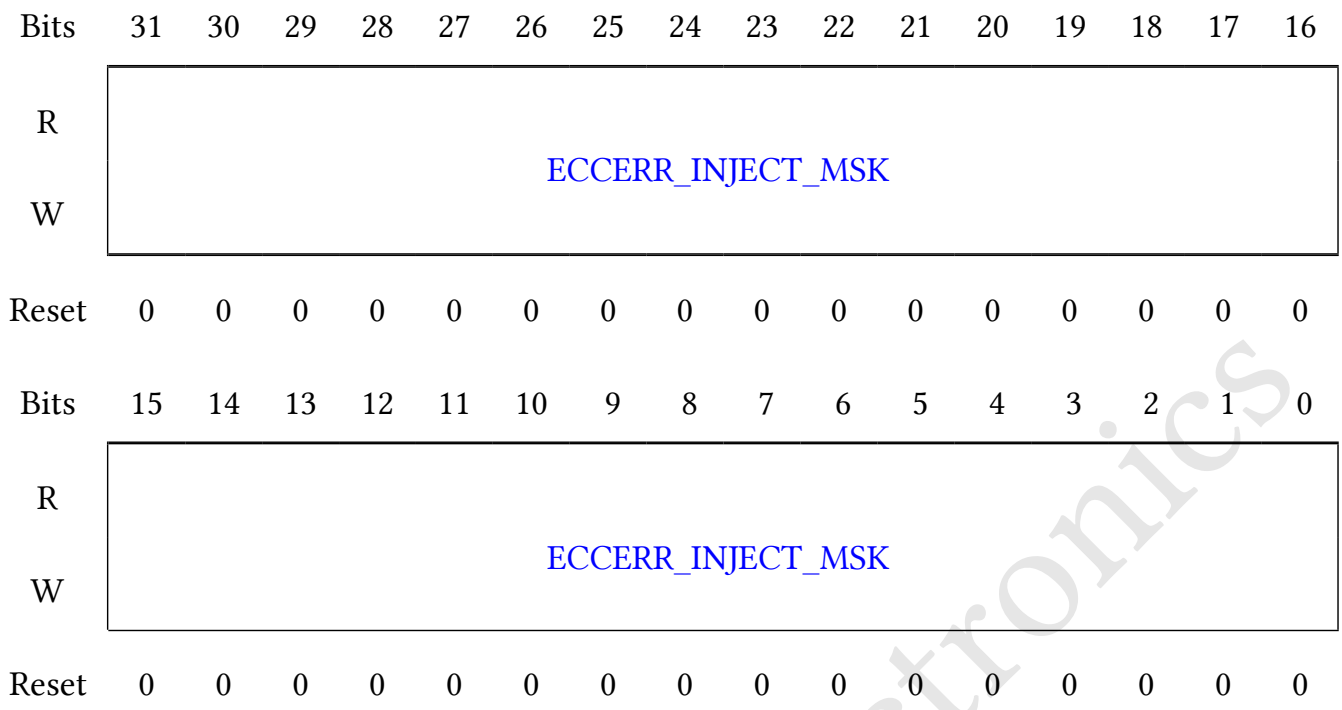


Table 3.10: EFM ECCERR_INJECT_MSK_1 Register Description

Field	Function
31 - 0 ECCERR_INJECT_MSK	**ECC Error Inject Mask High 32-bits** This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data[63:32] from the target Flash should be inverted or remain unmodified. For example, 0x00000002 means Flash read data bit[33] is inverted. This register is valid only when ECCERR_INJ_EN is 1.

Yuntu Microelectronics

3.2.1.9 EFM ECCERR_INJECT_ADDR Register

Offset: 20h

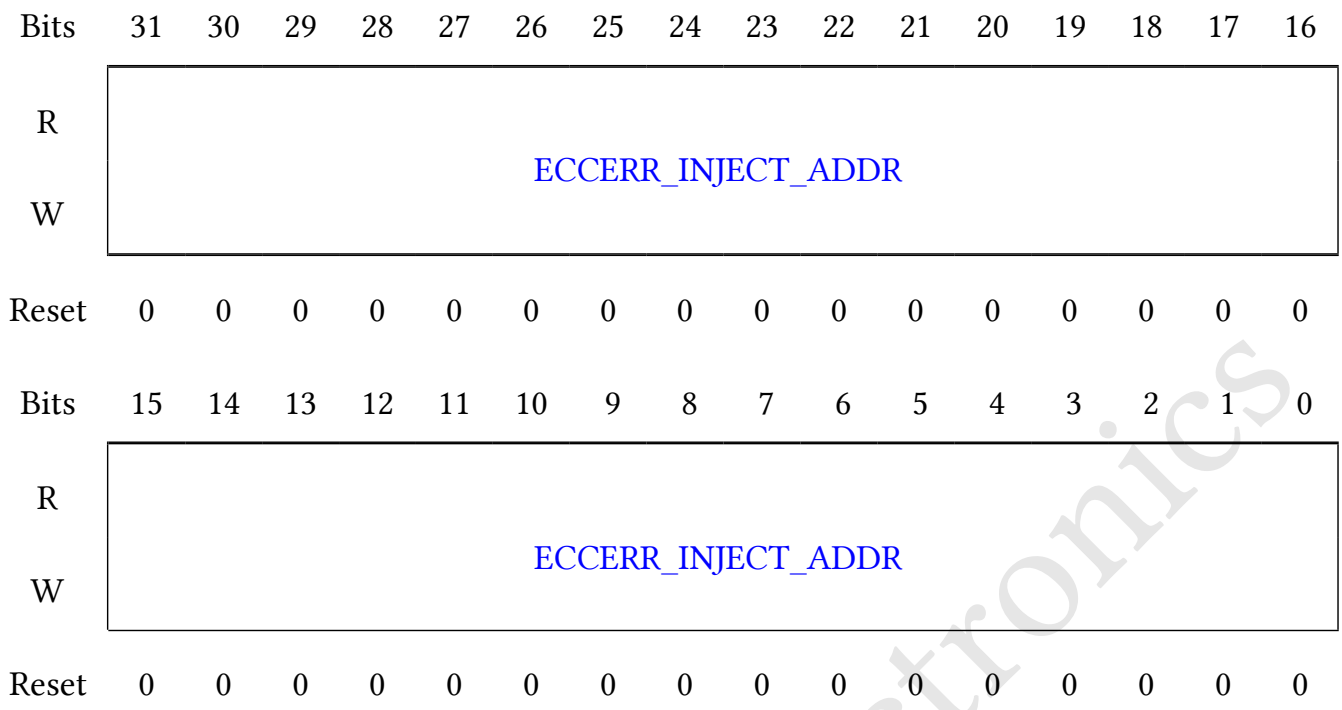


Table 3.11: EFM ECCERR_INJECT_ADDR Register Description

Field	Function
31 - 0 ECCERR_INJECT_ADDR	**ECC Error Inject Address** This field defines Flash address which to inject ECC error. This register is valid only when ECCERR_INJ_ALL is 0 and ECCERR_INJ_EN is 1.

Yuntu Microelectronics

3.2.1.10 EFM ADDR_PROT Register

Offset: 24h + n * 4h

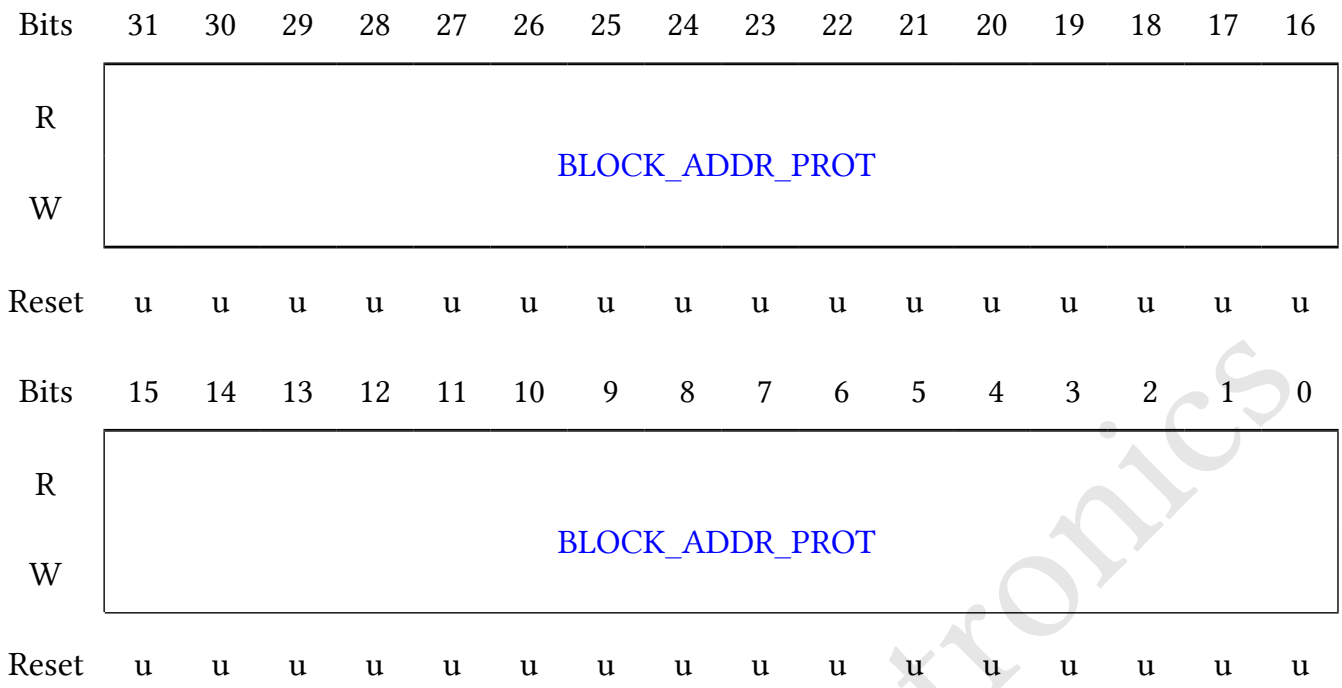


Table 3.12: EFM ADDR_PROT Register Description

Field	Function
31 - 0 BLOCK_ADDR_PROT	**Block Address protect** The address area of each Flash is divided into 32 regions. Each region is protected by one bit of ADDR_PROT. The address area of each region is 32KB for PFlash and 8KB for DFlash. If any bit is set to 0, the corresponding region is under protection (read only). During EFM initialization, ADDR_PROT registers load initial values from CUS_NVR array when CUS_NVR has programmed initial ADDR_PROT values with correct tag. After initial done, software can write 0 to value 1 bit of ADDR_PROT, but can not write 1 to value 0 bit.

Yuntu Microelectronics

3.2.1.11 EFM ECC_ERR_ADDR Register

Offset: 30h

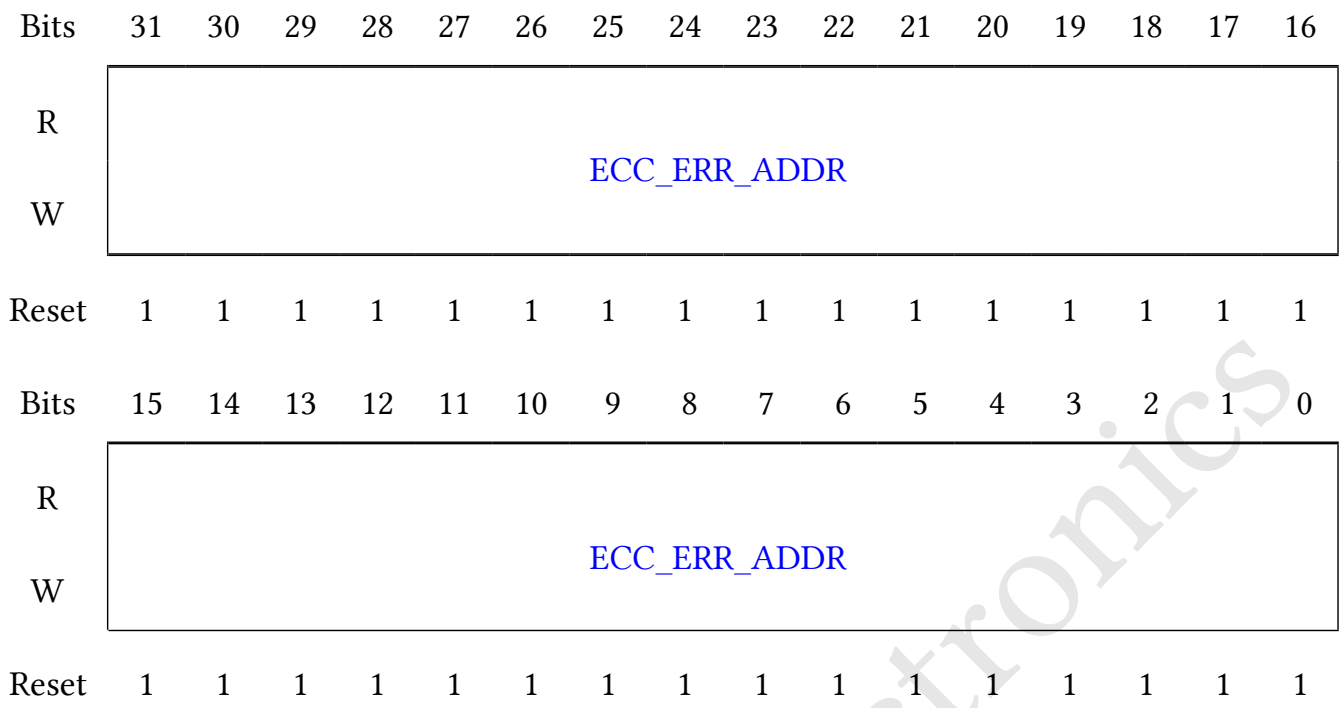


Table 3.13: EFM ECC_ERR_ADDR Register Description

Field	Function
31 - 0 ECC_ERR_ ADDR	**Ecc Error Address** These bits records the first detected ECC error address. Following ECC error addresses are not recorded. When detect an ECC error, hardware sets ECC error flag (CI_RECOVERR or CI_UNRECOVERR or RECOVERR or UNRECOVERRR). If ECC_ERR_ADDR is currently default, ECC_ERR_ADDR is written with error address at the same time. ECC_ERR_ADDR is cleared by writing 1 to the corresponding error flag. For example, if ECC_ERR_ADDR records CI_UNRECOVERR address, only write 1 to CI_UNRECOVERR can clear ECC_ERR_ADDR, writing 1 to CI_RECOVERR or RECOVERR or UNRECOVERRR can not clear ECC_ERR_ADDR.

Yuntu Microelectronics

3.2.1.12 EFM CMD_UNLOCK Register

Offset: 200h

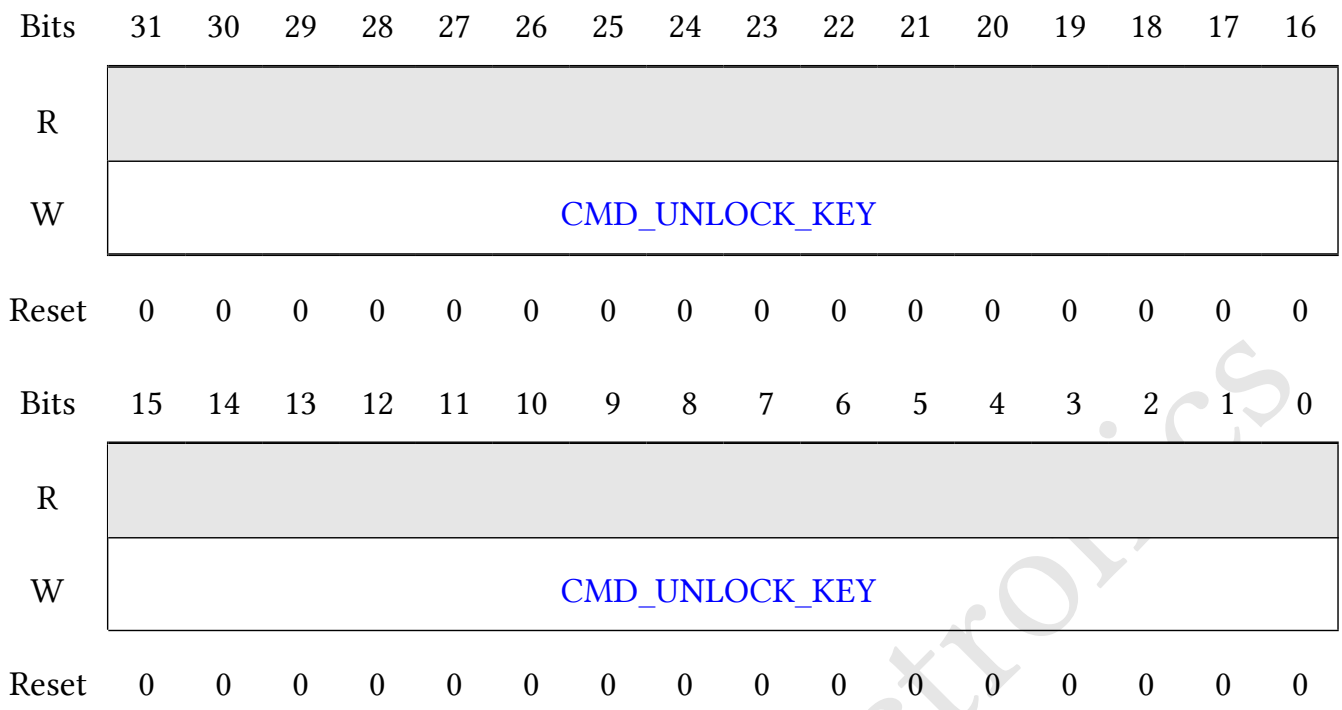


Table 3.14: EFM CMD_UNLOCK Register Description

Field	Function
31 - 0 CMD_UNLOCK_KEY	**EFM Command Unlock Key** Please write 0xfd9573f5 before launching a command

Yuntu Microelectronics

3.2.1.13 EFM CUS_KEY Register

Offset: 204h

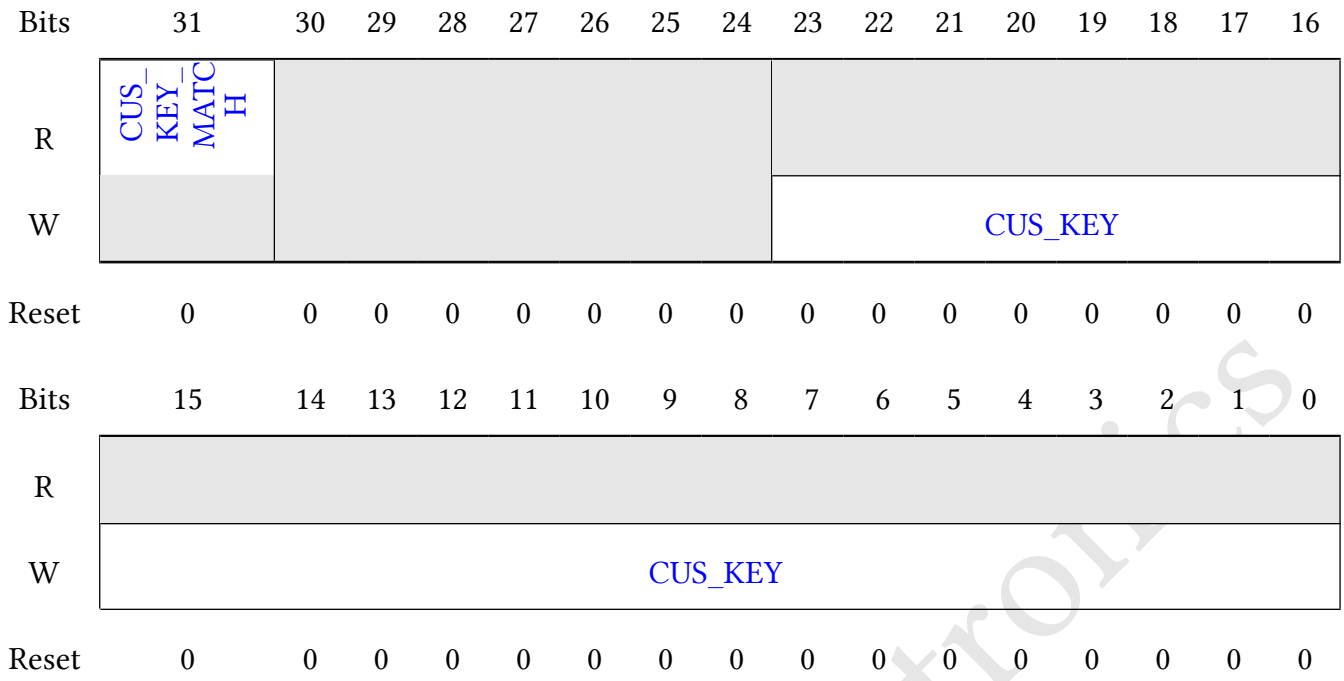


Table 3.15: EFM CUS_KEY Register Description

Field	Function
31 CUS_KEY_ MATCH	**Customer Key Match Status** This CUS_KEY_MATCH bit indicates whether Customer Key is matched. 0b - Customer Key is not matched 1b - Customer Key is matched
23 - 0 CUS_KEY	**EFM Customer Key** Following are Customer Key protected operations, please write CUS_KEY with correct value before doing these operations: - Read or erase or program CUS_NVR - Erase HCU_NVR array - Erase SHA_NVR array Default Customer Key value is 0x4dff32 and the Customer Key can be re-defined by programming the first address of CUS_NVR, it takes effect after EFM is reseted.

3.2.1.14 EFM CTRL_OVRD Register

Offset: 300h

Yuntu Microelectronics

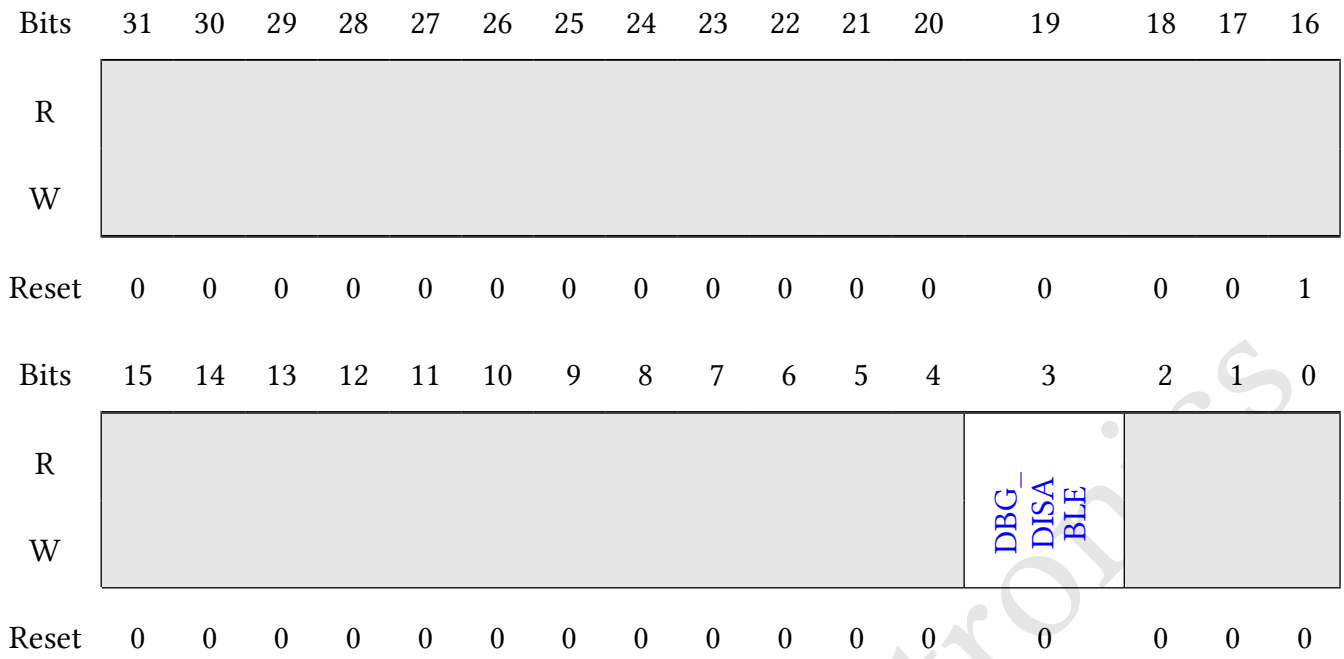


Table 3.16: EFM CTRL_OVRD Register Description

Field	Function
3 DBG_DISABLE	**Debugger Disable** This bit determines whether the debugger is disabled. 0b - Debugger is enabled 1b - Debugger is disabled

3.3 Functional Description

3.3.1 EFM Initial

STS[IDLE] is 0 after reset, it is set to 1 when initially done. Software should check register STS[RECOVERR], STS[UNRECOVERR] and ECC_ERR_ADDR for ECC error information detected during initial flow.

3.3.2 Setting Timing Control Register

Before issuing any program or erase command, it is necessary to configure the appropriate values that are needed during a program or erase operation to TIMING1[TPREPROG], TIMING1[TPROG], TIMING2[TERASE], TIMING2[TERASE_RETRY] and CTRL[PRESCALER] based on slow bus clock (APB). Configure CTRL[RWS] based on fast bus clock (AHB).

3.3.3 Command Operation Flow

Please follow these sequence steps for a command operation. Intermediate writing to EFM registers is not permitted between these steps.

Yuntu Microelectronics

- (a) Read STS[IDLE] bit. If STS[IDLE]==1, the address, data and command are ready to launch a new command sequence. Else if STS[IDLE]==0, the EFM is busy operating a command, please wait.
- (b) Read CMD_PROT register. If the corresponding CMD_PROT bit of the command you want to launch is 0, go to the next step.
- (c) Set CTRL[WE] to 1.
- (d) Write data to target address area. The internal buffers will store the address and data information.
  - For program command the written data will be programmed into Flash array, for other commands it is dummy data and will be ignored.
  - Program command may write one or more data to the 64-bit/256-bit target address area, other commands need to write only once.
  - The target address of sector erase (ERASE_SECTOR, auto or manual ERASE_RETRY) commands must be located in the target sector.
  - The target address of ERASE_BLOCK command must be located in Main array of the target block.
  - The target address of LOAD_AES_KEY command must be located in HCU_NVR array.
  - The target address of BOOT_SWAP command must located in BOOT_NVR.
- (e) Wait until STS[SET_ADDR] register is set high. This means address mapping to the target Flash array is done.
- (f) Clear CTRL[WE] to 0.
- (g) Write CMD_UNLOCK_KEY register with 0xFD9573F5
- (h) Write the valid command code to CMD register. After this step, check the status register:
  - If STS[ACCERR]=1, any illegal operations exist in your command operation flow. The command is not launched.
  - If STS[IDLE] =0, the command is launched and starts executing.
- (i) When command execution is completed, the FLASH controller will set the STS[IDLE] and STS[DONE] flags. If STS[CMD_FAIL], STS[UNRECOVERR] and STS[RECOVERR] flags are all 0, the command operation successes.

It is necessary to clear the STS[DONE] before commencing another command.

Table 3.17: EFM Command Scope

Command	PFlash0/1 Main	DFlash Main	CUS NVR	HCU NVR	SHA NVR	BOOT NVR	OTP NVR
PROG64	N	Y	N	N	N	N	N
PROG256	Y	Y	Y	Y	Y	Y	Y
ERASE_SECTOR	Y	Y	P	P	P	N	N
ERASE_BLOCK	Y	Y	N	N	N	N	N
ERASE_RETRY_AUTO	Y	Y	N	N	N	N	N
ERASE_RETRY_MAN	Y	Y	N	N	N	N	N
ERASE_RETRY_VREAD	Y	Y	N	N	N	N	N
LOAD_AES_KEY	N	N	N	Y	N	N	N
BOOT_SWAP	N	N	N	N	N	Y	N

Yuntu Microelectronics

NOTE. P : Customer Key Protected

3.3.4 Read While Write

There are three blocks in the EFM Module, each consisting of a Flash array with its control logic. When one block is operating a program or erase command, read access to another two blocks is available. Below is the access combination:

Table 3.18: Read While Write

Block	Write PFlash0	Write PFlash1	Write DFlash
Read PFlash0	N	Y	Y
Read PFlash1	Y	N	Y
Read DFlash	Y	Y	N

3.3.5 Boot Swap Operation

By default, register STS[BOOT_INFO] is 0, that is EFM block 0 address is mapped to PFlash0 and block 1 address is mapped to PFlash1.

- When STS[BOOT_INFO] is 0, execute an BOOT_SWAP command and then reset EFM will swap block0 and block1, STS[BOOT_INFO] is set to 1 and the address is re-mapped: access block 0 address will access PFlash1 and access block 1 address will access PFlash0.
- When STS[BOOT_INFO] is 1, execute an BOOT_SWAP command and then reset EFM will switch back the address mapping and clear the STS[BOOT_INFO] flag.

3.3.6 Disable JTAG and SWD

Program 0x5A5A5A5A_5A5A5A5A to bias address 0x8 of CUS_NVR (refer to "EFM Memory Map" table for the base address of CUS_NVR sector), then reset EFM, the connection between chip and debugger will be disabled.

3.3.7 Address Protection

The three registers (ADDR_PROTn, n is 0-2) respectively protect the Main array of PFlash0, PFlash1, DFlash. The address area of each block is divided into 32 regions, for block 0 (PFlash0) and Block 1 (PFlash1), each region is 32KB, but for block 2 (DFlash) a region is 8KB.

The reset value of ADDR_PROTn is 0xffffffff. Once any bit is written to 0, the corresponding region will be protected. That is this region is read-only, any program or erase command to this region is invalid. If any region of a block is protected, the ERASE_BLOCK operation on this block will be failed.

There are two ways to configure ADDR_PROTn:

- Program data to bias address 0x10, 0x18, 0x20 of CUS_NVR (refer to "EFM Memory Map" table for the base address of CUS_NVR sector) to set initial values of ADDR_PROTn, the high 32-bit of programmed data must be 0x5A5A5A5A, and the low 32-bit of programmed data should be the initial value of ADDR_PROTn. The programmed initial values are effective after resetting EFM.
- Write to ADDR_PROTn registers. Writing 0 to ADDR_PROTn is valid but writing 1 to ADDR_PROTn will be invalid.

Yuntu Microelectronics

3.3.8 Manual Erase Retry Operation

Please follow these sequence steps for manual erase-retry operation.

- (a) retry_cnt is defined by software to record the erase times, and set the initial value of retry_cnt to 0.
- (b) Write CTRL[RETRY_CFG] register with retry_cnt value.
- (c) Issue ERASE_RETRY_MAN command (refer to Command Operation Flow for more details).
- (d) Issue ERASE_RETRY_VREAD command (refer to Command Operation Flow for more details).
- (e) Check STS[CMD_FAIL] flag:
  - If STS[CMD_FAIL]==1, write 1 to this register to clear the flag, then go-to step(f);
  - If STS[CMD_FAIL]==0, Manual erase-retry succeed.
- (f) Check erase times:
  - If retry_cnt==20, Manual erase-retry failed.
  - If retry_cnt<20, retry_cnt+1, go-to step(b).

3.3.9 Flash Abnormal Operation

An access error can be detected during a command operation sequence right after writing the CMD register. When an access error is detected (STS[ACCERR] = 1), the command will not be executed. The following illegal operations will set STS[ACCERR] flag:

- Write command to the CMD register without address and data information (STS[SET_ADDR] = 0) is not effective.
- Write a command whose corresponding bit in CMD_PROT is value 1 causing an access error.
- Write a command to the CMD register without setting CMD_UNLOCK.
- Writing a invalid command to the CMD register is not effective.
- Write a new command to the CMD register while the EFM is busy (STS[IDLE] = 0) is not effective, and the command register will not be updated.
- Multiple write addresses of PROG256 operation should be in the same 32 Byte address area.
- Multiple write addresses of PROG64 operation should be in the same 8 Byte address area.
- Program or sector erase the protected address.
- Launch the BLOCK_ERASE command when any address of a block is under protection.
- Erase the HCU_NVR array or SHA_NVR array without appropriate Customer Key.
- Read or erase or program CUS_NVR array without appropriate Customer Key.
- Erase OTP_NVR array.
- Read or erase or program BOOT_NVR array . BOOT_NVR array can only be used in executing BOOT_SWAP command flow.
- Write NVR array in ERASE_BLOCK or erase-retry command flow. ERASE_BLOCK and erase-retry commands are only valid to Main array.

Yuntu Microelectronics

3.4 Glossary

EFM	Embeded Flash Module
PFlash	Program Flash
DFlash	Data Flash
NVR array	Non-volatile Register array
OTP	One-time Programmable
OTA	Over-The-Air
DPD mode	Deep Powerdown mode

Yuntu Microelectronics

Chapter 4

Read-Only Memory (ROM)

4.1 Introduction

Read-Only Memory or ROM, organized as a 64KB on-chip memory map, is an integrated-circuit memory module that contains configuration data. The ROM boot firmware is programmed by YTMicro to support the following four functions:

- Secure boot
- Fast wakeup from Powerdown Mode
- Structural Core Self-Test (SCST)
- System clock and watchdog configuration.

4.1.1 Features

- 64KB on-chip memory map
- The ROM boot firmware is programmed by YTMicro to support the following four functions: Secure boot, Fast wakeup from Powerdown Mode, Structural Core Self-Test (SCST), System clock and watchdog configuration:
  - Run once the chip releases the reset signal of the CM7 core
  - Support the parsing of the configurable BVT (Boot Vector Table) stored in the Flash memory
- Secure boot
  - Work as the root of trust
  - Support CMAC authorization of the customized code (i.e. bootloader, OTA) with the specified secure boot group configuration set in the BVT.(Note: The CMAC authorization is achieved by HCU.)
    - Support decryption of secure boot section configuration structure with the specified HCU AES key in secure boot group configuration
    - Support up to 8 secure boot section configurations (each includes the section start address, length and HCU AES key used for CMAC as well as CMAC stores address pointer)
  - The used HCU hardware user keys for secure boot section configuration structure encryption and CMAC authorization should be pre-programmed to HCU_NVR memory by the Car OEM or Tier-1 using the flash programmer
  - Support strict serial secure boot mode (enabled via BVT)
    - If CMAC authorization fails, the secure boot holds the CM7 core in static mode
  - Support normal serial secure boot mode (by default)

Yuntu Microelectronics

    - If CMAC authorization fails, the CM7 core continues to execute the bootloader and the applications, but all HCU hardware user keys will be disabled or fail to load.
- Fast wakeup from Powerdown Mode
  - Support fast wakeup when recovering from Powerdown mode
  - Reduce the ROM boot time by waiving the execution of secure boot and core test
  - Support for the enabling and configuration of the separated IVT (Interrupt Vector Table), by using the first two words of REGFILE that are retained during the Powerdown mode
- Structural Core Self-Test (SCST)
  - YTMicro provides the code for the CM7 core test, for the purpose of latent fault detection, which is an important safety mechanism of the CM7 core to aid the FuSa ASIL-D
- System watchdog configuration
  - Support watchdog enable/disable function and timeout configuration for CM7 main core (The default clock source of WDG in IPC will be selected to SIRC/4)

4.2 ROM Boot Code Functions and Flowchart

This section introduces the boot code functions and work flowchart.

4.2.1 Use the CUS_KEY As Root of Trust

User applications can use the CUS_KEY as the root of trust, it can be programmed by the customer with the default CUS_KEY(0x4DFF32), the readback also requires the correct/matched CUS_KEY, so it's secure enough to work as the root of trust.

The potential risk that should be highlighted here is: cannot recover the NVR erase/program/read operations if the user forgot the modified CUS_KEY, while the PFlash and DFlash erase/program/read operations will not be affected and can also work as normal.

See "EFM CUS_KEY Register" for more information.

4.2.2 The ROM Boot Work Flowchart

The figure below shows the complete work flowchart of the ROM boot code:

Yuntu Microelectronics

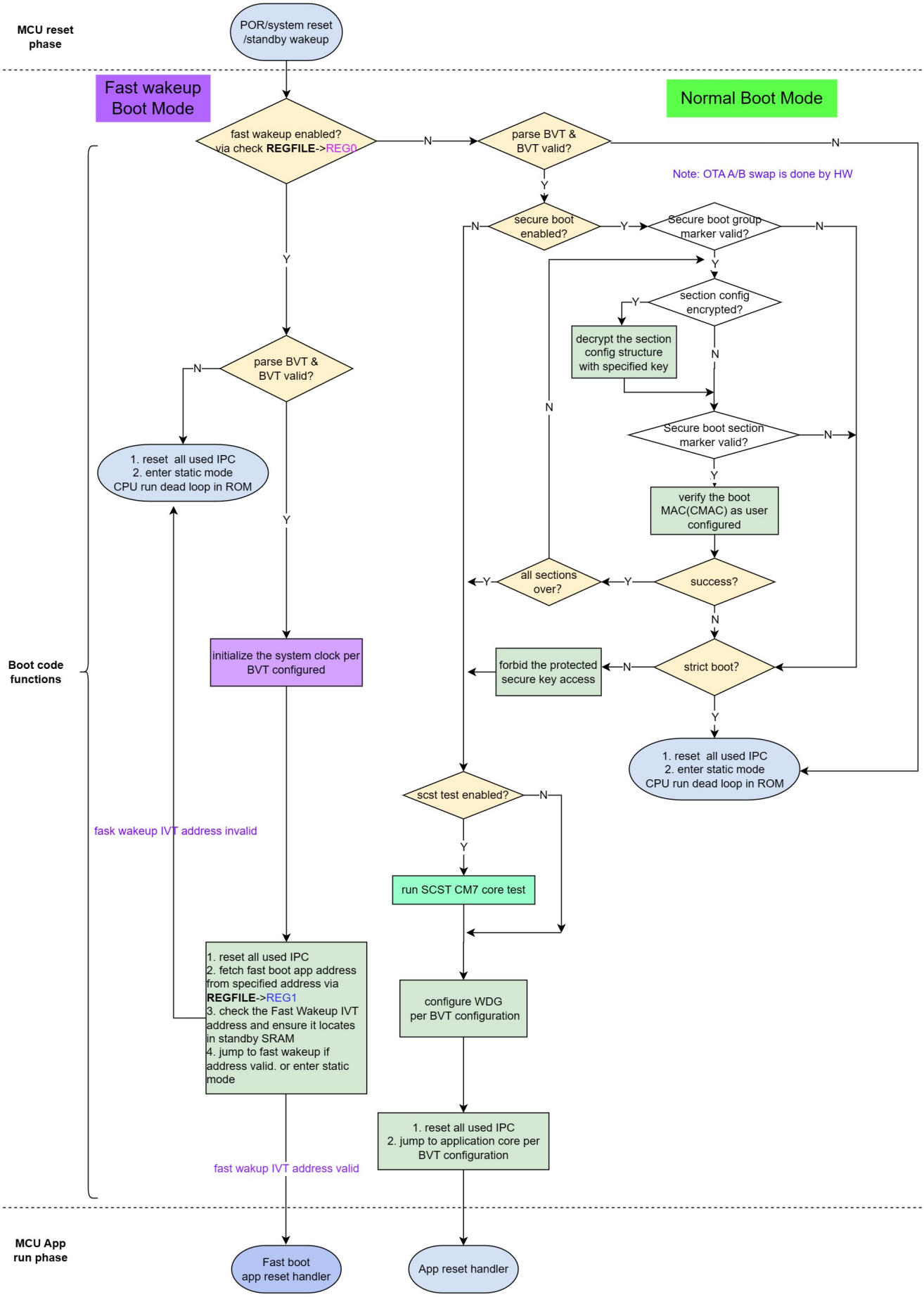


Figure 4.1: The ROM Boot Work Flowchart

Compared to the normal boot mode, the fast wakeup boot mode does not execute the secure boot and SCST, so it can save boot time.

4.3 Functional Descriptions

This section describes the detailed functionalities of the ROM boot codes.

4.3.1 Structure of BVT (Boot Vector Table)

BVT is used by ROM boot code to identify the user application image and contains many key configuration information about the application image.

4.3.2 Location of BVT

BVT is located at fixed Flash memory space with 1MB interval, for example 0x0200_0000, and 0x0210_0000.

BVT occupies 108 bytes, but it will use 2KB (0x800) PFlash due to the PFlash sector size is 2KB.

4.3.3 Boot Vector Table

Table 4.1: BVT Contents

Address Offset	Size in Bytes	Content	Comments
00h	4	Image Vector Table Marker	It is a magic number which marks the valid of a Boot Vector Table location. Its value must be 0xA55A_A55Ah.
04h	4	Boot Configuration Word	Configuration word that allows the user to select the various configuration in which device can be booted. Details are mentioned in the next section.
08h	4	Secure Boot Group Config Address	The start address of memory secure boot group configuration, a valid config group is identified by a fixed marker and consists of a number of secure boot section config information.
0Ch	4	Address of LC Configuration	Address of Configuration word that allows User to Advance LC. Details are mentioned below.
10h	4	CM7_0_main application Start Address	Boot address of CM7_0_main application in code flash area. It must honor core VTOR register alignment restrictions. CM7_0 is always the first user application core after boot (reset caused by either destructive/functional reset or powerdown mode wakeup)
14h	4	CM7_0_secondary application Start Address	Boot address of CM7_0_secondary application in code flash area. It must honor core VTOR register alignment restrictions. This filed is ignored if Lockstep configuration is enabled.
18h	4	CM7_1_main application Start Address	Boot address of CM7_1_main application in code flash area. It must honor core VTOR register alignment restrictions.
1Ch	4	CM7_1_secondary application Start Address	Boot address of CM7_1_secondary application in code flash area. It must honor core VTOR register alignment restrictions. This filed is ignored if Lockstep configuration is enabled.
20h	4	CM7_2_main application Start Address	Boot address of CM7_2_main application in code flash area. It must honor core VTOR register alignment restrictions.
24h	4	CM7_2_secondary application Start Address	Boot address of CM7_2_secondary application in code flash area. It must honor core VTOR register alignment restrictions. This filed is ignored if Lockstep configuration is enabled.
28h	4	APP_WDG_TIMEOUT	Set the APP WDG watchdog timeout value. The WDG is 3MHz clock from 12MHz SIRC divided by 4 for timeout count, so for example, set this value to 3000(0xBB8), the timeout period will be 1ms.
2Ch	20	Reserved	Reserved bytes for 32 byte alignment (the PFlash program size)
40h-0x3FF	TBD	Reserved for secure boot group and section config as well as CMAC result storge	The left address space of this sector (2KB) can be used to store the secure boot config group info/structure and secure boot section config structure as well as the CMAC results (16byte alignment required) of every secure boot section.

4.3.4 Boot Configuration Word Description

The BCW(Boot Configuration Word) layout is listed in the table below:

Yuntu Microelectronics

Table 4.2: Boot Configuration Word Layout

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved					CPDIVS	ROM_BOOT_SCST_EN	APP_WDG_EN	BOOT_SEQ_STRICT	BOOT_SEQ_EN	CM7_2_S_EN	CM7_2_M_EN	CM7_1_S_EN	CM7_1_M_EN	CM7_0_S_EN	CM7_0_M_EN

The detailed configuration fields are listed in the table below:

Table 4.3: The Detailed Configuration Fields

Bit field	Descriptions	
31-11	Reserved for future extension	
10	CPDIVS	Core Platform Clock Divider Value configuration: 0 - Divide by 1 1 - Divide by 2 This configuration determines the CPU core clock frequency for ROM boot code. After system reset, the FIRC is selected as the system reference clock by default. Tips: Using the 96MHz FIRC as the core platform clock instead of 48MHz FIRC (default)can accelerate the ROM boot execution, and reduce the time.
9	ROM_BOOT_SCST_EN	Control whether the CM7 application main core SCST is run or not in the ROM boot. 0 - Disable 1 - Enable Notes: If the CM7 core SCST is enabled and expects the tests to be called in application, the last 4KB OCRAM memory (0x0x2005_F000 to 0x2005_FFFF) must be kept in application project via not allocating the address space in its linker file.
8	APP_WDG_EN	Control the application main core's watchdog–WDG0 enablement before control is passed to application core(s). 0 - Disable 1 - Enable Boot code initializes the WDG0 before enabling the application cores if this bit is set 1b1. Notes: By default, WDG0 is disabled. If set this bit, application codes must take care of WDG0 feed dog to avoid timeout reset and detailed timeout period is configured via BVT -> APP_WDG_TIMEOUT at 0x28h offset.
7	BOOT_SEQ_STRICT	When BOOT_SEQ = 1, select normal/strict serial secure boot mode, ignored if BOOT_SEQ = 1 0 - Normal serial secure boot mode 1 - Strict serial secure boot mode
6	BOOT_SEQ	Secure boot mode configuration: 0 - Disable serial secure boot mode, normal boot mode 1 - Enable serial secure boot mode
5	CM7_2_S_EN	CM7_2 secondary core clock gate configuration after boot, ignored if CM7_2 is configured to be DCLS (Dual Core Lock Step) 0 - Disable 1 - Enable Notes: this bit is reserved on this device for future use.
4	CM7_2_M_EN	CM7_2 main core clock gate configuration after boot 0 - Disable 1 - Enable Notes: this bit is reserved on this device for future use.
3	CM7_1_S_EN	CM7_1 secondary core clock gate configuration after boot, ignored if CM7_1 is configured to be DCLS(Dual Core Lock Step) 0 - Disable 1 - Enable Notes: this bit is reserved on this device for future use.
2	CM7_1_M_EN	CM7_1 main core clock gate configuration after boot 0 - Disable 1 - Enable Notes: this bit is reserved on this device for future use.
1	CM7_0_S_EN	CM7_0 secondary core clock gate configuration after boot, ignored if CM7_0 is configured to be DCLS (Dual Core Lock Step) 0 - Disable 1 - Enable Notes: this bit is reserved on this device for future use.
0	CM7_0_M_EN	CM7_0 main core clock gate configuration after boot 0 - Disable 1 - Enable Notes: To ensure the application codes are executed as expected, this bit must be programmed to 1'b1.

Yuntu Microelectronics

4.3.5 Secure Boot

If secure boot is enabled with proper configuration of the boot configuration word located BVT, the secure boot group and section configurations must be configured correctly.

After any system reset, the ROM boot code checks the secure boot configuration (**BVT->BCW[BOOT_SEQ_EN]**). If it's enabled, the ROM boot code validates the secure boot group configuration via comparing the fixed Config Group Marker, and then decrypt the secure boot section configuration structures (if encrypt flag is set in secure boot group configuration structure) and further verify and authorize the secure boot section with the generated CMAC and the HCU AES key specified in secure boot section configuration structures.

4.3.5.1 Secure Boot Mode

The ROM boot code supports only serial secure boot mode and does not support parallel secure boot mode.

Normal serial secure boot mode (BVT->BCW[BOOT_SEQ_STRICT] = 0)

If the secure boot fails when the strict serial secure boot bit is not asserted in boot configuration word of BVT at the same time, the CM7 core can continue to run the application with all the HCU hardware user keys forbidden to accessed/loaded by user application.

Strict serial secure boot mode (BVT->BCW[BOOT_SEQ_STRICT] = 1)

If the secure boot fails and the strict serial secure boot bit is also asserted in boot configuration word of BVT at the same time, the CM7 core will go to static mode and run into an infinite loop in ROM.

4.3.5.2 Secure Boot Configuration

The secure boot configuration is achieved via secure boot config group structure and secure boot config section structures. For this device, a secure boot config group structure supports the maximal 8 secure boot sections configuration.

Yuntu Microelectronics

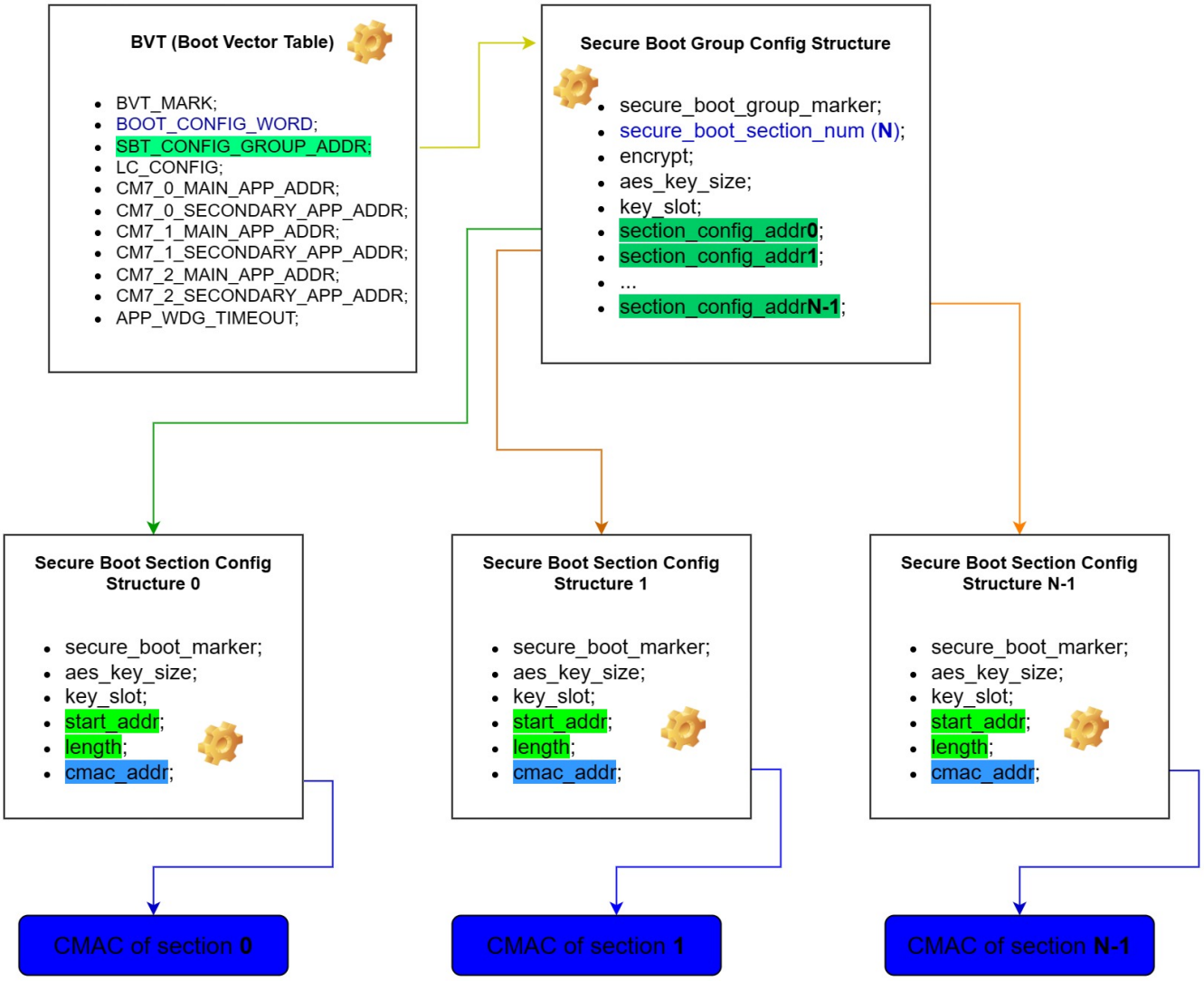


Figure 4.2: Secure Boot Group/Section Configuration Relationship with BVT

Yuntu Mi

It's recommended to place the secure boot group config structure, secure boot section config structures and CMAC of sections just after the BVT within a single Flash sector to save memory space. The user application memory map is as shown below:

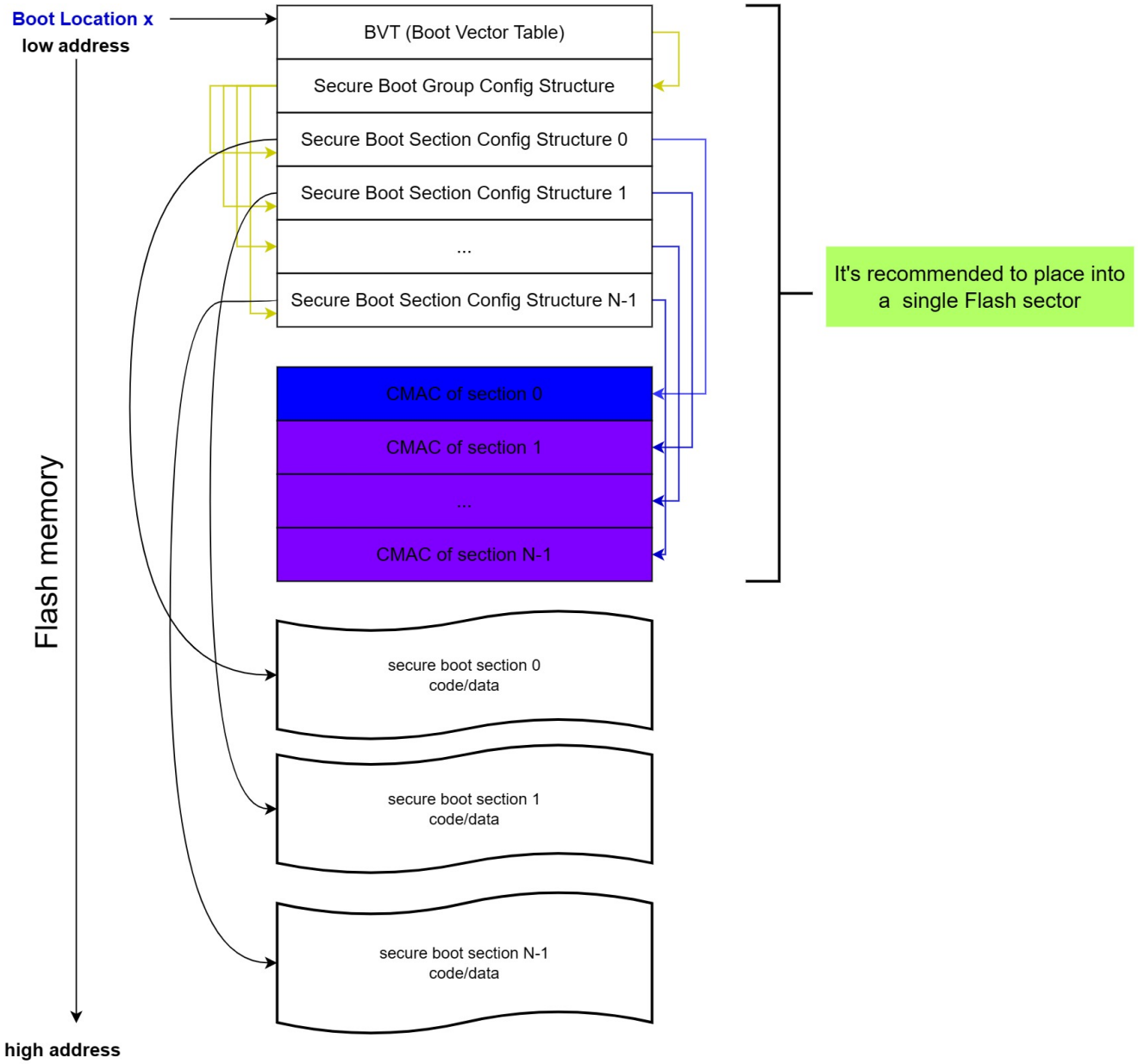


Figure 4.3: User Application Memory Map

4.3.5.2.1 Secure Boot Config Group Structure

The secure boot config group pointer stored in BVT points to a secure boot config group structure and contains the following configuration information:

1. Config Group Marker: fixed 0xACBEEFDD (4B)
2. Secure Boot Section Number(SBSN), <= max.8 (1B)
3. Encrypt Flag (1B)
4. AES Key Size/Type used for secure boot section config structure encrypt/decrypt (1B)
5. Key Slot used for secure boot section config structure encrypt/decrypt (1B)
6. Secure Boot Section Config Pointer (4B*SBSN)

Table 4.4: The Table of Secure Boot Config Group Structure

Address Offset	Size in Byte	Config Info.	Descriptions
0 x 00	4	Config Group Marker	A fixed marker of 0xACBEEFDD to identify the secure boot group config is valid.
0 x 04	1	Secure Boot Section Number (SBSN)	The secure boot sections are included in the config group. The max. Number of Secure boot sections is determined by the device family, for YTM32B1HAx, the max. Secure boot section number is 8.
0 x 05	1	Encrypt Flag	Indicate whether the secure boot section config info. is encrypt or decrypt, if it's true, ROM boot must decrypt the secure boot section before using it. Note: The encryption of the secure boot section config structures should be done by the application developer.
0 x 06	1	AES Key Type/Size	The AES-128/192/256 key used for secure boot section encrypt
0 x 07	1	Key Slot	The HCU hardware user key group/slot used for the secure boot section encrypt/decrypt.
0 x 08	4*SBSN	Secure Boot Section Config Pointer	The secure boot section config pointer (32bit address) is an array and has the SBSN of members. Each member points to a secure boot section configuration structure.

4.3.5.2.2 Secure Boot Section Configuration Structure

The secure boot section configuration structure should be 16 bytes aligned and has the following configuration information:

1. Config Section Marker: fixed 0x5AA5 (2B)
2. AES-128/192/256 type select (1B)
3. Key slot select (1B)
4. Start address (4B)
5. Length (4B)
6. CMAC address (4B)

Table 4.5: The Table of Secure Boot Section Configuration Structure

Address Offset	Size in Byte	Config Info.	Descriptions
0 x 00	2	Config Section Marker	A fixed marker of 0x5AA5 to identify the secure boot section config is valid.
0 x 02	1	AES Key Type/Size	The AES-128/192/256 key type/size used for this secure boot section CMAC calculation
0 x 03	1	Key Slot	The HCU hardware user key group/slot used for the secure boot section CMAC calculation
0 x 04	4	Section Start Address	The start address of secure boot section for CMAC generation/authorization
0 x 08	4	Length	The length of secure boot section for CMAC generation/authorization
0 x 0C	4	CMAC address	The address to store CMAC result of the secure boot section. The CMAC result is fixed to occupy 16 Bytes for AES-128/192/256 key.

4.3.6 Fast Wakeup from Powerdown Mode

ROM boot provides a fast wakeup mode when wakeup from the lowest power mode–Powerdown mode, which will trigger a system reset and most of SRAM except the 32KB standby SRAM, CPU registers and most peripherals register data loss. It will also parse the BVT, but will only configure the system clock (double the CM7 core clock frequency from default 48MHz 1/2 FIRC to 96MHz FIRC) per BVT->BCW configured, while the secure boot and CM7 core SCST will be ignored to reduce the ROM boot time.

The fast wakeup can be enabled by writing a fixed marker (0x5A5B6E6F) to register REGFILE->REG0.

After detecting the fast wakeup marker, ROM boot will then jump to the fast wakeup application stored into standby SRAM based on the fast wakeup IVT specified by register REGFILE->REG1.

4.4 Glossary

Yuntu Microelectronics

AES	Advanced Encryption Standard; a cipher primitive
BCW	Boot Configuration Word
BVT	Boot Vector Table
CMAC	Cipher-based Message Authentication Code
CPU	Central Processing Unit
FIRC	Fast Internal RC Oscillator
HCU	Hardware Cryptography Unit
IVT	Interrupt Vector Table
NVR	Non-Volatile Register
OTA	Over-The-Air
OTP	One-Time Programmable
ROM	Read-Only Memory
SBSN	Secure Boot Section Number
SCST	Structural Core Self-Test

Yuntu Microelectronics

Chapter 5

On-Chip RAM (OCRAM)

5.1 Chip-specific OCRAM information

5.1.1 Instantiation Information

YTM32B1HA0x contains 2 OCRAM modules, OCRAML and OCRAMU. Each OCRAM has a separate AHB-To-RAM bridge which can access up to 128KB RAM array. The two RAM arrays can be accessed at the same time by different masters, for example, OCRAML is accessed by Core and OCRAMU is accessed by DMA, this improves RAM access efficiency.

5.2 Introduction

5.2.1 Overview

The OCRAM Controller converts AHB read and write transactions to RAM array interface access, and supports RMW(Read Modified Write) when the write data size is less than 64bit. The controller also provides ECC safety function.

5.2.2 Features

- Up to 256KB
- Support 64/32/16/8bit access
- Support ECC safety function
  - 64bit data and 8bit ECC code
  - Single bit error correction
  - Double bit error detection
- Support RAM retention in Powerdown mode(32KB RAM which mapped to 0x2002_0000~0x2002_7FFF)

Yuntu Microelectronics

5.2.3 Block diagram

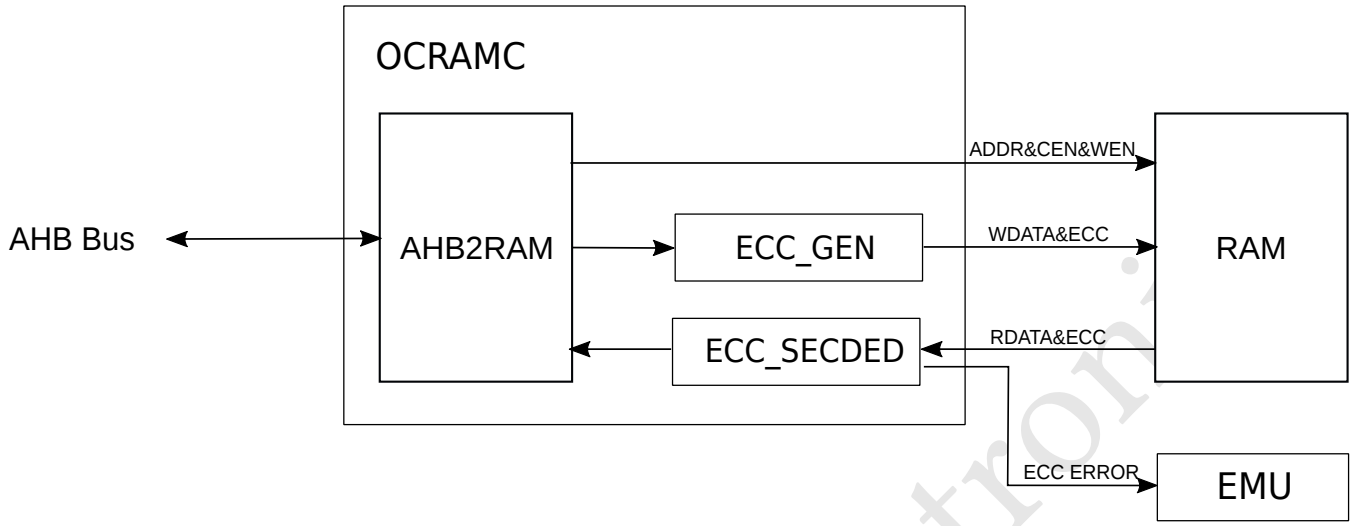


Figure 5.1: Block Diagram of OCRAM

5.3 Functional Description

5.3.1 RAM Initialization

After the chip is power-on, must initialize RAM to a known value using a 64-bit master before 32-bit masters can read or write to RAM. If RAM is not initialized, any attempt to execute a 32-bit read or write access terminates with an uncorrectable ECC error.

5.3.2 Reads

Read transfers, of any size, are completed with a zero wait state response on the system bus.

5.3.3 Writes

5.3.3.1 64 Bits Write

Aligned 64-bit writes execute in a single AHB data phase cycle, resulting in zero wait states on back-to-back writes of these sizes.

5.3.3.2 Less than 64-bit writes

Writes of size less than 64 bits incur a read-modify-write action as a consequence of the ECC coding scheme's 64-bit granularity. In the case of a read-modify-write action, the RAM controller performs SEC/DED on the read data. The write data is merged into the appropriate byte lanes along with the potentially corrected read data. A new codeword is generated based on the newly formed doubleword. This doubleword and its associated checkbits are subsequently written to RAM. On a read-modify-write

Yuntu Microelectronics

operation, the array read data is registered after it is decoded and before it is merged with the AHB write data. Therefore, writes of size less than 64 bits require the insertion of one wait state before the data phase can be completed.

5.3.3.3 Unaligned writes

Unaligned writes always generate read-modify-write operations in the ram controller in order to preserve the validity of the ECC codeword.

Yuntu Microelectronics

Chapter 6

Register File (REGFILE)

6.1 Introduction

Register file block is located at always on power domain which can retain value during Powerdown mode for some key information.

6.1.1 Features

- 32 bytes register file to retain value during Powerdown mode.
- Access through APB bus.

6.2 Register Definition

6.2.1 REGFILE Memory Map

REGFILE base address 40184000h

Table 6.1: REGFILE Memory Map

Offset	Register	Width	Access	Reset Value
0h	DATA Register (DR0)	32	RW	00000000h
4h	DATA Register (DR1)	32	RW	00000000h
8h	DATA Register (DR2)	32	RW	00000000h
ch	DATA Register (DR3)	32	RW	00000000h
10h	DATA Register (DR4)	32	RW	00000000h
14h	DATA Register (DR5)	32	RW	00000000h
18h	DATA Register (DR6)	32	RW	00000000h
1ch	DATA Register (DR7)	32	RW	00000000h

6.2.1.1 REGFILE DR Register

Offset: 0h + n * 4h

Yuntu Microelectronics

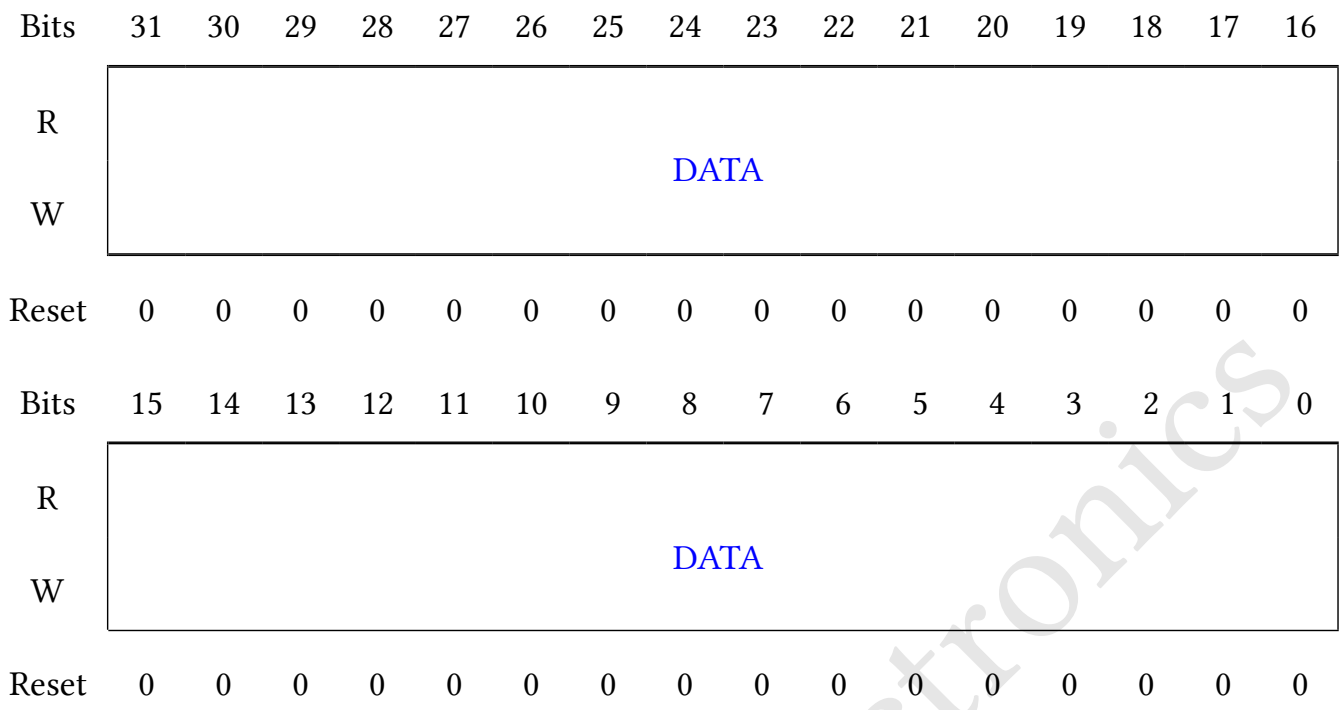


Table 6.2: REGFILE DR Register Description

Field	Function
31 - 0 DATA	**DATA** The data value stored at register file block. The register is only reseted by POR.

Yuntu Microelectronics

Part III

Signal Multiplexing and Pin Assignment

Yuntu Microelectronics

Chapter 7

Pinmux

7.1 Pinmux Table

Various function signals are assigned to chip pins under the select signals from Port Control Modules(PCTRL).

The following table is YTM32B1HA0 pinmux table.

Yuntu Microelectronics

Table 7.1: Pinmux Table

100 LQFP	144 LQFP	176 LQFP	NAME	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9	ALT10	ALT11	ALT12	ALT13	ALT14	ALT15
		1	PTA_18		PTA_18	eTMR4_CH0	LINFlexD1_TX	SPI1_SOUT	eTMR5_CH0				SPI6_SOUT					MPWM2_CH0	
		2	PTA_19		PTA_19	eTMR4_CH1	LINFlexD1_RX	SPI1_SCK					SPI6_SCK					MPWM2_CH1	
		3	PTA_20		PTA_20	eTMR4_CH2		SPI1_SIN					SPI6_SIN					MPWM2_CH2	
1	1	4	PTE_16	ADC0_S28	PTE_16	LINFlexD3_TX	SPI2_SIN	eTMR2_CH7	eTMR4_FLT0		TMU_OUT7	FMU_ERR_OUT1	SPI6_PCS0					MPWM2_CH3	FMU_ERR_IN1
2	2	5	PTE_15	ADC0_S27	PTE_15	LINFlexD3_RX	SPI2_SCK	eTMR2_CH6	eTMR4_FLT1		TMU_OUT6	FMU_ERR_OUT0	SPI6_PCS1	ACMP1_ACTIVE				MPWM2_CH4	FMU_ERR_IN0
		6	PTA_21		PTA_21	eTMR4_CH3		SPI1_PCS0				SPI2_PCS2	SPI6_PCS2					MPWM2_CH5	
3	3	7	VDD11	VDD11															
4	4	8	VDD25	VDD25															
		9	PTA_22		PTA_22	eTMR4_CH4		SPI1_PCS1					SPI6_PCS3					MPWM2_CH7	
5	5	10	PTE_11	ADC0_S30	PTE_11	SPI2_PCS0	lpTMR0_ALT1	eTMR2_CH5	LINFlexD3_TX	ETM_TRACE_D0	TMU_OUT5	LINFlexD4_TX					SENT1_RX_IN0	MPWM2_CH8	
6	6	11	PTE_10	ADC0_S29	PTE_10	SCU_CLKOUT	SPI2_PCS1	eTMR2_CH4	LINFlexD3_RX		TMU_OUT4	LINFlexD4_RX		SPI3_SIN			SENT1_RX_IN1	MPWM2_CH9	
7	7	12	PTE_13		PTE_13	eTMR4_CH5	SPI2_PCS2	eTMR2_FLT0	SPI2_PCS0							eTMR3_CH5	SENT1_RX_IN2	MPWM2_CH10	
		13	PTA_23		PTA_23	eTMR4_CH6												MPWM2_CH11	
8	8	14	PTE_5		PTE_5	TCLK_IN2	eTMR2_QD_PHA	eTMR2_CH3	CAN0_TX	eTMR3_CH5							SENT1_RX_IN3	MPWM2_CH12	
9	9	15	PTE_4		PTE_4	ETM_TRACE_D1	eTMR2_QD_PHB	eTMR2_CH2	CAN0_RX	eTMR3_CH4		SPI0_PCS0	eTMR3_CH7		SPI1_PCS1			MPWM2_CH13	
		16	PTA_24		PTA_24	eTMR4_CH7								eTMR4_CH0				MPWM2_CH14	
	10	17	PTA_25		PTA_25	eTMR5_CH0												MPWM2_CH15	
10	11	18	VDD	VDD															
	12	19	VSS	VSS															
11	13	20	VDDA	VDDA															
12	14	21	VREFH	VREFH															
13	15	22	VREFL	VREFL															
14	16	23	VSSA	VSSA															
15	17	24	PTB_7	EXTAL	PTB_7	I2C0_SCL		SPI3_SCK	eTMR4_FLT3		TMU_OUT2								
16	18	25	PTB_6	XTAL	PTB_6	I2C0_SDA		SPI3_SIN	eTMR4_FLT2		TMU_OUT1								
	19	26	PTA_26		PTA_26	eTMR5_CH1	SPI1_PCS0	SPI0_PCS0			-			eTMR4_CH1					
17	20	27	PTE_14		PTE_14	eTMR0_FLT1		eTMR2_FLT1				CAN4_RX			eTMR2_CH3	LINFlexD5_TX			
18	21	28	PTE_3		PTE_3	eTMR0_FLT0	SPI1_SIN	eTMR2_FLT0	SPI3_SOUT	TMU_IN6	ACMP0_OUT	CAN4_TX	eTMR2_CH3		-	LINFlexD5_RX			
	22	29	PTA_27		PTA_27	eTMR5_CH2	SPI1_SOUT	LINFlexD0_TX	CAN0_TX				-		eTMR4_CH2				
19	23	30	PTE_12	DAC1_OUT	PTE_12	eTMR0_FLT3	LINFlexD2_TX	eTMR5_FLT0	SPI3_PCS0				eTMR3_CH5	-		CAN5_TX			
	24	31	PTA_28		PTA_28	eTMR5_CH3	SPI1_SCK	LINFlexD0_RX	CAN0_RX					eTMR4_CH3					
20	25	32	PTD_17		PTD_17	eTMR0_FLT2	LINFlexD2_RX	eTMR5_FLT1	SPI3_PCS0		-	SPI5_PCS0			CAN5_RX	eTMR2_CH2	-		
	26	33	PTA_29		PTA_29	eTMR5_CH4		LINFlexD2_TX	SPI1_SIN				-	eTMR4_CH4					
	27	34	PTA_30		PTA_30	eTMR5_CH5	LINFlexD2_RX	SPI0_SOUT						eTMR4_CH5	SPI1_SOUT				
21	28	35	PTD_16	EXTAL32	PTD_16	eTMR0_CH1	CAN4_TX	SPI0_SIN	ACMP0_ACTIVE	ETM_TRACE_D2	ETM_TRACE_CLKOUT		eTMR4_CH7		LINFlexD8_TX		-		
22	29	36	PTD_15	XTAL32	PTD_15	eTMR0_CH0	CAN4_RX	SPI0_SCK	SPI3_PCS1	ETM_TRACE_D3	LINFlexD8_RX	SAI1_SYNC	-		CAN3_RX	eTMR4_CH6			
23	30	37	PTE_9		PTE_9	eTMR0_CH7	SPI1_SCK	I2C2_SDA	-	-		SPI5_SOUT			CAN3_TX	eTMR4_CH5			
	31	38	VSS	VSS															
	32	39	VDD	VDD															
	33	40	PTA_31		PTA_31	eTMR5_CH6		SPI0_PCS1			TMU_OUT8			eTMR4_CH6				MPWM0_CH4	
24	34	41	PTD_14		PTD_14	eTMR2_CH5	LINFlexD1_TX	I2C0_SCL	-	SAI1_MCLK		SPI5_SCK	SPI5_PCS3	SCU_CLKOUT	ACMP0_ACTIVE			MPWM2_CH6	
25	35	42	PTD_13		PTD_13	eTMR2_CH4	LINFlexD1_RX	I2C0_SDA	-	SAI1_DATA0	RTC_CLKOUT	SPI5_SIN	SPI5_PCS2					MPWM0_CH5	
	36	43	PTB_18	ADC0_S16	PTB_18	eTMR5_CH7		SPI1_PCS1			TMU_OUT9	LINFlexD2_TX		eTMR4_CH7				MPWM0_CH6	
		44	PTB_19		PTB_19	eTMR4_CH7				eTMR5_CH7	TMU_OUT10	LINFlexD2_RX					QSPI_DQSA		
	37	45	PTB_20	ADC0_S17	PTB_20	LINFlexD3_TX			I2C1_SDA						eTMR5_CH0		QSPI_IOFA[4]		
	38	46	PTB_21	ADC0_S18	PTB_21	LINFlexD3_RX			I2C1_SCL						eTMR5_CH1		QSPI_IOFA[5]		
26	39	47	PTE_8	ACMP0_IN3	PTE_8	eTMR0_CH6	SPI5_PCS1	I2C2_SCL	-	SAI1_BCLK		SPI3_PCS1				SENT0_RX_IN0	QSPI_IOFA[6]		
27	40	48	PTB_5		PTB_5	eTMR0_CH5	SPI0_PCS1	SPI0_PCS0		TMU_IN0	-	-	eTMR4_CH3	I2C2_SCL		SENT0_RX_IN1	QSPI_IOFA[7]		
28	41	49	PTB_4		PTB_4	eTMR0_CH4	SPI0_SOUT			TMU_IN1	-	-	eTMR4_CH2	SCU_CLKOUT		SENT0_RX_IN2			QSPI_IOFB[0]
29	42	50	PTC_3	ADC0_S11/ ACMP0_IN4	PTC_3	eTMR0_CH3	CAN0_TX	LINFlexD0_TX	SPI4_PCS0	QSPI_PCSFA						SENT0_RX_IN3			QSPI_IOFB[3]

Yuntu Microelectronics Reference

Table 7.1 continued from previous page

100 LQFP	144 LQFP	176 LQFP	NAME	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9	ALT10	ALT11	ALT12	ALT13	ALT14	ALT15
30	43	51	PTC_2	ADC0_S10/ ACMP0_IN5	PTC_2	eTMR0_CH2	CAN0_RX	LINFlexD0_RX	SPI4_SCK	ETM_TRACE_ CLKOUT	QSPI_IOFA[3]	-	SPI0_PCS2	SPI3_PCS2	-				
31	44	52	PTD_7	ACMP0_IN6	PTD_7	LINFlexD2_TX	eTMR0_CH3	eTMR2_FLT3	SPI4_SIN	ETM_TRACE_D0	QSPI_IOFA[1]	-			SPI3_PCS3	SPI0_PCS3	-		
32	45	53	PTD_6	ACMP_IN7[1]	PTD_6	LINFlexD2_RX	eTMR0_CH2	eTMR2_FLT2	SPI4_SOUT		SPI0_PCS0	-	-		eTMR4_CH4	-			QSPI_IOFB[1]
33	46	54	PTD_5		PTD_5	eTMR2_CH3	lpTMR0_ALT2	eTMR2_FLT1	SPI4_PCS1	TMU_IN7	SPI0_PCS1	-	-		eTMR0_CH2	-			QSPI_IOFB[2]
34	47	55	PTD_12		PTD_12	eTMR2_CH2		ETM_TRACE_D1	-	SPI0_SOUT	QSPI_IOFA[2]		-			SPI5_SIN			
35	48	56	PTD_11		PTD_11	eTMR2_CH1	eTMR2_QD_PHA	ETM_TRACE_D2		SPI0_SCK	QSPI_IOFA[0]	-	-		-		SPI5_SOUT		
36	49	57	PTD_10		PTD_10	eTMR2_CH0	eTMR2_QD_PHB	ETM_TRACE_D3	SPI0_SIN		QSPI_SCKFA	-	-		eTMR4_CH2		SPI5_SCK	SCU_CLKOUT	
37	50	58	VSS	VSS															
38	51	59	VDD	VDD															
39	52	60	PTC_1	ADC0_S9	PTC_1	eTMR0_CH1	SPI2_SOUT	CAN3_TX		eTMR1_CH7		CAN3_RX	-	-			-		QSPI_SCKFB
40	53	61	PTC_0	ADC0_S8	PTC_0	eTMR0_CH0	SPI2_SIN	CAN3_RX		eTMR1_CH6	-	CAN3_TX	-				-		QSPI_DQSB
41	54	62	PTD_9		PTD_9	LINFlexD4_TX		eTMR2_FLT3		eTMR1_CH5			-	I2C1_SCL	LINFlexD6_TX	FMU_ERR_IN2	-		QSPI_IOFB[4]
42	55	63	PTD_8		PTD_8	LINFlexD4_RX		eTMR2_FLT2		eTMR1_CH4		SPI3_SOUT	-	I2C1_SDA	LINFlexD6_RX	FMU_ERR_IN3	-		QSPI_IOFB[5]
43	56	64	PTC_17	ADC0_S15	PTC_17	eTMR1_FLT3	CAN2_TX	SPI4_PCS0	eTMR2_CH1			SPI3_SCK					-		QSPI_IOFB[6]
44	57	65	PTC_16	ADC0_S14	PTC_16	eTMR1_FLT2	CAN2_RX	SPI4_SCK	eTMR2_CH0		I2C1_SDA	SPI3_SIN	eTMR4_CH1			LINFlexD2_RX	-		QSPI_IOFB[7]
	58	66	PTB_22	ADC0_S19	PTB_22	eTMR5_CH2	SPI3_PCS1		LINFlexD1_TX			CAN2_RX	-	CAN1_TX					QSPI_IOFB[3]
45	59	67	PTC_15	ADC0_S13	PTC_15	eTMR1_CH3	SPI2_SCK	SPI4_SIN	LINFlexD4_TX	TMU_IN8	I2C1_SCL	CAN2_TX	-	CAN1_RX	LINFlexD2_TX		-	-	QSPI_PCSFB
	60	68	PTB_23	ADC0_S20	PTB_23		LINFlexD1_RX				CAN1_RX		-	eTMR5_CH3					
		69	PTB_24		PTB_24	eTMR5_CH4					CAN1_TX								
46	61	70	PTC_14	ADC0_S12	PTC_14	eTMR1_CH2	SPI2_PCS0	SPI4_SOUT	LINFlexD4_RX	TMU_IN9	eTMR3_CH4		-	CAN2_RX			-	-	
	62	71	PTB_25	ADC0_S21	PTB_25			SPI4_PCS1	SPI2_PCS0				eTMR5_CH5	CAN2_TX					
		72	PTB_26		PTB_26	eTMR5_CH6													
47	63	73	PTB_3	ADC0_S7	PTB_3	eTMR1_CH1	SPI0_SIN	eTMR1_QD_PHA	CAN4_TX	TMU_IN2					SPI2_SOUT	SAI0_MCLK			
	64	74	PTB_27	ADC0_S22	PTB_27	eTMR5_FLT2			SPI2_SOUT			LINFlexD5_TX	eTMR5_CH7					MPWM0_CH0	
	65	75	PTB_28	ADC0_S23	PTB_28	eTMR5_FLT3			SPI2_SIN		-		LINFlexD5_RX					MPWM0_CH1	
	66	76	VSS	VSS															
	67	77	VDD	VDD															
48	68	78	PTB_2	ADC0_S6	PTB_2	eTMR1_CH0	SPI0_SCK	eTMR1_QD_PHB		TMU_IN3	SAI0_DATA0		CAN4_RX	LINFlexD9_RX		SPI2_SIN		MPWM0_CH2	
	69	79	PTB_29		PTB_29				SPI2_SCK		SAI0_DATA1	LINFlexD6_TX		LINFlexD9_TX				MPWM0_CH3	
		80	PTB_30		PTB_30													MPWM0_CH4	
49	70	81	PTC_13		PTC_13	eTMR3_CH7	eTMR2_CH7				SAI0_SYNC				eTMR3_CH3				
		82	PTB_31		PTB_31								CAN7_TX					MPWM0_CH5	
		83	PTC_18		PTC_18						SAI0_DATA2	LINFlexD6_RX	CAN7_RX					MPWM0_CH6	
50	71	84	PTC_12		PTC_12	eTMR3_CH6	eTMR2_CH6	SPI2_PCS1			SAI0_BCLK		CAN6_TX		eTMR3_CH2				
	72	85	PTC_19		PTC_19				SPI2_PCS1		SAI0_DATA3	LINFlexD7_TX	CAN6_RX					MPWM0_CH7	
		86	PTC_20		PTC_20							LINFlexD7_RX						MPWM0_CH8	
		87	PTC_21		PTC_21													MPWM0_CH9	
		88	PTC_22		PTC_22							eTMR1_CH0						MPWM0_CH10	
	73	89	PTC_23		PTC_23	SPI0_SCK												MPWM0_CH11	
		90	PTC_24		PTC_24	eTMR4_CH0	eTMR3_CH0						CAN6_TX					MPWM0_CH12	
		91	PTC_25		PTC_25	eTMR4_CH1	eTMR3_CH1		SPI4_PCS1				CAN6_RX					MPWM0_CH13	
51	74	92	PTC_11		PTC_11	eTMR3_CH5	eTMR4_CH2	CAN5_TX		TMU_IN10			CAN5_RX		eTMR3_CH1	SPI4_SOUT			
		93	PTC_26		PTC_26	eTMR4_CH3	eTMR3_CH3				SPI4_SIN		CAN5_RX					MPWM0_CH14	
52	75	94	PTC_10	ADC1_S27	PTC_10	eTMR3_CH4		CAN5_RX	SPI4_PCS0	TMU_IN11	eTMR3_CH0	eTMR0_CH6		CAN5_TX		SPI2_PCS1			
	76	95	PTC_27		PTC_27	eTMR4_CH4					SPI4_SCK	CAN5_TX			eTMR3_CH4			MPWM0_CH15	
53	77	96	PTB_1	ADC0_S5	PTB_1	LINFlexD0_TX	SPI0_SOUT	TCLK_IN0	CAN0_TX	eTMR4_CH5				eTMR3_CH5		eTMR0_CH7		MPWM0_CH0	
54	78	97	PTB_0	ADC0_S4	PTB_0	LINFlexD0_RX	SPI0_PCS0	lpTMR0_ALT3	CAN0_RX	eTMR4_CH6				eTMR3_CH6		eTMR0_CH3		MPWM0_CH1	
	79	98	PTC_28		PTC_28	eTMR4_CH7			I2C1_SCL			CAN3_TX			eTMR3_CH7				
55	80	99	PTC_9		PTC_9	LINFlexD1_TX	eTMR1_FLT1	eTMR5_CH0	CAN4_TX	SPI0_SIN	I2C0_SDA		CAN1_RX	eTMR4_CH0			SPI5_PCS1	MPWM0_CH2	

Yuntu Microelectronics

Table 7.1 continued from previous page

100 LQFP	144 LQFP	176 LQFP	NAME	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9	ALT10	ALT11	ALT12	ALT13	ALT14	ALT15
56	81	100	PTC_8	DAC0_OUT	PTC_8	LINFlexD1_RX	eTMR1_FLT0	eTMR5_CH1	CAN4_RX	SPI0_SCK	I2C0_SCL		CAN1_TX	eTMR4_CH1			SPI5_PCS0	MPWM0_CH3	
	82	101	PTC_29		PTC_29	eTMR5_CH2			I2C1_SDA			CAN3_RX	eTMR4_CH2						
57	83	102	PTA_7	ADC0_S3	PTA_7	eTMR0_FLT2	eTMR5_CH3	RTC_CLKIN	I2C2_SCL			LINFlexD3_TX	SPI0_PCS1	eTMR4_CH3	CAN0_TX			SPI5_SCK	
	84	103	PTC_30		PTC_30	eTMR5_CH4						CAN4_TX	eTMR4_CH4						
58	85	104	PTA_6	ADC0_S2	PTA_6	eTMR0_FLT1	SPI1_PCS1	eTMR5_CH5	I2C2_SDA	SPI3_PCS1		LINFlexD3_RX	CAN0_RX	eTMR4_CH5			SPI5_SIN		
	86	105	PTC_31		PTC_31	eTMR5_CH6		I2C1_SDA					CAN4_RX		eTMR4_CH6				
59	87	106	PTE_7		PTE_7	eTMR0_CH7	eTMR3_FLT0			SPI3_SCK			LINFlexD4_RX		SPI5_SOUT				
	88	107	PTD_18	ADC1_S16	PTD_18	eTMR5_CH7	CAN5_TX	I2C1_SCL	I2C1_SDA				CAN7_TX						
	89	108	PTD_19	ADC1_S17	PTD_19		CAN5_RX		I2C1_SCL				CAN7_RX						
60	90	109	VSS	VSS															
61	91	110	VDD	VDD															
62	92	111	PTA_17		PTA_17	eTMR0_CH6	eTMR3_FLT0		eTMR5_FLT0	SPI3_SOUT		LINFlexD4_TX		EWDG_OUT_b				MPWM2_CH0	
63	93	112	PTB_17	ADC1_S26	PTB_17	eTMR0_CH5	SPI1_PCS3	eTMR5_FLT1		SPI3_PCS0		LINFlexD4_RX		eTMR3_CH7				MPWM2_CH1	
		113	PTD_20	ADC1_S31	PTD_20	eTMR5_CH1			SPI1_PCS2	SPI3_SIN									
64	94	114	PTB_16	ADC1_S15	PTB_16	eTMR0_CH4	SPI1_SOUT	LINFlexD4_TX					SENT1_RX_IN0						
65	95	115	PTB_15	ADC1_S14	PTB_15	eTMR0_CH3	SPI1_SIN			LINFlexD7_TX			SENT1_RX_IN1					MPWM2_CH2	
66	96	116	PTB_14	ADC1_S9	PTB_14	eTMR0_CH2	SPI1_SCK			LINFlexD7_RX			SENT1_RX_IN2					MPWM2_CH3	
		117	PTD_21		PTD_21	eTMR5_CH2													
67	97	118	PTB_13	ADC1_S8	PTB_13	eTMR0_CH1	eTMR3_FLT1	CAN2_TX			LINFlexD8_TX	SPI3_PCS2	SENT1_RX_IN3						
68	98	119	PTB_12	ADC1_S7	PTB_12	eTMR0_CH0	eTMR3_FLT2	CAN2_RX			LINFlexD8_RX	SPI3_PCS3							
	99	120	PTD_22	ADC1_S18	PTD_22	eTMR4_FLT2							eTMR5_CH3						
69	100	121	PTD_4	ADC1_S6	PTD_4	eTMR0_FLT3	eTMR3_FLT3				SPI5_PCS0		eTMR5_CH7	SPI1_PCS1		SPI5_PCS2			
70	101	122	PTD_3	ADC1_S3	PTD_3	eTMR3_CH5	SPI1_PCS0	I2C1_SCL	CAN5_TX	TMU_IN4	NMI_b		eTMR5_CH6	LINFlexD3_RX					
71	102	123	PTD_2	ADC1_S2	PTD_2	eTMR3_CH4	SPI1_SOUT	I2C1_SDA	CAN5_RX	TMU_IN5	SPI5_SOUT		eTMR5_CH5	LINFlexD3_TX					
	103	124	PTD_23	ADC1_S19	PTD_23	eTMR4_FLT3							eTMR5_CH4						
72	104	125	PTA_3	ADC1_S1	PTA_3	eTMR3_CH1	I2C0_SCL			LINFlexD0_TX	SPI5_SCK	FMU_ERR_OUT1	eTMR5_CH4	SPI1_SCK					FMU_ERR_IN1
73	105	126	PTA_2	ADC1_S0/ ACMP1_IN2	PTA_2	eTMR3_CH0	I2C0_SDA			LINFlexD0_RX	SPI5_SIN	FMU_ERR_OUT0	eTMR5_CH3				SPI1_SIN		FMU_ERR_IN0
	106	127	PTD_24	ADC1_S20	PTD_24								eTMR5_CH5						
74	107	128	PTB_11	ADC1_S30	PTB_11	eTMR3_CH3	LINFlexD5_TX					SPI4_SIN	eTMR5_CH2						
		129	VDD	VDD															
		130	VSS	VSS															
75	108	131	PTB_10	ADC1_S29	PTB_10	eTMR3_CH2	LINFlexD5_RX	SAI1_MCLK	LINFlexD9_TX			SPI4_SCK	eTMR5_CH1						
		132	PTD_25		PTD_25	eTMR5_CH6													
76	109	133	PTB_9	ADC0_S25	PTB_9	eTMR5_CH0	eTMR3_CH1	SAI1_DATA0	LINFlexD9_RX			SPI4_SOUT							
		134	PTD_26		PTD_26	eTMR5_CH7				SPI5_SCK									
	110	135	PTD_27	ADC1_S21	PTD_27	eTMR5_FLT2				SPI5_SOUT									
77	111	136	PTB_8	ADC0_S24	PTB_8	eTMR3_CH0		SAI1_BCLK		SPI0_PCS5		SPI4_PCS0	eTMR4_CH7						
	112	137	PTD_28	ADC1_S22	PTD_28	eTMR5_FLT3				SPI5_SIN									
78	113	138	PTA_1	ADC0_S1/ ACMP_IN1[2]	PTA_1	eTMR1_CH1	LINFlexD5_TX		eTMR1_QD_PHA	SPI0_PCS6	TMU_OUT0	SPI4_PCS1							
	114	139	PTD_29	ADC1_S23	PTD_29					SPI5_PCS2									
79	115	140	PTA_0	ADC0_S0/ ACMP_IN0[3]	PTA_0	eTMR2_CH1	LINFlexD5_RX		eTMR2_QD_PHA	SPI0_PCS7	TMU_OUT3	SPI4_PCS2		eTMR3_CH0					FMU_ERR_IN2
	116	141	PTD_30		PTD_30					SPI5_PCS3									
		142	PTD_31		PTD_31														
80	117	143	PTC_7	ADC1_S5	PTC_7	LINFlexD1_TX	CAN1_TX	eTMR3_CH3	eTMR3_CH7	eTMR1_QD_PHA		I2C1_SCL		SPI0_PCS0		CAN2_TX			FMU_ERR_IN3
81	118	144	PTC_6	ADC1_S4	PTC_6	LINFlexD1_RX	CAN1_RX	eTMR3_CH2	eTMR3_CH6	eTMR1_QD_PHB		I2C1_SDA		SPI0_PCS1		CAN2_RX	SPI1_PCS1	MPWM1_CH9	
		145	PTE_17		PTE_17														
82	119	146	PTA_16	ADC1_S13	PTA_16	eTMR1_CH3	SPI1_PCS2	SPI0_PCS4	LINFlexD6_TX			SPI4_PCS3						MPWM1_CH0	
		147	PTE_18		PTE_18														
83	120	148	PTA_15	ADC1_S12	PTA_15	eTMR1_CH2	SPI0_PCS3	SPI2_PCS3	LINFlexD6_RX	SPI5_PCS0	SAI0_SYNC							MPWM1_CH1	
84	121	149	PTE_6	ADC1_S11	PTE_6	SPI0_PCS2		eTMR3_CH7		SAI0_DATA1		eTMR4_CH6		ETM_TRACE_D2	ETM_TRACE_ CLKOUT			MPWM1_CH2	
85	122	150	PTE_2	ADC1_S10	PTE_2	SPI0_SOUT	lpTMR0_ALT3	eTMR3_CH6		SAI0_DATA2		eTMR0_CH3	eTMR4_CH0		ETM_TRACE_D3			MPWM1_CH3	
86	123	151	VSS	VSS															
87	124	152	VDD	VDD															

Yuntu Microelectronics

Table 7.1 continued from previous page

100 LQFP	144 LQFP	176 LQFP	NAME	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	ALT8	ALT9	ALT10	ALT11	ALT12	ALT13	ALT14	ALT15
	125	153	PTE_19		PTE_19	eTMR2_CH6			I2C4_SDA										
	126	154	PTE_20		PTE_20	eTMR4_CH0			I2C4_SCL				eTMR3_CH0						
88	127	155	PTA_14	ADC1_S28	PTA_14	eTMR0_FLT0	eTMR3_FLT1			eTMR1_FLT0	SAI0_DATA3		eTMR3_CH4	SPI1_PCS3	SPI5_PCS1			MPWM1_CH0	
	128	156	PTE_21		PTE_21	eTMR4_CH1				SPI4_SIN			eTMR3_CH1					MPWM1_CH1	
	129	157	PTE_22		PTE_22	eTMR4_CH2				SPI4_SCK			eTMR3_CH2					MPWM1_CH2	
89	130	158	PTA_13	ADC1_S25	PTA_13	eTMR1_CH7	CAN1_TX	SPI3_PCS0		eTMR2_QD_PHA	SAI0_DATA0	SPI1_PCS4	eTMR3_CH3					MPWM1_CH3	
	131	159	PTE_23		PTE_23	eTMR4_CH3	CAN1_RX			SPI4_PCS0			eTMR3_CH3					MPWM1_CH4	
	132	160	PTE_24		PTE_24	eTMR4_CH4	CAN2_TX			SPI4_PCS1			eTMR3_CH4					MPWM1_CH5	
	133	161	PTE_25		PTE_25	eTMR4_CH5	CAN2_RX			SPI4_SOUT			eTMR3_CH5					MPWM1_CH6	
90	134	162	PTA_12	ADC1_S24	PTA_12	eTMR1_CH6	CAN1_RX	SPI3_SCK		eTMR2_QD_PHB	SAI0_BCLK	ACMP1_OUT	SPI1_PCS5	eTMR3_CH2			SENT0_RX_IN0	MPWM1_CH7	
91	135	163	PTA_11		PTA_11	eTMR1_CH5	CAN1_TX	SPI3_SIN	ACMP0_ACTIVE	SPI1_PCS0			eTMR3_CH1	SPI7_PCS0			SENT0_RX_IN1	MPWM1_CH8	
92	136	164	PTA_10		PTA_10	eTMR1_CH4		SPI3_SOUT									SENT0_RX_IN2		JTAG_TDO SWD_SWO
93	137	165	PTE_1		PTE_1	SPI0_SIN		I2C1_SCL	SPI1_PCS0	eTMR1_FLT1			SPI0_SCK	SPI7_SIN	LINFlexD7_TX		SENT0_RX_IN3	MPWM1_CH10	
94	138	166	PTE_0		PTE_0	SPI0_SCK	TCLK_IN1	I2C1_SDA	SPI1_SOUT	eTMR1_FLT2			SPI0_SIN	SPI7_SOUT	LINFlexD7_RX			MPWM1_CH11	
		167	PTE_26		PTE_26	eTMR3_CH6	eTMR4_CH6											MPWM1_CH12	
95	139	168	PTC_5		PTC_5	eTMR2_CH0	RTC_CLKOUT	SPI3_PCS1	I2C3_SDA	eTMR2_QD_PHB	SAI0_MCLK			SPI7_SCK				MPWM1_CH13	JTAG_TDI
96	140	169	PTC_4	ACMP0_IN2/ ACMP1_IN3	PTC_4	eTMR1_CH0	RTC_CLKOUT		I2C3_SCL	eTMR1_QD_PHB									JTAG_TCK SWD_CLK
97	141	170	PTA_5		PTA_5	CAN3_TX	TCLK_IN1												RESET_b
		171	VSS	VSS															
		172	VDD	VDD															
98	142	173	PTA_4		PTA_4	CAN3_RX		ACMP0_OUT						SPI7_PCS1					JTAG_TMS SWD_IO
		174	PTE_27		PTE_27	eTMR3_CH7	eTMR4_CH7							SPI7_PCS3					
99	143	175	PTA_9	ADC0_S31	PTA_9	LINFlexD2_TX	SPI2_PCS0	SPI1_SIN	eTMR3_FLT2	eTMR1_FLT3	eTMR4_FLT0		SPI3_PCS0	SPI7_PCS2		FMU_ERR_IN2		MPWM1_CH14	
100	144	176	PTA_8	ADC0_S26	PTA_8	LINFlexD2_RX	SPI2_SOUT	SPI1_SCK	eTMR3_FLT3	eTMR4_FLT1			eTMR4_CH4			FMU_ERR_IN3		MPWM1_CH15	

1. ACMP0 and ACMP1 share ACMP_IN7.
2. ACMP0 and ACMP1 share ACMP_IN1.
3. ACMP0 and ACMP1 share ACMP_IN0.

Yuntu Microelectronics

7.2 Glossary

EXTAL	External Crystal Input
XTAL	External Crystal Output

Yuntu Microelectronics

Chapter 8

General Purpose Input/Output (GPIO)

8.1 Introduction

8.1.1 Features

Features of the GPIO module include:

- Port Data Output register with corresponding set/clear/toggle registers
- Port Data Direction register
- Inversion for data inputs
- Interrupt flag and enable registers for each pin
- Support for edge sensitive (rising, falling, both) or level sensitive (low, high)
- Asynchronous wake-up in low-power modes
- Pin interrupt is functional in all digital pin muxing modes
- Support getting state of the port in all digital pin muxing modes

8.1.2 Modes of Operation

8.1.2.1 Active Mode

In Active mode, the GPIO module operates normally.

8.1.2.2 Sleep Mode

In Sleep mode, the GPIO module continues to operates normally and can be configured to wake up the chip from Sleep mode if an enable interrupt is detected.

8.1.2.3 Deepsleep Mode

In Deepsleep mode, the GPIO configuration is static and the GPIO can be configured to wake up the chip from Deepsleep mode via an asynchronous wakeup signal if an enabled interrupt is detected.

8.1.2.4 Standby Mode

In Standby mode, the GPIO configuration is static and the GPIO can be configured to wake up the chip from Standby mode via an asynchronous wakeup signal if an enabled interrupt is detected.

Yuntu Microelectronics

8.1.2.5 Powerdown Mode

GPIO wakeup interrupt is not available under Powerdown mode, application should use WKU module and WKU pins to wake up the chip.

GPIO module will keep previous configuration until system wakes up from Powerdown mode.

8.2 Register Definition

8.2.1 GPIO Memory Map

GPIOA base address 40044000h
GPIOB base address 40044100h
GPIOC base address 40044200h
GPIOD base address 40044300h
GPIOE base address 40044400h

Table 8.1: GPIO Memory Map

Offset	Register	Width	Access	Reset Value
0h	Port Data Output Register (PDOR)	32	RW	00000000h
4h	Port Set Output Register (PSOR)	32	WO	00000000h
8h	Port Clear Output Register (PCOR)	32	WO	00000000h
ch	Port Toggle Output Register (PTOR)	32	WO	00000000h
10h	Port Data Input Register (PDIR)	32	RO	00000000h
14h	Port Output Enable Register (POER)	32	RW	00000000h
18h	Port Input Enable Register (PIER)	32	RW	FFFFFFFFh
1ch	Port Interrupt Status Flag Register (PIFR)	32	RW	00000000h
80h	Port Control Register (PCR0)	32	RW	00000000h
84h	Port Control Register (PCR1)	32	RW	00000000h
88h	Port Control Register (PCR2)	32	RW	00000000h
8ch	Port Control Register (PCR3)	32	RW	00000000h
90h	Port Control Register (PCR4)	32	RW	00000000h
94h	Port Control Register (PCR5)	32	RW	00000000h
98h	Port Control Register (PCR6)	32	RW	00000000h
9ch	Port Control Register (PCR7)	32	RW	00000000h
a0h	Port Control Register (PCR8)	32	RW	00000000h
a4h	Port Control Register (PCR9)	32	RW	00000000h
a8h	Port Control Register (PCR10)	32	RW	00000000h
ach	Port Control Register (PCR11)	32	RW	00000000h
b0h	Port Control Register (PCR12)	32	RW	00000000h
b4h	Port Control Register (PCR13)	32	RW	00000000h
b8h	Port Control Register (PCR14)	32	RW	00000000h
bch	Port Control Register (PCR15)	32	RW	00000000h
c0h	Port Control Register (PCR16)	32	RW	00000000h
c4h	Port Control Register (PCR17)	32	RW	00000000h
c8h	Port Control Register (PCR18)	32	RW	00000000h
cch	Port Control Register (PCR19)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
d0h	Port Control Register (PCR20)	32	RW	00000000h
d4h	Port Control Register (PCR21)	32	RW	00000000h
d8h	Port Control Register (PCR22)	32	RW	00000000h
dch	Port Control Register (PCR23)	32	RW	00000000h
e0h	Port Control Register (PCR24)	32	RW	00000000h
e4h	Port Control Register (PCR25)	32	RW	00000000h
e8h	Port Control Register (PCR26)	32	RW	00000000h
ech	Port Control Register (PCR27)	32	RW	00000000h
f0h	Port Control Register (PCR28)	32	RW	00000000h
f4h	Port Control Register (PCR29)	32	RW	00000000h
f8h	Port Control Register (PCR30)	32	RW	00000000h
fch	Port Control Register (PCR31)	32	RW	00000000h

8.2.1.1 GPIO PDOR Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PDO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.2: GPIO PDOR Register Description

Field	Function
31 - 0 PDO	**Port Data Output** When configured for general-purpose output, it shows the logic level on the corresponding pin.

8.2.1.2 GPIO PSOR Register

Offset: 4h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	PSO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	PSO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.3: GPIO PSOR Register Description

Field	Function
31 - 0 PSO	**Port Set output** Writing to this register will update the contents of the corresponding bit in PDOR. 0b - No change. 1b - Corresponding bit in PDOR is set to logic 1.

8.2.1.3 GPIO PCOR Register

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	PCO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	PCO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.4: GPIO PCOR Register Description

Field	Function
31 - 0 PCO	**Port Clear Output** Writing to this register will update the contents of the corresponding bit in PDOR. 0b - No change. 1b - Corresponding bit in PDOR is set to logic 0.

Yuntu Microelectronics

8.2.1.4 GPIO PTOR Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	PTO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	PTO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.5: GPIO PTOR Register Description

Field	Function
31 - 0 PTO	**Port Toggle Output** Writing to this register will update the contents of the corresponding bit in PDOR. 0b - No change. 1b - Corresponding bit in PDOR is set to the inverse of its existing logic state.

8.2.1.5 GPIO PDIR Register

Offset: 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PDI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.6: GPIO PDIR Register Description

Field	Function
31 - 0 PDI	**Port Data Input** When configured for digital function, it shows the logic level on the corresponding pin. When configured for analog function, it always reads 0.

Yuntu Microelectronics

8.2.1.6 GPIO POER Register

Offset: 14h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.7: GPIO POER Register Description

Field	Function
31 - 0 POE	**Port Output Enable** Configure the corresponding pin for output. 0b - Port output disable 1b - Port output enable

8.2.1.7 GPIO PIER Register

Offset: 18h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PIE															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PIE															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 8.8: GPIO PIER Register Description

Field	Function
31 - 0 PIE	**Port Input Enable** Configure the corresponding pin for input. 0b - Port input disable 1b - Port input enable

Yuntu Microelectronics

8.2.1.8 GPIO PIFR Register

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PIF															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PIF															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 8.9: GPIO PIFR Register Description

Field	Function
31 - 0 PIF	**Port Interrupt Status Flag** Indicate the detection for the defined event in PCTRL[IRQC] of the corresponding pin. 0b - No event detected. 1b - The event defined in PCTRL[IRQC] is detected.

8.2.1.9 GPIO PCR Register

Offset: 80h + n * 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R												INVE	IRQC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 8.10: GPIO PCR Register Description

Field	Function
4 INVE	**Invert Enable** Invert the input logic level when configured for digital function.
3 - 0 IRQC	**Interrupt Configuration** 0000b - Disabled. 0001b - DMA request on rising edge. 0010b - DMA request on falling edge. 0011b - DMA request on either edge. 0100b - Reserved. 0101b - Reserved. 0110b - Reserved. 0111b - Reserved. 1000b - Interrupt when logic 0. 1001b - Interrupt on rising-edge. 1010b - Interrupt on falling-edge. 1011b - Interrupt on either edge. 1100b - Interrupt when logic 1. 1101b - Reserved 1110b - Reserved 1111b - Reserved

8.3 Functional description

8.3.1 General-purpose input

Reading the PDIR register gets the logic state of GPIO pins when the pin is configured as GPIO input enable. The PDIR register supports getting the logic state while the port configured as other digital function.

8.3.2 General-purpose output

Writing the PDOR registers controls the logic state of GPIO pins when the pin is configured as GPIO output enable.

The GPIO module supports bit manipulation on the GPIO outputs including pin data set, pin data clear and pin data toggle.

8.3.3 Port interrupt and DMA request

Each of port pin can generate DMA request or interrupt with controlled by PCR[IRQC].

Yuntu Microelectronics

Chapter 9

Port Control (PCTRL)

9.1 Chip-specific Information

9.1.1 Pin Property

Table 9.1: Pin Property

176 LQFP	144 LQFP	100 LQFP	Pin Name	Default Function	Default Status	Support Pullup/ Pulldown	Support Slew Rate	Support High Drive	Pin Filter	PD Wakeup Pin
1	-	-	PTA_18	-	Hi-Z	Y	Y	-	Digital filter	-
2	-	-	PTA_19	-	Hi-Z	Y	Y	-	Digital filter	-
3	-	-	PTA_20	-	Hi-Z	Y	Y	-	Digital filter	-
4	1	1	PTE_16	ADC0_S28	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[32]
5	2	2	PTE_15	ADC0_S27	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[33]
6	-	-	PTA_21	-	Hi-Z	Y	Y	-	Digital filter	-
7	3	3	VDD11	VDD11	-	-	-	-	-	-
8	4	4	VDD25	VDD25	-	-	-	-	-	-
9	-	-	PTA_22	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[34]
10	5	5	PTE_11	ADC0_S30	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[0]
11	6	6	PTE_10	ADC0_S29	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[1]
12	7	7	PTE_13	-	Hi-Z	Y	Y	-	Digital filter	-
13	-	-	PTA_23	-	Hi-Z	Y	Y	-	Digital filter	-
14	8	8	PTE_5	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[2]
15	9	9	PTE_4	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[3]
16	-	-	PTA_24	-	Hi-Z	Y	Y	-	Digital filter	-
17	10	-	PTA_25	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[57]
18	11	10	VDDA	Power	-	-	-	-	-	-
19	12	-	VSS	Ground	-	-	-	-	-	-
20	13	11	VDDA	ADC Power	-	-	-	-	-	-
21	14	12	VREFH	ADC/ACMP Reference High	-	-	-	-	-	-
22	15	13	VREFL	ADC/ACMP Reference Low	-	-	-	-	-	-
23	16	14	VSSA	Ground	-	-	-	-	-	-
24	17	15	PTB_7	EXTAL	Hi-Z	Y	Y	-	Digital filter	-
25	18	16	PTB_6	XTAL	Hi-Z	Y	Y	-	Digital filter	-
26	19	-	PTA_26	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[61]
27	20	17	PTE_14	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[35]
28	21	18	PTE_3	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[4]
29	22	-	PTA_27	-	Hi-Z	Y	Y	-	Digital filter	-
30	23	19	PTE_12	DAC1_OUT	Hi-Z	Y	Y	-	Digital filter	-
31	24	-	PTA_28	ACMP1_IN5	Hi-Z	Y	Y	-	Digital filter	-
32	25	20	PTD_17	-	Hi-Z	Y	Y	-	Digital filter	-
33	26	-	PTA_29	-	Hi-Z	Y	Y	-	Digital filter	-
34	27	-	PTA_30	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[58]
35	28	21	PTD_16	EXTAL32	Hi-Z	Y	Y	-	Digital filter	-
36	29	22	PTD_15	XTAL32	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[36]
37	30	23	PTE_9		Hi-Z	Y	Y	-	Digital filter	WKU_PIN[7]
38	31	-	VSS	Power	-	-	-	-	-	-
39	32	-	VDDB	Ground	-	-	-	-	-	-
40	33	-	PTA_31	-	Hi-Z	Y	Y	-	Digital filter	-
41	34	24	PTD_14	-	Hi-Z	Y	Y	-	Digital filter	-
42	35	25	PTD_13	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[37]
43	36	-	PTB_18	ADC0_S16	Hi-Z	Y	Y	-	Digital filter	-
44	-	-	PTB_19	-	Hi-Z	Y	Y	-	Digital filter	-
45	37	-	PTB_20	ADC0_S17	Hi-Z	Y	Y	-	Digital filter	-
46	38	-	PTB_21	ADC0_S18	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[38]
47	39	26	PTE_8	ACMP0_IN3	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[8]

Yuntu Microelectronics

Table 9.1 continued from previous page

176 LQFP	144 LQFP	100 LQFP	Pin Name	Default Function	Default Status	Support Pullup/ Pulldown	Support Slew Rate	Support High Drive	Pin Filter	PD Wakeup Pin
48	40	27	PTB_5	-	Hi-Z	Y	Y	Y	Digital filter	-
49	41	28	PTB_4	ACMP1_IN4	Hi-Z	Y	Y	Y	Digital filter	-
50	42	29	PTC_3	ADC0_S11 /ACMP0_IN4	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[9]
51	43	30	PTC_2	ADC0_S10 /ACMP0_IN5	Hi-Z	Y	Y	Y	Digital filter	-
52	44	31	PTD_7	ACMP0_IN6	Hi-Z	Y	Y	Y	Digital filter	-
53	45	32	PTD_6	ACMP_IN7	Hi-Z	Y	Y	-	Digital filter	-
54	46	33	PTD_5	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[11]
55	47	34	PTD_12	-	Hi-Z	Y	Y	Y	Digital filter	-
56	48	35	PTD_11	-	Hi-Z	Y	Y	Y	Digital filter	-
57	49	36	PTD_10	-	Hi-Z	Y	Y	Y	Digital filter	-
58	50	37	VSS	Ground	-	-	-	-	-	-
59	51	38	VDDB	Power	-	-	-	-	-	-
60	52	39	PTC_1	ADC0_S9	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[12]
61	53	40	PTC_0	ADC0_S8	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[13]
62	54	41	PTD_9	-	Hi-Z	Y	Y	-	Digital filter	-
63	55	42	PTD_8	-	Hi-Z	Y	Y	-	Digital filter	-
64	56	43	PTC_17	ADC0_S15	Hi-Z	Y	Y	Y	Digital filter	-
65	57	44	PTC_16	ADC0_S14	Hi-Z	Y	Y	Y	Digital filter	-
66	58	-	PTB_22	ADC0_S19	Hi-Z	Y	Y	-	Digital filter	-
67	59	45	PTC_15	ADC0_S13	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[14]
68	60	-	PTB_23	ADC0_S20	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[39]
69	-	-	PTB_24	-	Hi-Z	Y	Y	-	Digital filter	-
70	61	46	PTC_14	ADC0_S12	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[15]
71	62	-	PTB_25	ADC0_S21	Hi-Z	Y	Y	-	Digital filter	-
72	-	-	PTB_26	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[10]
73	63	47	PTB_3	ADC0_S7	Hi-Z	Y	Y	-	Digital filter	-
74	64	-	PTB_27	ADC0_S22	Hi-Z	Y	Y	-	Digital filter	-
75	65	-	PTB_28	ADC0_S23	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[59]
76	66	-	VSS	Ground	-	-	-	-	-	-
77	67	-	VDDB	Power	-	-	-	-	-	-
78	68	48	PTB_2	ADC0_S6	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[40]
79	69	-	PTB_29	-	Hi-Z	Y	Y	-	Digital filter	-
80	-	-	PTB_30	-	Hi-Z	Y	Y	-	Digital filter	-
81	70	49	PTC_13	-	Hi-Z	Y	Y	-	Digital filter	-
82	-	-	PTB_31	-	Hi-Z	Y	Y	-	Digital filter	-
83	-	-	PTC_18	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[60]
84	71	50	PTC_12	-	Hi-Z	Y	Y	-	Digital filter	-
85	72	-	PTC_19	-	Hi-Z	Y	Y	-	Digital filter	-
86	-	-	PTC_20	-	Hi-Z	Y	Y	-	Digital filter	-
87	-	-	PTC_21	-	Hi-Z	Y	Y	-	Digital filter	-
88	-	-	PTC_22	-	Hi-Z	Y	Y	-	Digital filter	-
89	73	-	PTC_23	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[62]
90	-	-	PTC_24	-	Hi-Z	Y	Y	-	Digital filter	-
91	-	-	PTC_25	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[41]
92	74	51	PTC_11	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[42]
93	-	-	PTC_26	-	Hi-Z	Y	Y	-	Digital filter	-
94	75	52	PTC_10	ADC1_S27	Hi-Z	Y	Y	-	Digital filter	-
95	76	-	PTC_27	-	Hi-Z	Y	Y	-	Digital filter	-
96	77	53	PTB_1	ADC0_S5	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[16]
97	78	54	PTB_0	ADC0_S4	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[17]
98	79	-	PTC_28	-	Hi-Z	Y	Y	-	Digital filter	-
99	80	55	PTC_9	ACMP1_IN6	Hi-Z	Y	Y	Y	Digital filter	WKU_PIN[43]
100	81	56	PTC_8	DAC0_OUT	Hi-Z	Y	Y	Y	Digital filter	-
101	82	-	PTC_29	-	Hi-Z	Y	Y	-	Digital filter	-
102	83	57	PTA_7	ADC0_S3	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[18]
103	84	-	PTC_30	-	Hi-Z	Y	Y	-	Digital filter	-
104	85	58	PTA_6	ADC0_S2	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[19]
105	86	-	PTC_31	-	Hi-Z	Y	Y	-	Digital filter	-
106	87	59	PTE_7	-	Hi-Z	Y	Y	-	Digital filter	-
107	88	-	PTD_18	ADC1_S16	Hi-Z	Y	Y	-	Digital filter	-
108	89	-	PTD_19	ADC1_S17	Hi-Z	Y	Y	-	Digital filter	-
109	90	60	VSS	Ground	-	-	-	-	-	-
110	91	61	VDDB	Power	-	-	-	-	-	-
111	92	62	PTA_17	-	Hi-Z	Y	Y	-	Digital filter	-
112	93	63	PTB_17	ADC1_S26	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[44]
113	-	-	PTD_20	ADC1_S31	Hi-Z	Y	Y	-	Digital filter	-
114	94	64	PTB_16	ADC1_S15	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[53]
115	95	65	PTB_15	ADC1_S14	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[54]
116	96	66	PTB_14	ADC1_S9	Hi-Z	Y	Y	-	Digital filter	-
117	-	-	PTD_21	-	Hi-Z	Y	Y	-	Digital filter	-
118	97	67	PTB_13	ADC1_S8	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[20]
119	98	68	PTB_12	ADC1_S7	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[21]
120	99	-	PTD_22	ADC1_S18	Hi-Z	Y	Y	-	Digital filter	-
121	100	69	PTD_4	ADC1_S6	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[45]
122	101	70	PTD_3	ADC1_S3	Hi-Z	Y	Y	-	Passive/ Digital filter	WKU_PIN[22]
123	102	71	PTD_2	ADC1_S2	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[23]
124	103	-	PTD_23	ADC1_S19	Hi-Z	Y	Y	-	Digital filter	-
125	104	72	PTA_3	ADC1_S1	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[24]
126	105	73	PTA_2	ADC1_S0 /ACMP1_IN2	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[25]
127	106	-	PTD_24	ADC1_S20	Hi-Z	Y	Y	-	Digital filter	-
128	107	74	PTB_11	ADC1_S30	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[63]
129	-	-	VDDA	Power	-	-	-	-	-	-

Yuntu Microelectronics

Table 9.1 continued from previous page

176 LQFP	144 LQFP	100 LQFP	Pin Name	Default Function	Default Status	Support Pullup/ Pulldown	Support Slew Rate	Support High Drive	Pin Filter	PD Wakeup Pin
130	-	-	VSS	Ground	-	-	-	-	-	-
131	108	75	PTB_10	ADC1_S29	Hi-Z	Y	Y	-	Digital filter	-
132	-	-	PTD_25	-	Hi-Z	Y	Y	-	Digital filter	-
133	109	76	PTB_9	ADC0_S25	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[46]
134	-	-	PTD_26	-	Hi-Z	Y	Y	-	Digital filter	-
135	110	-	PTD_27	ADC1_S21	Hi-Z	Y	Y	-	Digital filter	-
136	111	77	PTB_8	ADC0_S24	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[47]
137	112	-	PTD_28	ADC1_S22	Hi-Z	Y	Y	-	Digital filter	-
138	113	78	PTA_1	ADC0_S1 /ACMP_IN1	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[55]
139	114	-	PTD_29	ADC1_S23	Hi-Z	Y	Y	-	Digital filter	-
140	115	79	PTA_0	ADC0_S0 /ACMP_IN0	Hi-Z	Y	Y	-	Digital filter	-
141	116	-	PTD_30	-	Hi-Z	Y	Y	-	Digital filter	-
142	-	-	PTD_31	-	Hi-Z	Y	Y	-	Digital filter	-
143	117	80	PTC_7	ADC1_S5	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[26]
144	118	81	PTC_6	ADC1_S4	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[27]
145	-	-	PTE_17	-	Hi-Z	Y	Y	-	Digital filter	-
146	119	82	PTA_16	ADC1_S13	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[56]
147	-	-	PTE_18	-	Hi-Z	Y	Y	-	Digital filter	-
148	120	83	PTA_15	ADC1_S12	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[48]
149	121	84	PTE_6	ADC1_S11	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[28]
150	122	85	PTE_2	ADC1_S10	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[29]
151	123	86	VSS	Ground	-	-	-	-	-	-
152	124	87	VDDA	Power	-	-	-	-	-	-
153	125	-	PTE_19	-	Hi-Z	Y	Y	-	Digital filter	-
154	126	-	PTE_20	-	Hi-Z	Y	Y	-	Digital filter	-
155	127	88	PTA_14	ADC1_S28	Hi-Z	Y	Y	-	Digital filter	-
156	128	-	PTE_21	-	Hi-Z	Y	Y	-	Digital filter	-
157	129	-	PTE_22	-	Hi-Z	Y	Y	-	Digital filter	-
158	130	89	PTA_13	ADC1_S25	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[30]
159	131	-	PTE_23	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[49]
160	132	-	PTE_24	-	Hi-Z	Y	Y	-	Digital filter	-
161	133	-	PTE_25	-	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[50]
162	134	90	PTA_12	ADC1_S24	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[31]
163	135	91	PTA_11	-	Hi-Z	Y	Y	Y	Digital filter	-
164	136	92	PTA_10	JTAG_TDO _SWD_SWO	Output Low	Y	Y	Y	Digital filter	-
165	137	93	PTE_1	-	Hi-Z	Y	Y	Y	Digital filter	WKU_PIN[5]
166	138	94	PTE_0	-	Hi-Z	Y	Y	Y	Digital filter	WKU_PIN[6]
167	-	-	PTE_26	-	Hi-Z	Y	Y	-	Digital filter	-
168	139	95	PTC_5	JTAG_TDI	Input, Pullup	Y	Y	-	Digital filter	-
169	140	96	PTC_4	JTAG_TCK _SWD_CLK	Input, Pulldown	Y	Y	-	Digital filter	-
170	141	97	PTA_5	RCU_RESET_b	Input, Pullup	Y	Y	-	Passive/ Digital filter	RESET_B
171	-	-	VSS	Ground	-	-	-	-	-	-
172	-	-	VDDA	Power	-	-	-	-	-	-
173	142	98	PTA_4	JTAG_TMS _SWD_IO	Input, Pullup	Y	Y	-	Digital filter	-
174	-	-	PTE_27	-	Hi-Z	Y	Y	-	Digital filter	-
175	143	99	PTA_9	ADC0_S31	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[51]
176	144	100	PTA_8	ADC0_S26	Hi-Z	Y	Y	-	Digital filter	WKU_PIN[52]

9.2 Introduction

9.2.1 Overview

The Port Control (PCTRL) module provides support for port control. The functions should be configured independently for each pin in the 32-bit port and affect the pin regardless of its pin muxing state. Not all pins within each port are implemented.

9.2.2 Features

The PCTRL module has the following port control features:

- Individual pull control fields with pullup, pulldown, and pull-disable support
- Individual slew rate field supporting fast and slow slew rates
- Individual input passive filter field supporting enable and disable of the individual input passive filter on selected pins

Yuntu Microelectronics

- Individual mux control field supporting analog or pin disabled, GPIO, and up to 6 chip-specific digital functions
- Individual digital filter for data inputs
- Individual lock function to avoid misoperation

9.2.3 Modes of Operation

9.2.3.1 Active Mode

In Active mode, the PCTRL module operates normally.

9.2.3.2 Sleep Mode

In Sleep mode, the PCTRL module continues to operates normally.

9.2.3.3 Deepsleep Mode

In Deepsleep mode, the PCTRL module will keep registers settings and operates normally.

9.2.3.4 Standby Mode

In Standby mode, the PCTRL module will keep registers settings and operates normally.

9.2.3.5 Powerdown Mode

In Powerdown mode, the PCTRL will be reset when waking up from Powerdown mode. Pin configuration like pull up/pull down/slew rate setting will be kept under Powerdown and operates normally.

9.3 Register Definition

9.3.1 PCTRL Memory Map

PCTRLA base address 40048000h
PCTRLB base address 4004c000h
PCTRLC base address 40050000h
PCTRLD base address 40054000h
PCTRLE base address 40058000h

Table 9.2: PCTRL Memory Map

Offset	Register	Width	Access	Reset Value
0h	Port Control Register (PCR0)	32	RW	see description
4h	Port Control Register (PCR1)	32	RW	see description
8h	Port Control Register (PCR2)	32	RW	see description
ch	Port Control Register (PCR3)	32	RW	see description
10h	Port Control Register (PCR4)	32	RW	see description
14h	Port Control Register (PCR5)	32	RW	see description
18h	Port Control Register (PCR6)	32	RW	see description
1ch	Port Control Register (PCR7)	32	RW	see description
20h	Port Control Register (PCR8)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
24h	Port Control Register (PCR9)	32	RW	see description
28h	Port Control Register (PCR10)	32	RW	see description
2ch	Port Control Register (PCR11)	32	RW	see description
30h	Port Control Register (PCR12)	32	RW	see description
34h	Port Control Register (PCR13)	32	RW	see description
38h	Port Control Register (PCR14)	32	RW	see description
3ch	Port Control Register (PCR15)	32	RW	see description
40h	Port Control Register (PCR16)	32	RW	see description
44h	Port Control Register (PCR17)	32	RW	see description
48h	Port Control Register (PCR18)	32	RW	see description
4ch	Port Control Register (PCR19)	32	RW	see description
50h	Port Control Register (PCR20)	32	RW	see description
54h	Port Control Register (PCR21)	32	RW	see description
58h	Port Control Register (PCR22)	32	RW	see description
5ch	Port Control Register (PCR23)	32	RW	see description
60h	Port Control Register (PCR24)	32	RW	see description
64h	Port Control Register (PCR25)	32	RW	see description
68h	Port Control Register (PCR26)	32	RW	see description
6ch	Port Control Register (PCR27)	32	RW	see description
70h	Port Control Register (PCR28)	32	RW	see description
74h	Port Control Register (PCR29)	32	RW	see description
78h	Port Control Register (PCR30)	32	RW	see description
7ch	Port Control Register (PCR31)	32	RW	see description

9.3.1.1 PCTRL PCR Register

Port control register

Offset: 0h + n * 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK										DFW					DFE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					MUX					DSE		PFE		SRE	PE	PS
W																
Reset	0	0	0	0	u	u	u	u	0	u	0	0	0	u	u	u

Yuntu Microelectronics

Table 9.3: PCTRL PCR Register Description

Field	Function
31 LOCK	**Lock Enable** Lock Enable configuration for digital pin muxing. 0b - Lock disable. 1b - Lock enable, PCR's other bits and related GPIO's output register(PDOR,PSOR,PCOR,PTOR,POER) will be locked until write 0 to this bit.
21 - 17 DFW	**Digital Filter Width** The digital filter configuration is valid in all digital pin muxing modes. Glitches that are longer than this register setting will pass through the digital filter, and glitches that are equal to or less than this register setting are filtered. Changing the filter length must be done only after all filters are disabled.
16 DFE	**Digital Filter Enable** **NOTE**. when using digital filter under low power mode, make sure filter source clock is available under low power mode.
11 - 8 MUX	**Port Function Selection** Not all pin muxing slots are implemented. Unimplemented pin muxing slots are reserved. Please refer to pinmux table for detailed peripheral functions. The corresponding pin is configured in the following pin muxing slot as follows: 0000b - Alternative 0, Analog function, digital function is disabled. 0001b - Alternative 1, GPIO function. 0010b - Alternative 2. 0011b - Alternative 3. 0100b - Alternative 4. 0101b - Alternative 5. 0110b - Alternative 6. 0111b - Alternative 7. 1000b - Alternative 8. 1001b - Alternative 9. 1010b - Alternative 10. 1011b - Alternative 11. 1100b - Alternative 12. 1101b - Alternative 13. 1110b - Alternative 14. 1111b - Alternative 15.
6 DSE	**Drive Strength Enable** Drive strength configuration for digital pin muxing. 0b - Low drive strength, if the pin is configured as a digital output. 1b - High drive strength, if the pin is configured as a digital output.
4	**Passive Filter Enable**

Yuntu Microelectronics

Field	Function
PFE	Passive filter configuration for digital pin muxing. 0b - Passive input filter is disabled. 1b - Passive input filter is enabled, if the pin is configured as a digital input.
2 SRE	**Slew Rate Enable** Slew rate configuration for digital pin muxing. 0b - Fast slew rate, if the pin is configured as a digital output. 1b - Slow slew rate, if the pin is configured as a digital output.
1 PE	**Pull Enable** Pull configuration for digital pin muxing. 0b - Internal pullup or pulldown resistor is not enabled. 1b - Internal pullup or pulldown resistor is enabled.
0 PS	**Pull Select** Pull configuration for digital pin muxing. 0b - Internal pulldown resistor is enabled, if PE field is set. 1b - Internal pullup resistor is enabled, if PE field is set.

9.4 Functional description

9.4.1 Port Control

Each port pin has a corresponding Port Control register. The Port Control register configures the following functions for each pin within the 32-bit port.

- Pullup or pulldown enable
- Slew rate configuration
- Passive input filter enable on selected pins
- Pin Muxing mode
- Digital filter configuration
- Lock function

Yuntu Microelectronics

Part IV

Core and System Modules

Yuntu Microelectronics

Chapter 10

CM7 Overview

10.1 ARM Cortex-M7 Introduction

The Cortex-M7 processor is a low gate count, highly energy efficient processor that is intended for microcontroller and deeply embedded applications. The processor is based on the Armv7-M architecture and is primarily for use in environments where security is an important consideration.

For detailed information:

- Cortex-M7 processor, see Arm Cortex-M7 Processor Technical Reference Manual
- Cortex-M7 peripheral and control, see Arm Cortex-M7 Devices Generic User Guide
- Cortex-M7 Datasheet, see Arm Cortex-M7 Processor Datasheet

10.1.1 Features

- An in-order issue, super-scalar pipeline with dynamic branch prediction
- DSP extensions
- The ARMv7-M Thumb instruction set, defined in the Arm v7-M Architecture Reference Manual
- Banked Stack Pointer (SP)
- Hardware integer divide instructions, SDIV and UDIV
- Handler and Thread modes
- Automatic processor state saving and restoration for low-latency Interrupt Service Routine (ISR) entry and exit
- Support for ARMv7-M unaligned accesses
- Low-latency interrupt processing
- A low-cost debug solution with CoreSight components
  - Support breakpoints
  - Support watchpoints, tracing, and system profiling
  - Support printf() style debugging through an Instrumentation Trace Macrocell (ITM)
  - Support Trace Port Interface Unit (TPIU)
  - Support Debug Access Port (DAP)
- Support ETM (Embedded Trace Macrocell)
- Support Floating Point Unit(FPU)
- Support WIC
- Support memory interfaces that include
  - Harvard architecture-based instruction and data caches, and an AXIM interface

Yuntu Microelectronics

- A dedicated low-latency AHBP interface
- A 64-bit AXI AMBA4 memory interface with a 8 KB instruction cache and an 8 KB data cache for efficient access to external resources. The instruction and data caches are ECC protected.
- A 32-bit AHBS for interfacing with slaves
- 64-bit and 32-bit memory interfaces for the connection to local Tightly Coupled Memories called ITCM and DTCM
- Cortex-M7 core interfaces with low-latency 32 KB ITCM, 64 KB DTCM0 and 64KB DTCM1
- Support 16 regions MPU

10.1.2 Buses, Interconnects, and Interfaces

This table discusses Cortex-M7 buses and their associated interconnects and interfaces.

Table 10.1: Buses and Associated Information

Bus name	Description
AXIM	Using the XHB400 module, this bus is first translated to the AHB-Lite bus that interfaces with the AXBS crossbar switch providing high-bandwidth access to on-chip memories and peripherals.
AHBP	This bus connects to the AXBS crossbar switch providing high-bandwidth access to on-chip peripherals.
PPB	This bus provides access to these modules: · Arm modules such as NVIC, ETM, ITM, DWT and ROM tables

10.1.3 Core Configuration

This table describes Cortex-M7 parameter settings.

Table 10.2: Core Configuration

Parameter	Configuration
FPU	Single precision
DSP extension instructions	· Single cycle 16/32-bit MAC · Single cycle dual 16-bit MAC · 8/16-bit SIMD arithmetic · Hardware divide (2-12 cycles)
Armv7-M security extensions	Not implemented
I-cache size	8 KB
D-cache size	8 KB
Caches ECC	Implemented
On core MPU region	16
Number of IRQs	240
IRQ priority width configuration	4 (16 interrupt priority levels)
Debug (breakpoint/watchpoint)	Full comparator set: 4 DWT and 8 FPB comparators
Internal trace support	ITM and DWT trace functionality implemented
ETM support	Instruction ETM interface implemented
CTI	Implemented
WIC support	WIC implemented
Dual-redundant core (lock-step) CPU functionality	Lock-step implemented
RAR	All asynchronously reset
ITCM	32 KB for CM7
DTCM	128 KB for CM7

10.2 Nested Vectored Interrupt Controller (NVIC)

Table 10.3: Interrupt Table

Vector	IRQ	Module	Description	Async[1]	Interrupt
0	-	ARM_CORE	SP	-	__StackTop
1	-	ARM_CORE	PC	-	Reset

Yuntu Microelectronics

Table 10.3 continued from previous page

Vector	IRQ	Module	Description	Async[1]	Interrupt
2	-	ARM_CORE	NMI	Y	NMI
3	-	ARM_CORE	Hard Fault	-	HardFault
4	-	ARM_CORE	MemManage Fault	-	MemManage
5	-	ARM_CORE	Bus Fault	-	BusFault
6	-	ARM_CORE	Usage Fault	-	UsageFault
7	-	-	-	-	-
8	-	-	-	-	-
9	-	-	-	-	-
10	-	-	-	-	-
11	-	ARM_CORE	Supervisor call (SVCall)	-	SVC
12	-	ARM_CORE	Debug Monitor	-	DebugMon
13	-	-	-	-	-
14	-	ARM_CORE	Pendable request for system service (PendableSrvReq)	-	PendSV
15	-	ARM_CORE	System tick timer (SysTick)	-	SysTick
16	0	DMA	DMA channel 0 transfer complete	-	DMA0
17	1	DMA	DMA channel 1 transfer complete	-	DMA1
18	2	DMA	DMA channel 2 transfer complete	-	DMA2
19	3	DMA	DMA channel 3 transfer complete	-	DMA3
20	4	DMA	DMA channel 4 transfer complete	-	DMA4
21	5	DMA	DMA channel 5 transfer complete	-	DMA5
22	6	DMA	DMA channel 6 transfer complete	-	DMA6
23	7	DMA	DMA channel 7 transfer complete	-	DMA7
24	8	DMA	DMA channel 8 transfer complete	-	DMA8
25	9	DMA	DMA channel 9 transfer complete	-	DMA9
26	10	DMA	DMA channel 10 transfer complete	-	DMA10
27	11	DMA	DMA channel 11 transfer complete	-	DMA11
28	12	DMA	DMA channel 12 transfer complete	-	DMA12
29	13	DMA	DMA channel 13 transfer complete	-	DMA13
30	14	DMA	DMA channel 14 transfer complete	-	DMA14
31	15	DMA	DMA channel 15 transfer complete	-	DMA15
32	16	DMA	DMA channel 16 transfer complete	-	DMA16
33	17	DMA	DMA channel 17 transfer complete	-	DMA17
34	18	DMA	DMA channel 18 transfer complete	-	DMA18
35	19	DMA	DMA channel 19 transfer complete	-	DMA19
36	20	DMA	DMA channel 20 transfer complete	-	DMA20
37	21	DMA	DMA channel 21 transfer complete	-	DMA21
38	22	DMA	DMA channel 22 transfer complete	-	DMA22
39	23	DMA	DMA channel 23 transfer complete	-	DMA23
40	24	DMA	DMA channel 24 transfer complete	-	DMA24
41	25	DMA	DMA channel 25 transfer complete	-	DMA25
42	26	DMA	DMA channel 26 transfer complete	-	DMA26
43	27	DMA	DMA channel 27 transfer complete	-	DMA27
44	28	DMA	DMA channel 28 transfer complete	-	DMA28
45	29	DMA	DMA channel 29 transfer complete	-	DMA29
46	30	DMA	DMA channel 30 transfer complete	-	DMA30
47	31	DMA	DMA channel 31 transfer complete	-	DMA31
48	32	DMA	DMA error interrupt channels 0-32	-	DMA_Error
49	33	CM7 FPU	FPU ORed sources (Input Denormal, Inexact, Underflow, Overflow, Divide-by-Zero, Invalid Operand)	-	FPU
50	34	Flash	Command complete	-	EFM
51	35	Flash	Read collision	-	EFM_Error
52	36	PCU	Low voltage detect interrupt	Y	PCU
53	37	Flash	Flash ECC error interrupt	-	EFM_ECC
54	38	WDG	Interrupt request out before wdg reset out	Y	WDG
55	39	-	Reserved	-	-
56	40	I2C0	LPI2C Interrupt	Y	I2C0_Master_Slave
57	41	SPI0	SPI Interrupt	Y	SPI0
58	42	SPI1	SPI Interrupt	Y	SPI1
59	43	SPI2	SPI Interrupt	Y	SPI2
60	44	I2C1	LPI2C Interrupt	Y	I2C1_Master_Slave
61	45	LINFlexD0	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD0
62	46	-	-	-	-

Yuntu Microelectronics

Table 10.3 continued from previous page

Vector	IRQ	Module	Description	Async[1]	Interrupt
63	47	LINFlexD1	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD1
64	48	-	-	-	-
65	49	LINFlexD2	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD2
66	50	-	-	-	-
67	51	QSPI	TX Buffer Fill interrupt Transfer Complete / Transaction Finished RX Buffer Drain interrupt Buffer Overflow / Underrun interrupt Serial Flash Communication Error interrupt All interrupts ORed output	-	QSPI
68	52	-	-	-	-
69	53	ADC0	ADC interrupt request	Y	ADC0
70	54	ADC1	ADC interrupt request	Y	ADC1
71	55	ACMP0	ACMP interrupt	Y	ACMP0
72	56	ACMP1	ACMP interrupt	Y	ACMP1
73	57	-	-	-	-
74	58	EMU	Single bit error correction Double bit error non-correctable	-	EMU
75	59	-	-	-	-
76	60	RTC	RTC alarm interrupt	Y	RTC
77	61	RTC	RTC seconds interrupt	Y	RTC_Seconds
78	62	pTMR0	Periodic timer 0 channel0 interrupt	-	pTMR0_Ch0
79	63	pTMR0	Periodic timer 0 channel1 interrupt	-	pTMR0_Ch1
80	64	pTMR0	Periodic timer 0 channel2 interrupt	-	pTMR0_Ch2
81	65	pTMR0	Periodic timer 0 channel3 interrupt	-	pTMR0_Ch3
82	66	PTU0	PTU interrupt	-	PTU0
83	67	pTMR1	Periodic timer 1 channel0 interrupt	-	pTMR1_Ch0
84	68	pTMR1	Periodic timer 1 channel1 interrupt	-	pTMR1_Ch1
85	69	pTMR1	Periodic timer 1 channel2 interrupt	-	pTMR1_Ch2
86	70	pTMR1	Periodic timer 1 channel3 interrupt	-	pTMR1_Ch3
87	71	SCU(CMU)	SCU(CMU) interrupt or PLL loss of lock	Y	CMU
88	72	lpTMR	Low Power Timer interrupt interrupt	Y	lpTMR0
89	73	PORTA	PORTA Interrupt	Y	GPIOA
90	74	PORTB	PORTB Interrupt	Y	GPIOB
91	75	PORTC	PORTC Interrupt	Y	GPIOC
92	76	PORTD	PORTD Interrupt	Y	GPIOD
93	77	PORTE	PORTE Interrupt	Y	GPIOE
94	78	-	-	-	-
95	79	-	-	-	-
96	80	-	-	-	-
97	81	-	-	-	-
98	82	PTU1	PTU interrupt	-	PTU1
99	83	-	-	-	-
100	84	CAN0	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN0_ORed
101	85	CAN0	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN0_Error
102	86	CAN0	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN0_Wake_Up
103	87	CAN0	ORed Message buffer (0-15)	-	CAN0_ORed_0_15_MB
104	88	CAN0	ORed Message buffer (16-31)	-	CAN0_ORed_16_31_MB
105	89	CAN0	ORed Message buffer (32-47)	-	CAN0_ORed_32_47_MB

Yuntu Microelectronics

Table 10.3 continued from previous page

Vector	IRQ	Module	Description	Async[1]	Interrupt
106	90	CAN0	ORed Message buffer (48-63)	-	CAN0_ORed_48_63_MB
107	91	CAN0	ORed Message buffer (64-79)	-	CAN0_ORed_64_79_MB
108	92	CAN0	ORed Message buffer (80-95)	-	CAN0_ORed_32_47_MB
109	93	CAN0	ORed Message buffer (96-111)	-	CAN0_ORed_96_111_MB
110	94	CAN0	ORed Message buffer (112-127)	-	CAN0_ORed_112_127_MB
111	95	CAN1	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN1_ORed
112	96	CAN1	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN1_Error
113	97	CAN1	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN1_Wake_Up
114	98	CAN1	ORed Message buffer (0-15)	-	CAN1_ORed_0_15_MB
115	99	CAN1	ORed Message buffer (16-31)	-	CAN1_ORed_16_31_MB
116	100	CAN1	ORed Message buffer (32-47)	-	CAN1_ORed_32_47_MB
117	101	CAN1	ORed Message buffer (48-63)	-	CAN1_ORed_48_63_MB
118	102	CAN1	ORed Message buffer (64-79)	-	CAN1_ORed_64_79_MB
119	103	CAN1	ORed Message buffer (80-95)	-	CAN1_ORed_32_47_MB
120	104	CAN1	ORed Message buffer (96-111)	-	CAN1_ORed_96_111_MB
121	105	CAN1	ORed Message buffer (112-127)	-	CAN1_ORed_112_127_MB
122	106	CAN2	OR＇ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN2_ORed
123	107	CAN2	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN2_Error
124	108	CAN2	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN2_Wake_Up
125	109	CAN2	ORed Message buffer (0-15)	-	CAN2_ORed_0_15_MB
126	110	CAN2	ORed Message buffer (16-31)	-	CAN2_ORed_16_31_MB
127	111	CAN2	ORed Message buffer (32-47)	-	CAN2_ORed_32_47_MB
128	112	CAN2	ORed Message buffer (48-63)	-	CAN2_ORed_48_63_MB
129	113	eTMR0	Channel 0/1 interrupt	-	eTMR0_Ch0_Ch1
130	114	eTMR0	channel 2/3 interrupt	-	eTMR0_Ch2_Ch3
131	115	eTMR0	channel 4/5 interrupt	-	eTMR0_Ch4_Ch5
132	116	eTMR0	channel 6/7 interrupt	-	eTMR0_Ch6_Ch7
133	117	eTMR0	Fault interrupt	-	eTMR0_Fault
134	118	eTMR0	Counter overflow interrupt	-	eTMR0_Ovf
135	119	eTMR1	Channel 0/1 interrupt	-	eTMR1_Ch0_Ch1
136	120	eTMR1	channel 2/3 interrupt	-	eTMR1_Ch2_Ch3
137	121	eTMR1	channel 4/5 interrupt	-	eTMR1_Ch4_Ch5
138	122	eTMR1	channel 6/7 interrupt	-	eTMR1_Ch6_Ch7
139	123	eTMR1	Fault interrupt	-	eTMR1_Fault
140	124	eTMR1	Counter overflow interrupt	-	eTMR1_Ovf
141	125	eTMR2	Channel 0/1 interrupt	-	eTMR2_Ch0_Ch1
142	126	eTMR2	channel 2/3 interrupt	-	eTMR2_Ch2_Ch3
143	127	eTMR2	channel 4/5 interrupt	-	eTMR2_Ch4_Ch5
144	128	eTMR2	channel 6/7 interrupt	-	eTMR2_Ch6_Ch7
145	129	eTMR2	Fault interrupt	-	eTMR2_Fault
146	130	eTMR2	Counter overflow interrupt	-	eTMR2_Ovf

Yuntu Microelectronics

Table 10.3 continued from previous page

Vector	IRQ	Module	Description	Async[1]	Interrupt
147	131	eTMR3	Channel 0/1 interrupt	-	eTMR3_Ch0_Ch1
148	132	eTMR3	channel 2/3 interrupt	-	eTMR3_Ch2_Ch3
149	133	eTMR3	channel 4/5 interrupt	-	eTMR3_Ch4_Ch5
150	134	eTMR3	channel 6/7 interrupt	-	eTMR3_Ch6_Ch7
151	135	eTMR3	Fault interrupt	-	eTMR3_Fault
152	136	eTMR3	Counter overflow interrupt	-	eTMR3_Ovf
153	137	eTMR4	Channel 0/1 interrupt	-	eTMR4_Ch0_Ch1
154	138	eTMR4	channel 2/3 interrupt	-	eTMR4_Ch2_Ch3
155	139	eTMR4	channel 4/5 interrupt	-	eTMR4_Ch4_Ch5
156	140	eTMR4	channel 6/7 interrupt	-	eTMR4_Ch6_Ch7
157	141	eTMR4	Fault interrupt	-	eTMR4_Fault
158	142	eTMR4	Counter overflow interrupt	-	eTMR4_Ovf
159	143	eTMR5	Channel 0/1 interrupt	-	eTMR5_Ch0_Ch1
160	144	eTMR5	channel 2/3 interrupt	-	eTMR5_Ch2_Ch3
161	145	eTMR5	channel 4/5 interrupt	-	eTMR5_Ch4_Ch5
162	146	eTMR5	channel 6/7 interrupt	-	eTMR5_Ch6_Ch7
163	147	eTMR5	Fault interrupt	-	eTMR5_Fault
164	148	eTMR5	Counter overflow interrupt	-	eTMR5_Ovf
165	149	pTMR2	Periodic timer 2 channel0 interrupt	-	pTMR2_Ch0
166	150	pTMR2	Periodic timer 2 channel1 interrupt	-	pTMR2_Ch1
167	151	pTMR2	Periodic timer 2 channel2 interrupt	-	pTMR2_Ch2
168	152	pTMR2	Periodic timer 2 channel3 interrupt	-	pTMR2_Ch3
169	153	-	-	-	-
170	154	-	-	-	-
171	155	-	-	-	-
172	156	-	-	-	-
173	157	SENT0	SENT0 interrupt	Y	SENT0
174	158	SENT1	SENT1 interrupt	Y	SENT1
175	159	FMU	Alarm interrupt	Y	FMU_Alarm
176	160	FMU	Interrupt request (CFG state timeout)	Y	FMU_MISC
177	161	TRNG	TRNG interrupt	-	TRNG
178	162	HCU	HCU interrupt	-	HCU
179	163	INTM	Interrupt monitor interrupt	-	INTM
180	164	TMR0	Timer channle interrupt	-	TMR0
181	165	LINFlexD3	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD3
182	166	LINFlexD4	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD4
183	167	LINFlexD5	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD5
184	168	I2C2	LPI2C Interrupt	Y	I2C2_Master_Slave
185	169	SPI3	SPI Interrupt	Y	SPI3
186	170	SPI4	SPI Interrupt	Y	SPI4
187	171	SPI5	SPI Interrupt	Y	SPI5
188	172	CAN3	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN3_ORed
189	173	CAN3	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN3_Error
190	174	CAN3	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN3_Wake_Up
191	175	CAN3	ORed Message buffer (0-15)	-	CAN3_ORed_0_15_MB
192	176	CAN3	ORed Message buffer (16-31)	-	CAN3_ORed_16_31_MB
193	177	CAN3	ORed Message buffer (32-47)	-	CAN3_ORed_32_47_MB

Yuntu Microelectronics

Table 10.3 continued from previous page

Vector	IRQ	Module	Description	Async[1]	Interrupt
194	178	CAN3	ORed Message buffer (48-63)	-	CAN3_ORed_48_63_MB
195	179	CAN4	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN4_ORed
196	180	CAN4	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN4_Error
197	181	CAN4	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN4_Wake_Up
198	182	CAN4	ORed Message buffer (0-15)	-	CAN4_ORed_0_15_MB
199	183	CAN4	ORed Message buffer (16-31)	-	CAN4_ORed_16_31_MB
200	184	CAN4	ORed Message buffer (32-47)	-	CAN4_ORed_32_47_MB
201	185	CAN4	ORed Message buffer (48-63)	-	CAN4_ORed_48_63_MB
202	186	CAN5	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN5_ORed
203	187	CAN5	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN5_Error
204	188	CAN5	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN5_Wake_Up
205	189	CAN5	ORed Message buffer (0-15)	-	CAN5_ORed_0_15_MB
206	190	CAN5	ORed Message buffer (16-31)	-	CAN5_ORed_16_31_MB
207	191	WKU	Pin wakeup interrupt	Y	WKU
208	192	LINFlexD6	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD6
209	193	LINFlexD7	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD7
210	194	LINFlexD8	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD8
211	195	LINFlexD9	Transmit Interrupt Receive Interrupt Error Interrupt	Y	LINFlexD9
212	196	SPI6	SPI Interrupt	Y	SPI6
213	197	SPI7	SPI Interrupt	Y	SPI7
214	198	I2C3	LPI2C Interrupt	Y	I2C3_Master_Slave
215	199	I2C4	LPI2C Interrupt	Y	I2C4_Master_Slave
216	200	-	-	-	-
217	201	CAN6	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN6_ORed
218	202	CAN6	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN6_Error

Yuntu Microelectronics

Table 10.3 continued from previous page

Vector	IRQ	Module	Description	Async[1]	Interrupt
219	203	CAN6	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN6_Wake_Up
220	204	CAN6	ORed Message buffer (0-15)	-	CAN6_ORed_0_15_MB
221	205	CAN6	ORed Message buffer (16-31)	-	CAN6_ORed_16_31_MB
222	206	TMR1	Timer channle interrupt	-	TMR1
223	207	-	-	-	-
224	208	CM7 CTI	Debug interrupt from DWT compare outputs or ETM event outputs	-	CORE_CTI
225	209	-	-	-	-
226	210	SAI0	SAI TX/RX interrupts	-	SAI0
227	211	SAI1	SAI TX/RX interrupts	-	SAI1
228	212	-	-	-	-
229	213	-	-	-	-
230	214	-	-	-	-
231	215	-	-	-	-
232	216	-	-	-	-
233	217	-	-	-	-
234	218	CAN7	OR'ed [Bus Off OR Bus Off Done OR Transmit Warning OR Receive Warning] Interrupt indicating Transmit Error Counter transition from <96 to >= 96 Interrupt indicating Receive Error Counter transition from <96 to >= 96	-	CAN7_ORed
235	219	CAN7	Interrupt indicating that errors were detected on the CAN bus Interrupt indicating that errors were detected on the CAN bus for FD messages in the Fast Bit Rate region Interrupt asserted due to correctable error detected in memory read access Interrupt asserted due to non-correctable error detected in memory read access by host(CPU) Interrupt asserted due to non-correctable error detected in memory read access by the FlexCAN	-	CAN7_Error
236	220	CAN7	Interrupt asserted when Pretended Networking operation is enabled, and a valid message matches the selected filter criteria during Low Power mode Interrupt asserted when Pretended Networking operation is enabled, and no-reception of a matching message for a defined quantity of time during low power mode	Y	CAN7_Wake_Up
237	221	CAN7	ORed Message buffer (0-15)	-	CAN7_ORed_0_15_MB
238	222	CAN7	ORed Message buffer (16-31)	-	CAN7_ORed_16_31_MB
239	223	MPWM0	MPWM channel0-7 interrupt	-	MPWM0_Ch0_Ch7
240	224	MPWM0	MPWM channel8-15 interrupt	-	MPWM0_Ch8_Ch15
241	225	MPWM1	MPWM channel0-7 interrupt	-	MPWM1_Ch0_Ch7
242	226	MPWM1	MPWM channel8-15 interrupt	-	MPWM1_Ch8_Ch15
243	227	MPWM2	MPWM channel0-7 interrupt	-	MPWM2_Ch0_Ch7
244	228	MPWM2	MPWM channel8-15 interrupt	-	MPWM2_Ch8_Ch15

1. Async means the specified IRQ is asynchronous interrupt. And asynchronous interrupt can wake up MCU from Deepsleep and Standby power mode.

10.3 Wakeup Interrupt Controller (WIC)

The Wakeup Interrupt Controller (WIC) is a peripheral that can detect an interrupt and wake the processor from Deepsleep and Standby mode. The WIC is enabled only when the DEEPSLEEP bit in the **SCR** is set to 1.

The WIC is not programmable, and does not have any registers or user interface. It operates entirely from hardware signals.

When the WIC is enabled and the processor enters deep sleep mode, the power management unit in the system can power down most of the Cortex-M7 processor. This might have the side effect of stopping the SysTick timer. When the WIC receives an interrupt, it takes several clock cycles to wake up the processor and restore its state, before it can process the interrupt. This means interrupt latency is increased in deep sleep mode.

Yuntu Microelectronics

— NOTE —

If the processor detects a connection to a debugger, it disables the WIC.

10.4 FPU Configuration

The Cortex-M7 Floating-Point Unit (FPU) implements the **FPv5 floating-point** extensions.

The FPU fully supports **single-precision** add, subtract, multiply, divide, multiply and accumulate, and square root operations. It also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions.

The FPU provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard.

The FPU contains 32 single-precision extension registers, which you can also access as 16 doubleword registers for load, store, and move operations.

10.5 JTAG Controller Configuration

The Cortex-M7 processor implements a complete hardware debug solution. This provides high system visibility of the processor and memory through either a traditional **JTAG** port or a 2-pin Serial Wire Debug (**SWD**) port that is ideal for microcontrollers and other small package devices.

When implemented, debuggers can use:

- The Breakpoint Unit (BPU), which supports eight hardware breakpoint comparators.
- The Data Watchpoint and Trace (DWT), which supports four watchpoint comparators.

10.6 Glossary

BPU	Breakpoint Unit
DWT	Data Watchpoint and Trace
FPU	Floating-Point Unit
JTAG	Joint Test Action Group
SWD	Serial Wire Debug
WIC	Wakeup Interrupt Controller

Yuntu Microelectronics

Chapter 11

Direct Memory Access (DMA)

11.1 Chip Specific Information

YTM32B1HA0x has 1 DMA instance with 32 channels, each channel support 128 hardware mux selection.

11.1.1 DMA Channel MUX table

DMA module has internal channel mux for each channel, with dma channel mux, each dma channel could support up to 128 hardware request triggers.

DMA hardware trigger enables DMA to start data transfer when hardware request is asserted. User can use DMA to send data to peripheral when peripheral is idle, or receive data from peripheral when new data is available.

Table 11.1: DMA Channel Mux Table

No.	DMA_Request Description	Module
0	Disabled	Disabled
1	-	-
2	Receive DMA request	LINFlexD0_RX
3	Transmit DMA request	LINFlexD0_TX
4	Receive DMA request	LINFlexD1_RX
5	Transmit DMA request	LINFlexD1_TX
6	Receive DMA request	LINFlexD2_RX
7	Transmit DMA request	LINFlexD2_TX
8	DMA RX request	I2C1_RX
9	DMA TX request	I2C1_TX
10	Receive DMA request	LINFlexD3_RX
11	Transmit DMA request	LINFlexD3_TX
12	Receive DMA request	LINFlexD4_RX
13	Transmit DMA request	LINFlexD4_TX
14	Receive DMA request	SPI0_RX
15	Transmit DMA request	SPI0_TX
16	Receive DMA request	SPI1_RX
17	Transmit DMA request	SPI1_TX
18	Receive DMA request	SPI2_RX

Yuntu Microelectronics

Table 11.1 continued from previous page

No.	DMA_Request Description	Module
19	Transmit DMA request	SPI2_TX
20	Channel 0 DMA transfer request	eTMR1_CH0
21	Channel 1 DMA transfer request	eTMR1_CH1
22	Channel 2 DMA transfer request	eTMR1_CH2
23	Channel 3 DMA transfer request	eTMR1_CH3
24	Channel 4 DMA transfer request	eTMR1_CH4
25	Channel 5 DMA transfer request	eTMR1_CH5
26	Channel 6 DMA transfer request	eTMR1_CH6
27	Channel 7 DMA transfer request	eTMR1_CH7
28	Channel 0 DMA transfer request	eTMR2_CH0
29	Channel 1 DMA transfer request	eTMR2_CH1
30	Channel 2 DMA transfer request	eTMR2_CH2
31	Channel 3 DMA transfer request	eTMR2_CH3
32	Channel 4 DMA transfer request	eTMR2_CH4
33	Channel 5 DMA transfer request	eTMR2_CH5
34	Channel 6 DMA transfer request	eTMR2_CH6
35	Channel 7 DMA transfer request	eTMR2_CH7
36	eTMR0 DMA transfer request	eTMR0
37	eTMR3 DMA transfer request	eTMR3
38	eTMR4 DMA transfer request	eTMR4
39	eTMR5 DMA transfer request	eTMR5
40	-	-
41	-	-
42	DMA request	ADC0
43	DMA request	ADC1
44	DMA RX request	I2C0_RX
45	DMA TX request	I2C0_TX
46	DMA request	PTU0
47	DMA request	PTU1
48	DMA request	ACMP0
49	DMA request	GPIOA
50	DMA request	GPIOB
51	DMA request	GPIOC
52	DMA request	GPIOD
53	DMA request	GPIOE
54	CAN DMA request	FlexCAN0
55	CAN DMA request	FlexCAN1
56	CAN DMA request	FlexCAN2
57	CAN DMA request	FlexCAN3
58	CAN DMA request	FlexCAN4
59	DMA request	lpTMR0
60	CAN DMA request	FlexCAN5
61	-	-
62	-	-
63	-	-

Yuntu Microelectronics

Table 11.1 continued from previous page

No.	DMA_Request Description	Module
64	Receive DMA request	SPI3_RX
65	Transmit DMA request	SPI3_TX
66	Receive DMA request	SPI4_RX
67	Transmit DMA request	SPI4_TX
68	Receive DMA request	SPI5_RX
69	Transmit DMA request	SPI5_TX
70	DMA RX request	I2C2_RX
71	DMA TX request	I2C2_TX
72	Receive DMA request	LINFlexD5_RX
73	Transmit DMA request	LINFlexD5_TX
74	HCU INGRESS	HCU_INGRESS
75	HCU EGRESS	HCU_EGRESS
76	Receive DMA request	LINFlexD6_RX
77	Transmit DMA request	LINFlexD6_TX
78	Receive DMA request	LINFlexD7_RX
79	Transmit DMA request	LINFlexD7_TX
80	Receive DMA request	LINFlexD8_RX
81	Transmit DMA request	LINFlexD8_TX
82	Receive DMA request	LINFlexD9_RX
83	Transmit DMA request	LINFlexD9_TX
84	DMA RX request	I2C3_RX
85	DMA TX request	I2C3_TX
86	Receive DMA request	SPI6_RX
87	Transmit DMA request	SPI6_TX
88	Receive DMA request	SPI7_RX
89	Transmit DMA request	SPI7_TX
90	DMA RX request	I2C4_RX
91	DMA TX request	I2C4_TX
92	-	-
93	-	-
94	DMA request	ACMP1
95	-	-
96	CAN DMA request	FlexCAN6
97	CAN DMA request	FlexCAN7
98	-	-
99	-	-
100	-	-
101	-	-
102	DMA fast RX request	SENT0_FAST
103	DMA slow RX request	SENT0_SLOW
104	DMA fast RX request	SENT1_FAST
105	DMA slow RX request	SENT1_SLOW
106	RX buffer drain DMA request	QSPI_RX
107	TX buffer fill DMA request	QSPI_TX
108	MPWM DMA request	MPWM0

Yuntu Microelectronics

Table 11.1 continued from previous page

No.	DMA_Request Description	Module
109	MPWM DMA request	MPWM1
110	MPWM DMA request	MPWM2
111	-	-
112	IEEE 1588 PTP timer ch[0]	EMAC_CH0
113	IEEE 1588 PTP timer ch[1]	EMAC_CH1
114	IEEE 1588 PTP timer ch[2]	EMAC_CH2
115	IEEE 1588 PTP timer ch[3]	EMAC_CH3
116	Channel 0 DMA receive request	SAI0_RX_CH0
117	Channel 1 DMA receive request	SAI0_RX_CH1
118	Channel 2 DMA receive request	SAI0_RX_CH2
119	Channel 3 DMA receive request	SAI0_RX_CH3
120	Channel 0 DMA transmit request	SAI0_TX_CH0
121	Channel 1 DMA transmit request	SAI0_TX_CH1
122	Channel 2 DMA transmit request	SAI0_TX_CH2
123	Channel 3 DMA transmit request	SAI0_TX_CH3
124	DMA receive request	SAI1_TX
125	DMA transmit request	SAI1_RX

11.1.2 Power Mode Description

Table 11.2: DMA Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Function disabled
Deep Sleep	Function disabled
Debug	Support pause DMA transfer when CPU halt under debug mode

11.2 Introduction

Direct Memory Access (DMA) module allows hardware access SoC memory without CPU instructions. With DMA modules, CPU only have to configuration DMA channel transfer structure(CTS), and DMA will start the transfer automatically after given trigger comes.

In order to support various peripherals in chip, the DMA has an internal channel mux module to support hardware trigger selection from peripherals.

11.2.1 Features

- All address range data transfer from source to destination
- Support separate source/destination data size configuration
  - Double word(64-bit), word(32-bit), half word(16-bit), byte(8-bit) transfer size
- Support separate source/destination address offset configuration
  - Address increase/decrease/keep selectable
- Up to 32 DMA channels
  - Fix priority and round-robin arbitration

Yuntu Microelectronics

  - Support channel to channel link
- Software/Hardware/Link trigger
- Up to 128 peripheral hardware triggers
- Internal data fifo for data transfer
- Support update DMA transfer information from system memory after transfer complete
- Support data transfer loop and trigger loop

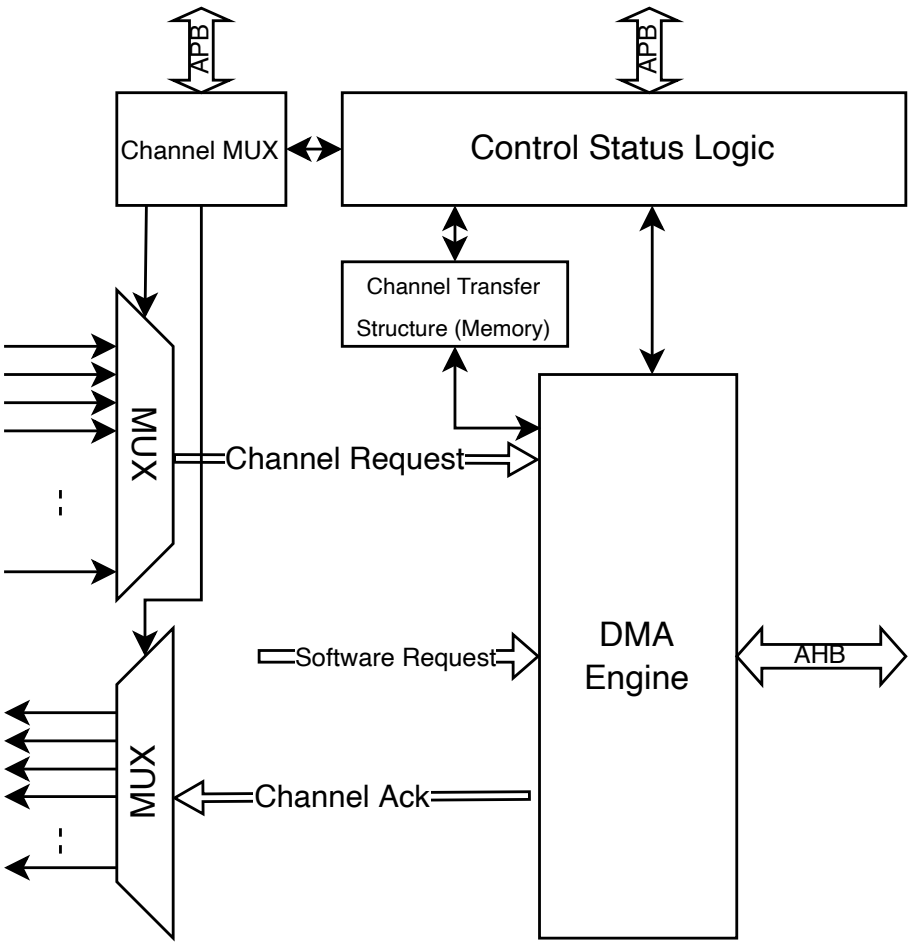


Figure 11.1: Block Diagram of DMA

11.2.2 DMA Channel Transfer Structure (CTS)

In order to support different data transfer configuration, DMA use Channel Transfer Structure(CTS) for each transfer channel.

Table 11.3: DMA Channel Transfer Structure Table

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Source address																															
SMOD					SSIZE			DMOD					DSIZE			SOFF															
Byte couter and loop done offset control																															
Source Trigger loop done offset																															
DADDR																															
Trigger counter and Trigger Loop Link information																DOFF															
Destination trigger loop done offset or ram reload address																															
Trigger counter and Trigger Loop Link information reload value																Channel control information															

NOTE. CTS information is stored in DMA internal memory, CTS reset value is random, so user have to initialize all CTS registers before using DMA.

11.2.3 DMA Trigger MUX

In order to support more hardware trigger source, DMA has an internal trigger MUX to mux hardware trigger source to DMA channel request.

NOTE. DMA MUX not support connect one hardware to different DMA channels simultaneously.

11.3 Register Definition

11.3.1 DMA Memory Map

DMA0 base address 40020000h

Table 11.4: DMA Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Channel Trigger Loop Done Interrupt Flag Register (CHTLDIF)	32	RW	00000000h
8h	Channel Trigger Loop Half Done Interrupt Flag Register (CHTLHDIF)	32	RW	00000000h
ch	DMA Channel Error Interrupt Flag Register (CHEIF)	32	RW	00000000h
10h	DMA Channel Error Interrupt Enable Register (CHEIE)	32	RW	00000000h
14h	Error Status Register (ERS)	32	RW	00000000h
18h	Hardware Request Enable Register (REQEN)	32	RW	00000000h
1ch	Request Status Register (REQS)	32	RW	00000000h
20h	Channel Start Register (START)	32	RW	00000000h
24h	Channel Done Register (DONE)	32	RW	00000000h
40h	Channel Mux Register (CHMUX0)	8	RW	00h
41h	Channel Mux Register (CHMUX1)	8	RW	00h
42h	Channel Mux Register (CHMUX2)	8	RW	00h
43h	Channel Mux Register (CHMUX3)	8	RW	00h
44h	Channel Mux Register (CHMUX4)	8	RW	00h
45h	Channel Mux Register (CHMUX5)	8	RW	00h
46h	Channel Mux Register (CHMUX6)	8	RW	00h
47h	Channel Mux Register (CHMUX7)	8	RW	00h
48h	Channel Mux Register (CHMUX8)	8	RW	00h
49h	Channel Mux Register (CHMUX9)	8	RW	00h
4ah	Channel Mux Register (CHMUX10)	8	RW	00h
4bh	Channel Mux Register (CHMUX11)	8	RW	00h
4ch	Channel Mux Register (CHMUX12)	8	RW	00h
4dh	Channel Mux Register (CHMUX13)	8	RW	00h
4eh	Channel Mux Register (CHMUX14)	8	RW	00h
4fh	Channel Mux Register (CHMUX15)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
50h	Channel Mux Register (CHMUX16)	8	RW	00h
51h	Channel Mux Register (CHMUX17)	8	RW	00h
52h	Channel Mux Register (CHMUX18)	8	RW	00h
53h	Channel Mux Register (CHMUX19)	8	RW	00h
54h	Channel Mux Register (CHMUX20)	8	RW	00h
55h	Channel Mux Register (CHMUX21)	8	RW	00h
56h	Channel Mux Register (CHMUX22)	8	RW	00h
57h	Channel Mux Register (CHMUX23)	8	RW	00h
58h	Channel Mux Register (CHMUX24)	8	RW	00h
59h	Channel Mux Register (CHMUX25)	8	RW	00h
5ah	Channel Mux Register (CHMUX26)	8	RW	00h
5bh	Channel Mux Register (CHMUX27)	8	RW	00h
5ch	Channel Mux Register (CHMUX28)	8	RW	00h
5dh	Channel Mux Register (CHMUX29)	8	RW	00h
5eh	Channel Mux Register (CHMUX30)	8	RW	00h
5fh	Channel Mux Register (CHMUX31)	8	RW	00h
400h	Source Address Register (CTS0_SADDR)	32	RW	see description
404h	Source Address Offset Register (CTS0_SOFF)	16	RW	see description
406h	Transfer Control Register (CTS0_TCR)	16	RW	see description
408h	Transfer Loop Byte Count (CTS0_BCNT_BCNT)	32	RW	see description
408h	Transfer Loop Offset Disable Mapping (CTS0_BCNT_LODIS)	32	RW	see description
408h	Transfer Loop Offset Enable Mapping (CTS0_BCNT_LOEN)	32	RW	see description
40ch	Source Trigger Loop Offset Register (CTS0_STO)	32	RW	see description
410h	Destination Address Register (CTS0_DADDR)	32	RW	see description
414h	Destination Address Offset Register (CTS0_DOFF)	16	RW	see description
416h	Trigger Count with Link Disable (CTS0_TCNT_LKDIS)	16	RW	see description
416h	Trigger Count with Link Enable (CTS0_TCNT_LKEN)	16	RW	see description
418h	Destination Trigger Loop Offset Register (CTS0_DTO_RLD_DTO)	32	RW	see description
418h	RAM Reload Address Register (CTS0_DTO_RLD_RLD)	32	RW	see description
41ch	Control and Status Register (CTS0_CSR)	16	RW	see description
41eh	TCNT Reload Value Register (CTS0_TCNTRV)	16	RW	see description
420h	Source Address Register (CTS1_SADDR)	32	RW	see description
424h	Source Address Offset Register (CTS1_SOFF)	16	RW	see description
426h	Transfer Control Register (CTS1_TCR)	16	RW	see description
428h	Transfer Loop Byte Count (CTS1_BCNT_BCNT)	32	RW	see description
428h	Transfer Loop Offset Disable Mapping (CTS1_BCNT_LODIS)	32	RW	see description
428h	Transfer Loop Offset Enable Mapping (CTS1_BCNT_LOEN)	32	RW	see description
42ch	Source Trigger Loop Offset Register (CTS1_STO)	32	RW	see description
430h	Destination Address Register (CTS1_DADDR)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
434h	Destination Address Offset Register (CTS1_DOFF)	16	RW	see description
436h	Trigger Count with Link Disable (CTS1_TCNT_LKDIS)	16	RW	see description
436h	Trigger Count with Link Enable (CTS1_TCNT_LKEN)	16	RW	see description
438h	Destination Trigger Loop Offset Register (CTS1_DTO_RLD_DTO)	32	RW	see description
438h	RAM Reload Address Register (CTS1_DTO_RLD_RLD)	32	RW	see description
43ch	Control and Status Register (CTS1_CSR)	16	RW	see description
43eh	TCNT Reload Value Register (CTS1_TCNTRV)	16	RW	see description
440h	Source Address Register (CTS2_SADDR)	32	RW	see description
444h	Source Address Offset Register (CTS2_SOFF)	16	RW	see description
446h	Transfer Control Register (CTS2_TCR)	16	RW	see description
448h	Transfer Loop Byte Count (CTS2_BCNT_BCNT)	32	RW	see description
448h	Transfer Loop Offset Disable Mapping (CTS2_BCNT_LODIS)	32	RW	see description
448h	Transfer Loop Offset Enable Mapping (CTS2_BCNT_LOEN)	32	RW	see description
44ch	Source Trigger Loop Offset Register (CTS2_STO)	32	RW	see description
450h	Destination Address Register (CTS2_DADDR)	32	RW	see description
454h	Destination Address Offset Register (CTS2_DOFF)	16	RW	see description
456h	Trigger Count with Link Disable (CTS2_TCNT_LKDIS)	16	RW	see description
456h	Trigger Count with Link Enable (CTS2_TCNT_LKEN)	16	RW	see description
458h	Destination Trigger Loop Offset Register (CTS2_DTO_RLD_DTO)	32	RW	see description
458h	RAM Reload Address Register (CTS2_DTO_RLD_RLD)	32	RW	see description
45ch	Control and Status Register (CTS2_CSR)	16	RW	see description
45eh	TCNT Reload Value Register (CTS2_TCNTRV)	16	RW	see description
460h	Source Address Register (CTS3_SADDR)	32	RW	see description
464h	Source Address Offset Register (CTS3_SOFF)	16	RW	see description
466h	Transfer Control Register (CTS3_TCR)	16	RW	see description
468h	Transfer Loop Byte Count (CTS3_BCNT_BCNT)	32	RW	see description
468h	Transfer Loop Offset Disable Mapping (CTS3_BCNT_LODIS)	32	RW	see description
468h	Transfer Loop Offset Enable Mapping (CTS3_BCNT_LOEN)	32	RW	see description
46ch	Source Trigger Loop Offset Register (CTS3_STO)	32	RW	see description
470h	Destination Address Register (CTS3_DADDR)	32	RW	see description
474h	Destination Address Offset Register (CTS3_DOFF)	16	RW	see description
476h	Trigger Count with Link Disable (CTS3_TCNT_LKDIS)	16	RW	see description
476h	Trigger Count with Link Enable (CTS3_TCNT_LKEN)	16	RW	see description
478h	Destination Trigger Loop Offset Register (CTS3_DTO_RLD_DTO)	32	RW	see description
478h	RAM Reload Address Register (CTS3_DTO_RLD_RLD)	32	RW	see description
47ch	Control and Status Register (CTS3_CSR)	16	RW	see description
47eh	TCNT Reload Value Register (CTS3_TCNTRV)	16	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
480h	Source Address Register (CTS4_SADDR)	32	RW	see description
484h	Source Address Offset Register (CTS4_SOFF)	16	RW	see description
486h	Transfer Control Register (CTS4_TCR)	16	RW	see description
488h	Transfer Loop Byte Count (CTS4_BCNT_BCNT)	32	RW	see description
488h	Transfer Loop Offset Disable Mapping (CTS4_BCNT_LODIS)	32	RW	see description
488h	Transfer Loop Offset Enable Mapping (CTS4_BCNT_LOEN)	32	RW	see description
48ch	Source Trigger Loop Offset Register (CTS4_STO)	32	RW	see description
490h	Destination Address Register (CTS4_DADDR)	32	RW	see description
494h	Destination Address Offset Register (CTS4_DOFF)	16	RW	see description
496h	Trigger Count with Link Disable (CTS4_TCNT_LKDIS)	16	RW	see description
496h	Trigger Count with Link Enable (CTS4_TCNT_LKEN)	16	RW	see description
498h	Destination Trigger Loop Offset Register (CTS4_DTO_RLD_DTO)	32	RW	see description
498h	RAM Reload Address Register (CTS4_DTO_RLD_RLD)	32	RW	see description
49ch	Control and Status Register (CTS4_CSR)	16	RW	see description
49eh	TCNT Reload Value Register (CTS4_TCNTRV)	16	RW	see description
4a0h	Source Address Register (CTS5_SADDR)	32	RW	see description
4a4h	Source Address Offset Register (CTS5_SOFF)	16	RW	see description
4a6h	Transfer Control Register (CTS5_TCR)	16	RW	see description
4a8h	Transfer Loop Byte Count (CTS5_BCNT_BCNT)	32	RW	see description
4a8h	Transfer Loop Offset Disable Mapping (CTS5_BCNT_LODIS)	32	RW	see description
4a8h	Transfer Loop Offset Enable Mapping (CTS5_BCNT_LOEN)	32	RW	see description
4ach	Source Trigger Loop Offset Register (CTS5_STO)	32	RW	see description
4b0h	Destination Address Register (CTS5_DADDR)	32	RW	see description
4b4h	Destination Address Offset Register (CTS5_DOFF)	16	RW	see description
4b6h	Trigger Count with Link Disable (CTS5_TCNT_LKDIS)	16	RW	see description
4b6h	Trigger Count with Link Enable (CTS5_TCNT_LKEN)	16	RW	see description
4b8h	Destination Trigger Loop Offset Register (CTS5_DTO_RLD_DTO)	32	RW	see description
4b8h	RAM Reload Address Register (CTS5_DTO_RLD_RLD)	32	RW	see description
4bch	Control and Status Register (CTS5_CSR)	16	RW	see description
4beh	TCNT Reload Value Register (CTS5_TCNTRV)	16	RW	see description
4c0h	Source Address Register (CTS6_SADDR)	32	RW	see description
4c4h	Source Address Offset Register (CTS6_SOFF)	16	RW	see description
4c6h	Transfer Control Register (CTS6_TCR)	16	RW	see description
4c8h	Transfer Loop Byte Count (CTS6_BCNT_BCNT)	32	RW	see description
4c8h	Transfer Loop Offset Disable Mapping (CTS6_BCNT_LODIS)	32	RW	see description
4c8h	Transfer Loop Offset Enable Mapping (CTS6_BCNT_LOEN)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
4cch	Source Trigger Loop Offset Register (CTS6_STO)	32	RW	see description
4d0h	Destination Address Register (CTS6_DADDR)	32	RW	see description
4d4h	Destination Address Offset Register (CTS6_DOFF)	16	RW	see description
4d6h	Trigger Count with Link Disable (CTS6_TCNT_LKDIS)	16	RW	see description
4d6h	Trigger Count with Link Enable (CTS6_TCNT_LKEN)	16	RW	see description
4d8h	Destination Trigger Loop Offset Register (CTS6_DTO_RLD_DTO)	32	RW	see description
4d8h	RAM Reload Address Register (CTS6_DTO_RLD_RLD)	32	RW	see description
4dch	Control and Status Register (CTS6_CSR)	16	RW	see description
4deh	TCNT Reload Value Register (CTS6_TCNTRV)	16	RW	see description
4e0h	Source Address Register (CTS7_SADDR)	32	RW	see description
4e4h	Source Address Offset Register (CTS7_SOFF)	16	RW	see description
4e6h	Transfer Control Register (CTS7_TCR)	16	RW	see description
4e8h	Transfer Loop Byte Count (CTS7_BCNT_BCNT)	32	RW	see description
4e8h	Transfer Loop Offset Disable Mapping (CTS7_BCNT_LODIS)	32	RW	see description
4e8h	Transfer Loop Offset Enable Mapping (CTS7_BCNT_LOEN)	32	RW	see description
4ech	Source Trigger Loop Offset Register (CTS7_STO)	32	RW	see description
4f0h	Destination Address Register (CTS7_DADDR)	32	RW	see description
4f4h	Destination Address Offset Register (CTS7_DOFF)	16	RW	see description
4f6h	Trigger Count with Link Disable (CTS7_TCNT_LKDIS)	16	RW	see description
4f6h	Trigger Count with Link Enable (CTS7_TCNT_LKEN)	16	RW	see description
4f8h	Destination Trigger Loop Offset Register (CTS7_DTO_RLD_DTO)	32	RW	see description
4f8h	RAM Reload Address Register (CTS7_DTO_RLD_RLD)	32	RW	see description
4fch	Control and Status Register (CTS7_CSR)	16	RW	see description
4feh	TCNT Reload Value Register (CTS7_TCNTRV)	16	RW	see description
500h	Source Address Register (CTS8_SADDR)	32	RW	see description
504h	Source Address Offset Register (CTS8_SOFF)	16	RW	see description
506h	Transfer Control Register (CTS8_TCR)	16	RW	see description
508h	Transfer Loop Byte Count (CTS8_BCNT_BCNT)	32	RW	see description
508h	Transfer Loop Offset Disable Mapping (CTS8_BCNT_LODIS)	32	RW	see description
508h	Transfer Loop Offset Enable Mapping (CTS8_BCNT_LOEN)	32	RW	see description
50ch	Source Trigger Loop Offset Register (CTS8_STO)	32	RW	see description
510h	Destination Address Register (CTS8_DADDR)	32	RW	see description
514h	Destination Address Offset Register (CTS8_DOFF)	16	RW	see description
516h	Trigger Count with Link Disable (CTS8_TCNT_LKDIS)	16	RW	see description
516h	Trigger Count with Link Enable (CTS8_TCNT_LKEN)	16	RW	see description
518h	Destination Trigger Loop Offset Register (CTS8_DTO_RLD_DTO)	32	RW	see description
518h	RAM Reload Address Register (CTS8_DTO_RLD_RLD)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
51ch	Control and Status Register (CTS8_CSR)	16	RW	see description
51eh	TCNT Reload Value Register (CTS8_TCNTRV)	16	RW	see description
520h	Source Address Register (CTS9_SADDR)	32	RW	see description
524h	Source Address Offset Register (CTS9_SOFF)	16	RW	see description
526h	Transfer Control Register (CTS9_TCR)	16	RW	see description
528h	Transfer Loop Byte Count (CTS9_BCNT_BCNT)	32	RW	see description
528h	Transfer Loop Offset Disable Mapping (CTS9_BCNT_LODIS)	32	RW	see description
528h	Transfer Loop Offset Enable Mapping (CTS9_BCNT_LOEN)	32	RW	see description
52ch	Source Trigger Loop Offset Register (CTS9_STO)	32	RW	see description
530h	Destination Address Register (CTS9_DADDR)	32	RW	see description
534h	Destination Address Offset Register (CTS9_DOFF)	16	RW	see description
536h	Trigger Count with Link Disable (CTS9_TCNT_LKDIS)	16	RW	see description
536h	Trigger Count with Link Enable (CTS9_TCNT_LKEN)	16	RW	see description
538h	Destination Trigger Loop Offset Register (CTS9_DTO_RLD_DTO)	32	RW	see description
538h	RAM Reload Address Register (CTS9_DTO_RLD_RLD)	32	RW	see description
53ch	Control and Status Register (CTS9_CSR)	16	RW	see description
53eh	TCNT Reload Value Register (CTS9_TCNTRV)	16	RW	see description
540h	Source Address Register (CTS10_SADDR)	32	RW	see description
544h	Source Address Offset Register (CTS10_SOFF)	16	RW	see description
546h	Transfer Control Register (CTS10_TCR)	16	RW	see description
548h	Transfer Loop Byte Count (CTS10_BCNT_BCNT)	32	RW	see description
548h	Transfer Loop Offset Disable Mapping (CTS10_BCNT_LODIS)	32	RW	see description
548h	Transfer Loop Offset Enable Mapping (CTS10_BCNT_LOEN)	32	RW	see description
54ch	Source Trigger Loop Offset Register (CTS10_STO)	32	RW	see description
550h	Destination Address Register (CTS10_DADDR)	32	RW	see description
554h	Destination Address Offset Register (CTS10_DOFF)	16	RW	see description
556h	Trigger Count with Link Disable (CTS10_TCNT_LKDIS)	16	RW	see description
556h	Trigger Count with Link Enable (CTS10_TCNT_LKEN)	16	RW	see description
558h	Destination Trigger Loop Offset Register (CTS10_DTO_RLD_DTO)	32	RW	see description
558h	RAM Reload Address Register (CTS10_DTO_RLD_RLD)	32	RW	see description
55ch	Control and Status Register (CTS10_CSR)	16	RW	see description
55eh	TCNT Reload Value Register (CTS10_TCNTRV)	16	RW	see description
560h	Source Address Register (CTS11_SADDR)	32	RW	see description
564h	Source Address Offset Register (CTS11_SOFF)	16	RW	see description
566h	Transfer Control Register (CTS11_TCR)	16	RW	see description
568h	Transfer Loop Byte Count (CTS11_BCNT_BCNT)	32	RW	see description
568h	Transfer Loop Offset Disable Mapping (CTS11_BCNT_LODIS)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
568h	Transfer Loop Offset Enable Mapping (CTS11_BCNT_LOEN)	32	RW	see description
56ch	Source Trigger Loop Offset Register (CTS11_STO)	32	RW	see description
570h	Destination Address Register (CTS11_DADDR)	32	RW	see description
574h	Destination Address Offset Register (CTS11_DOFF)	16	RW	see description
576h	Trigger Count with Link Disable (CTS11_TCNT_LKDIS)	16	RW	see description
576h	Trigger Count with Link Enable (CTS11_TCNT_LKEN)	16	RW	see description
578h	Destination Trigger Loop Offset Register (CTS11_DTO_RLD_DTO)	32	RW	see description
578h	RAM Reload Address Register (CTS11_DTO_RLD_RLD)	32	RW	see description
57ch	Control and Status Register (CTS11_CSR)	16	RW	see description
57eh	TCNT Reload Value Register (CTS11_TCNTRV)	16	RW	see description
580h	Source Address Register (CTS12_SADDR)	32	RW	see description
584h	Source Address Offset Register (CTS12_SOFF)	16	RW	see description
586h	Transfer Control Register (CTS12_TCR)	16	RW	see description
588h	Transfer Loop Byte Count (CTS12_BCNT_BCNT)	32	RW	see description
588h	Transfer Loop Offset Disable Mapping (CTS12_BCNT_LODIS)	32	RW	see description
588h	Transfer Loop Offset Enable Mapping (CTS12_BCNT_LOEN)	32	RW	see description
58ch	Source Trigger Loop Offset Register (CTS12_STO)	32	RW	see description
590h	Destination Address Register (CTS12_DADDR)	32	RW	see description
594h	Destination Address Offset Register (CTS12_DOFF)	16	RW	see description
596h	Trigger Count with Link Disable (CTS12_TCNT_LKDIS)	16	RW	see description
596h	Trigger Count with Link Enable (CTS12_TCNT_LKEN)	16	RW	see description
598h	Destination Trigger Loop Offset Register (CTS12_DTO_RLD_DTO)	32	RW	see description
598h	RAM Reload Address Register (CTS12_DTO_RLD_RLD)	32	RW	see description
59ch	Control and Status Register (CTS12_CSR)	16	RW	see description
59eh	TCNT Reload Value Register (CTS12_TCNTRV)	16	RW	see description
5a0h	Source Address Register (CTS13_SADDR)	32	RW	see description
5a4h	Source Address Offset Register (CTS13_SOFF)	16	RW	see description
5a6h	Transfer Control Register (CTS13_TCR)	16	RW	see description
5a8h	Transfer Loop Byte Count (CTS13_BCNT_BCNT)	32	RW	see description
5a8h	Transfer Loop Offset Disable Mapping (CTS13_BCNT_LODIS)	32	RW	see description
5a8h	Transfer Loop Offset Enable Mapping (CTS13_BCNT_LOEN)	32	RW	see description
5ach	Source Trigger Loop Offset Register (CTS13_STO)	32	RW	see description
5b0h	Destination Address Register (CTS13_DADDR)	32	RW	see description
5b4h	Destination Address Offset Register (CTS13_DOFF)	16	RW	see description
5b6h	Trigger Count with Link Disable (CTS13_TCNT_LKDIS)	16	RW	see description
5b6h	Trigger Count with Link Enable (CTS13_TCNT_LKEN)	16	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
5b8h	Destination Trigger Loop Offset Register (CTS13_DTO_RLD_DTO)	32	RW	see description
5b8h	RAM Reload Address Register (CTS13_DTO_RLD_RLD)	32	RW	see description
5bch	Control and Status Register (CTS13_CSR)	16	RW	see description
5beh	TCNT Reload Value Register (CTS13_TCNTRV)	16	RW	see description
5c0h	Source Address Register (CTS14_SADDR)	32	RW	see description
5c4h	Source Address Offset Register (CTS14_SOFF)	16	RW	see description
5c6h	Transfer Control Register (CTS14_TCR)	16	RW	see description
5c8h	Transfer Loop Byte Count (CTS14_BCNT_BCNT)	32	RW	see description
5c8h	Transfer Loop Offset Disable Mapping (CTS14_BCNT_LODIS)	32	RW	see description
5c8h	Transfer Loop Offset Enable Mapping (CTS14_BCNT_LOEN)	32	RW	see description
5cch	Source Trigger Loop Offset Register (CTS14_STO)	32	RW	see description
5d0h	Destination Address Register (CTS14_DADDR)	32	RW	see description
5d4h	Destination Address Offset Register (CTS14_DOFF)	16	RW	see description
5d6h	Trigger Count with Link Disable (CTS14_TCNT_LKDIS)	16	RW	see description
5d6h	Trigger Count with Link Enable (CTS14_TCNT_LKEN)	16	RW	see description
5d8h	Destination Trigger Loop Offset Register (CTS14_DTO_RLD_DTO)	32	RW	see description
5d8h	RAM Reload Address Register (CTS14_DTO_RLD_RLD)	32	RW	see description
5dch	Control and Status Register (CTS14_CSR)	16	RW	see description
5deh	TCNT Reload Value Register (CTS14_TCNTRV)	16	RW	see description
5e0h	Source Address Register (CTS15_SADDR)	32	RW	see description
5e4h	Source Address Offset Register (CTS15_SOFF)	16	RW	see description
5e6h	Transfer Control Register (CTS15_TCR)	16	RW	see description
5e8h	Transfer Loop Byte Count (CTS15_BCNT_BCNT)	32	RW	see description
5e8h	Transfer Loop Offset Disable Mapping (CTS15_BCNT_LODIS)	32	RW	see description
5e8h	Transfer Loop Offset Enable Mapping (CTS15_BCNT_LOEN)	32	RW	see description
5ech	Source Trigger Loop Offset Register (CTS15_STO)	32	RW	see description
5f0h	Destination Address Register (CTS15_DADDR)	32	RW	see description
5f4h	Destination Address Offset Register (CTS15_DOFF)	16	RW	see description
5f6h	Trigger Count with Link Disable (CTS15_TCNT_LKDIS)	16	RW	see description
5f6h	Trigger Count with Link Enable (CTS15_TCNT_LKEN)	16	RW	see description
5f8h	Destination Trigger Loop Offset Register (CTS15_DTO_RLD_DTO)	32	RW	see description
5f8h	RAM Reload Address Register (CTS15_DTO_RLD_RLD)	32	RW	see description
5fch	Control and Status Register (CTS15_CSR)	16	RW	see description
5feh	TCNT Reload Value Register (CTS15_TCNTRV)	16	RW	see description
600h	Source Address Register (CTS16_SADDR)	32	RW	see description
604h	Source Address Offset Register (CTS16_SOFF)	16	RW	see description
606h	Transfer Control Register (CTS16_TCR)	16	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
608h	Transfer Loop Byte Count (CTS16_BCNT_BCNT)	32	RW	see description
608h	Transfer Loop Offset Disable Mapping (CTS16_BCNT_LODIS)	32	RW	see description
608h	Transfer Loop Offset Enable Mapping (CTS16_BCNT_LOEN)	32	RW	see description
60ch	Source Trigger Loop Offset Register (CTS16_STO)	32	RW	see description
610h	Destination Address Register (CTS16_DADDR)	32	RW	see description
614h	Destination Address Offset Register (CTS16_DOFF)	16	RW	see description
616h	Trigger Count with Link Disable (CTS16_TCNT_LKDIS)	16	RW	see description
616h	Trigger Count with Link Enable (CTS16_TCNT_LKEN)	16	RW	see description
618h	Destination Trigger Loop Offset Register (CTS16_DTO_RLD_DTO)	32	RW	see description
618h	RAM Reload Address Register (CTS16_DTO_RLD_RLD)	32	RW	see description
61ch	Control and Status Register (CTS16_CSR)	16	RW	see description
61eh	TCNT Reload Value Register (CTS16_TCNTRV)	16	RW	see description
620h	Source Address Register (CTS17_SADDR)	32	RW	see description
624h	Source Address Offset Register (CTS17_SOFF)	16	RW	see description
626h	Transfer Control Register (CTS17_TCR)	16	RW	see description
628h	Transfer Loop Byte Count (CTS17_BCNT_BCNT)	32	RW	see description
628h	Transfer Loop Offset Disable Mapping (CTS17_BCNT_LODIS)	32	RW	see description
628h	Transfer Loop Offset Enable Mapping (CTS17_BCNT_LOEN)	32	RW	see description
62ch	Source Trigger Loop Offset Register (CTS17_STO)	32	RW	see description
630h	Destination Address Register (CTS17_DADDR)	32	RW	see description
634h	Destination Address Offset Register (CTS17_DOFF)	16	RW	see description
636h	Trigger Count with Link Disable (CTS17_TCNT_LKDIS)	16	RW	see description
636h	Trigger Count with Link Enable (CTS17_TCNT_LKEN)	16	RW	see description
638h	Destination Trigger Loop Offset Register (CTS17_DTO_RLD_DTO)	32	RW	see description
638h	RAM Reload Address Register (CTS17_DTO_RLD_RLD)	32	RW	see description
63ch	Control and Status Register (CTS17_CSR)	16	RW	see description
63eh	TCNT Reload Value Register (CTS17_TCNTRV)	16	RW	see description
640h	Source Address Register (CTS18_SADDR)	32	RW	see description
644h	Source Address Offset Register (CTS18_SOFF)	16	RW	see description
646h	Transfer Control Register (CTS18_TCR)	16	RW	see description
648h	Transfer Loop Byte Count (CTS18_BCNT_BCNT)	32	RW	see description
648h	Transfer Loop Offset Disable Mapping (CTS18_BCNT_LODIS)	32	RW	see description
648h	Transfer Loop Offset Enable Mapping (CTS18_BCNT_LOEN)	32	RW	see description
64ch	Source Trigger Loop Offset Register (CTS18_STO)	32	RW	see description
650h	Destination Address Register (CTS18_DADDR)	32	RW	see description
654h	Destination Address Offset Register (CTS18_DOFF)	16	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
656h	Trigger Count with Link Disable (CTS18_TCNT_LKDIS)	16	RW	see description
656h	Trigger Count with Link Enable (CTS18_TCNT_LKEN)	16	RW	see description
658h	Destination Trigger Loop Offset Register (CTS18_DTO_RLD_DTO)	32	RW	see description
658h	RAM Reload Address Register (CTS18_DTO_RLD_RLD)	32	RW	see description
65ch	Control and Status Register (CTS18_CSR)	16	RW	see description
65eh	TCNT Reload Value Register (CTS18_TCNTRV)	16	RW	see description
660h	Source Address Register (CTS19_SADDR)	32	RW	see description
664h	Source Address Offset Register (CTS19_SOFF)	16	RW	see description
666h	Transfer Control Register (CTS19_TCR)	16	RW	see description
668h	Transfer Loop Byte Count (CTS19_BCNT_BCNT)	32	RW	see description
668h	Transfer Loop Offset Disable Mapping (CTS19_BCNT_LODIS)	32	RW	see description
668h	Transfer Loop Offset Enable Mapping (CTS19_BCNT_LOEN)	32	RW	see description
66ch	Source Trigger Loop Offset Register (CTS19_STO)	32	RW	see description
670h	Destination Address Register (CTS19_DADDR)	32	RW	see description
674h	Destination Address Offset Register (CTS19_DOFF)	16	RW	see description
676h	Trigger Count with Link Disable (CTS19_TCNT_LKDIS)	16	RW	see description
676h	Trigger Count with Link Enable (CTS19_TCNT_LKEN)	16	RW	see description
678h	Destination Trigger Loop Offset Register (CTS19_DTO_RLD_DTO)	32	RW	see description
678h	RAM Reload Address Register (CTS19_DTO_RLD_RLD)	32	RW	see description
67ch	Control and Status Register (CTS19_CSR)	16	RW	see description
67eh	TCNT Reload Value Register (CTS19_TCNTRV)	16	RW	see description
680h	Source Address Register (CTS20_SADDR)	32	RW	see description
684h	Source Address Offset Register (CTS20_SOFF)	16	RW	see description
686h	Transfer Control Register (CTS20_TCR)	16	RW	see description
688h	Transfer Loop Byte Count (CTS20_BCNT_BCNT)	32	RW	see description
688h	Transfer Loop Offset Disable Mapping (CTS20_BCNT_LODIS)	32	RW	see description
688h	Transfer Loop Offset Enable Mapping (CTS20_BCNT_LOEN)	32	RW	see description
68ch	Source Trigger Loop Offset Register (CTS20_STO)	32	RW	see description
690h	Destination Address Register (CTS20_DADDR)	32	RW	see description
694h	Destination Address Offset Register (CTS20_DOFF)	16	RW	see description
696h	Trigger Count with Link Disable (CTS20_TCNT_LKDIS)	16	RW	see description
696h	Trigger Count with Link Enable (CTS20_TCNT_LKEN)	16	RW	see description
698h	Destination Trigger Loop Offset Register (CTS20_DTO_RLD_DTO)	32	RW	see description
698h	RAM Reload Address Register (CTS20_DTO_RLD_RLD)	32	RW	see description
69ch	Control and Status Register (CTS20_CSR)	16	RW	see description
69eh	TCNT Reload Value Register (CTS20_TCNTRV)	16	RW	see description
6a0h	Source Address Register (CTS21_SADDR)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
6a4h	Source Address Offset Register (CTS21_SOFF)	16	RW	see description
6a6h	Transfer Control Register (CTS21_TCR)	16	RW	see description
6a8h	Transfer Loop Byte Count (CTS21_BCNT_BCNT)	32	RW	see description
6a8h	Transfer Loop Offset Disable Mapping (CTS21_BCNT_LODIS)	32	RW	see description
6a8h	Transfer Loop Offset Enable Mapping (CTS21_BCNT_LOEN)	32	RW	see description
6ach	Source Trigger Loop Offset Register (CTS21_STO)	32	RW	see description
6b0h	Destination Address Register (CTS21_DADDR)	32	RW	see description
6b4h	Destination Address Offset Register (CTS21_DOFF)	16	RW	see description
6b6h	Trigger Count with Link Disable (CTS21_TCNT_LKDIS)	16	RW	see description
6b6h	Trigger Count with Link Enable (CTS21_TCNT_LKEN)	16	RW	see description
6b8h	Destination Trigger Loop Offset Register (CTS21_DTO_RLD_DTO)	32	RW	see description
6b8h	RAM Reload Address Register (CTS21_DTO_RLD_RLD)	32	RW	see description
6bch	Control and Status Register (CTS21_CSR)	16	RW	see description
6beh	TCNT Reload Value Register (CTS21_TCNTRV)	16	RW	see description
6c0h	Source Address Register (CTS22_SADDR)	32	RW	see description
6c4h	Source Address Offset Register (CTS22_SOFF)	16	RW	see description
6c6h	Transfer Control Register (CTS22_TCR)	16	RW	see description
6c8h	Transfer Loop Byte Count (CTS22_BCNT_BCNT)	32	RW	see description
6c8h	Transfer Loop Offset Disable Mapping (CTS22_BCNT_LODIS)	32	RW	see description
6c8h	Transfer Loop Offset Enable Mapping (CTS22_BCNT_LOEN)	32	RW	see description
6cch	Source Trigger Loop Offset Register (CTS22_STO)	32	RW	see description
6d0h	Destination Address Register (CTS22_DADDR)	32	RW	see description
6d4h	Destination Address Offset Register (CTS22_DOFF)	16	RW	see description
6d6h	Trigger Count with Link Disable (CTS22_TCNT_LKDIS)	16	RW	see description
6d6h	Trigger Count with Link Enable (CTS22_TCNT_LKEN)	16	RW	see description
6d8h	Destination Trigger Loop Offset Register (CTS22_DTO_RLD_DTO)	32	RW	see description
6d8h	RAM Reload Address Register (CTS22_DTO_RLD_RLD)	32	RW	see description
6dch	Control and Status Register (CTS22_CSR)	16	RW	see description
6deh	TCNT Reload Value Register (CTS22_TCNTRV)	16	RW	see description
6e0h	Source Address Register (CTS23_SADDR)	32	RW	see description
6e4h	Source Address Offset Register (CTS23_SOFF)	16	RW	see description
6e6h	Transfer Control Register (CTS23_TCR)	16	RW	see description
6e8h	Transfer Loop Byte Count (CTS23_BCNT_BCNT)	32	RW	see description
6e8h	Transfer Loop Offset Disable Mapping (CTS23_BCNT_LODIS)	32	RW	see description
6e8h	Transfer Loop Offset Enable Mapping (CTS23_BCNT_LOEN)	32	RW	see description
6ech	Source Trigger Loop Offset Register (CTS23_STO)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
6f0h	Destination Address Register (CTS23_DADDR)	32	RW	see description
6f4h	Destination Address Offset Register (CTS23_DOFF)	16	RW	see description
6f6h	Trigger Count with Link Disable (CTS23_TCNT_LKDIS)	16	RW	see description
6f6h	Trigger Count with Link Enable (CTS23_TCNT_LKEN)	16	RW	see description
6f8h	Destination Trigger Loop Offset Register (CTS23_DTO_RLD_DTO)	32	RW	see description
6f8h	RAM Reload Address Register (CTS23_DTO_RLD_RLD)	32	RW	see description
6fch	Control and Status Register (CTS23_CSR)	16	RW	see description
6feh	TCNT Reload Value Register (CTS23_TCNTRV)	16	RW	see description
700h	Source Address Register (CTS24_SADDR)	32	RW	see description
704h	Source Address Offset Register (CTS24_SOFF)	16	RW	see description
706h	Transfer Control Register (CTS24_TCR)	16	RW	see description
708h	Transfer Loop Byte Count (CTS24_BCNT_BCNT)	32	RW	see description
708h	Transfer Loop Offset Disable Mapping (CTS24_BCNT_LODIS)	32	RW	see description
708h	Transfer Loop Offset Enable Mapping (CTS24_BCNT_LOEN)	32	RW	see description
70ch	Source Trigger Loop Offset Register (CTS24_STO)	32	RW	see description
710h	Destination Address Register (CTS24_DADDR)	32	RW	see description
714h	Destination Address Offset Register (CTS24_DOFF)	16	RW	see description
716h	Trigger Count with Link Disable (CTS24_TCNT_LKDIS)	16	RW	see description
716h	Trigger Count with Link Enable (CTS24_TCNT_LKEN)	16	RW	see description
718h	Destination Trigger Loop Offset Register (CTS24_DTO_RLD_DTO)	32	RW	see description
718h	RAM Reload Address Register (CTS24_DTO_RLD_RLD)	32	RW	see description
71ch	Control and Status Register (CTS24_CSR)	16	RW	see description
71eh	TCNT Reload Value Register (CTS24_TCNTRV)	16	RW	see description
720h	Source Address Register (CTS25_SADDR)	32	RW	see description
724h	Source Address Offset Register (CTS25_SOFF)	16	RW	see description
726h	Transfer Control Register (CTS25_TCR)	16	RW	see description
728h	Transfer Loop Byte Count (CTS25_BCNT_BCNT)	32	RW	see description
728h	Transfer Loop Offset Disable Mapping (CTS25_BCNT_LODIS)	32	RW	see description
728h	Transfer Loop Offset Enable Mapping (CTS25_BCNT_LOEN)	32	RW	see description
72ch	Source Trigger Loop Offset Register (CTS25_STO)	32	RW	see description
730h	Destination Address Register (CTS25_DADDR)	32	RW	see description
734h	Destination Address Offset Register (CTS25_DOFF)	16	RW	see description
736h	Trigger Count with Link Disable (CTS25_TCNT_LKDIS)	16	RW	see description
736h	Trigger Count with Link Enable (CTS25_TCNT_LKEN)	16	RW	see description
738h	Destination Trigger Loop Offset Register (CTS25_DTO_RLD_DTO)	32	RW	see description
738h	RAM Reload Address Register (CTS25_DTO_RLD_RLD)	32	RW	see description
73ch	Control and Status Register (CTS25_CSR)	16	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
73eh	TCNT Reload Value Register (CTS25_TCNTRV)	16	RW	see description
740h	Source Address Register (CTS26_SADDR)	32	RW	see description
744h	Source Address Offset Register (CTS26_SOFF)	16	RW	see description
746h	Transfer Control Register (CTS26_TCR)	16	RW	see description
748h	Transfer Loop Byte Count (CTS26_BCNT_BCNT)	32	RW	see description
748h	Transfer Loop Offset Disable Mapping (CTS26_BCNT_LODIS)	32	RW	see description
748h	Transfer Loop Offset Enable Mapping (CTS26_BCNT_LOEN)	32	RW	see description
74ch	Source Trigger Loop Offset Register (CTS26_STO)	32	RW	see description
750h	Destination Address Register (CTS26_DADDR)	32	RW	see description
754h	Destination Address Offset Register (CTS26_DOFF)	16	RW	see description
756h	Trigger Count with Link Disable (CTS26_TCNT_LKDIS)	16	RW	see description
756h	Trigger Count with Link Enable (CTS26_TCNT_LKEN)	16	RW	see description
758h	Destination Trigger Loop Offset Register (CTS26_DTO_RLD_DTO)	32	RW	see description
758h	RAM Reload Address Register (CTS26_DTO_RLD_RLD)	32	RW	see description
75ch	Control and Status Register (CTS26_CSR)	16	RW	see description
75eh	TCNT Reload Value Register (CTS26_TCNTRV)	16	RW	see description
760h	Source Address Register (CTS27_SADDR)	32	RW	see description
764h	Source Address Offset Register (CTS27_SOFF)	16	RW	see description
766h	Transfer Control Register (CTS27_TCR)	16	RW	see description
768h	Transfer Loop Byte Count (CTS27_BCNT_BCNT)	32	RW	see description
768h	Transfer Loop Offset Disable Mapping (CTS27_BCNT_LODIS)	32	RW	see description
768h	Transfer Loop Offset Enable Mapping (CTS27_BCNT_LOEN)	32	RW	see description
76ch	Source Trigger Loop Offset Register (CTS27_STO)	32	RW	see description
770h	Destination Address Register (CTS27_DADDR)	32	RW	see description
774h	Destination Address Offset Register (CTS27_DOFF)	16	RW	see description
776h	Trigger Count with Link Disable (CTS27_TCNT_LKDIS)	16	RW	see description
776h	Trigger Count with Link Enable (CTS27_TCNT_LKEN)	16	RW	see description
778h	Destination Trigger Loop Offset Register (CTS27_DTO_RLD_DTO)	32	RW	see description
778h	RAM Reload Address Register (CTS27_DTO_RLD_RLD)	32	RW	see description
77ch	Control and Status Register (CTS27_CSR)	16	RW	see description
77eh	TCNT Reload Value Register (CTS27_TCNTRV)	16	RW	see description
780h	Source Address Register (CTS28_SADDR)	32	RW	see description
784h	Source Address Offset Register (CTS28_SOFF)	16	RW	see description
786h	Transfer Control Register (CTS28_TCR)	16	RW	see description
788h	Transfer Loop Byte Count (CTS28_BCNT_BCNT)	32	RW	see description
788h	Transfer Loop Offset Disable Mapping (CTS28_BCNT_LODIS)	32	RW	see description

Yuntu Microelectronics

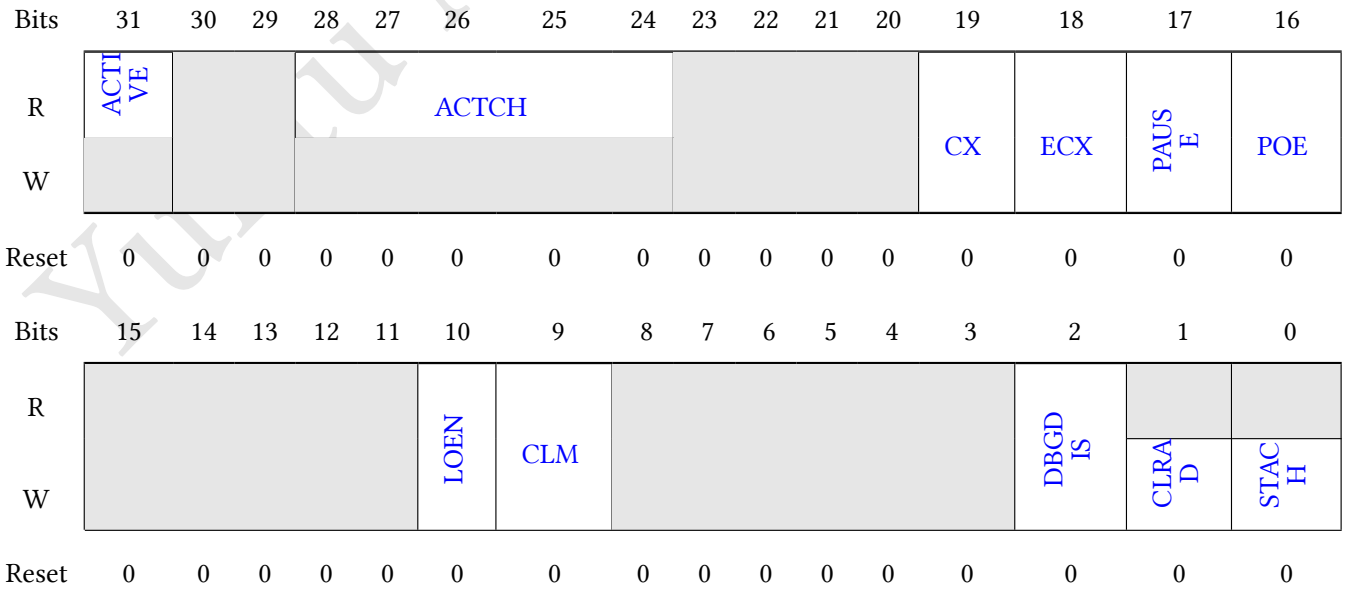
Offset	Register	Width	Access	Reset Value
788h	Transfer Loop Offset Enable Mapping (CTS28_BCNT_LOEN)	32	RW	see description
78ch	Source Trigger Loop Offset Register (CTS28_STO)	32	RW	see description
790h	Destination Address Register (CTS28_DADDR)	32	RW	see description
794h	Destination Address Offset Register (CTS28_DOFF)	16	RW	see description
796h	Trigger Count with Link Disable (CTS28_TCNT_LKDIS)	16	RW	see description
796h	Trigger Count with Link Enable (CTS28_TCNT_LKEN)	16	RW	see description
798h	Destination Trigger Loop Offset Register (CTS28_DTO_RLD_DTO)	32	RW	see description
798h	RAM Reload Address Register (CTS28_DTO_RLD_RLD)	32	RW	see description
79ch	Control and Status Register (CTS28_CSR)	16	RW	see description
79eh	TCNT Reload Value Register (CTS28_TCNTRV)	16	RW	see description
7a0h	Source Address Register (CTS29_SADDR)	32	RW	see description
7a4h	Source Address Offset Register (CTS29_SOFF)	16	RW	see description
7a6h	Transfer Control Register (CTS29_TCR)	16	RW	see description
7a8h	Transfer Loop Byte Count (CTS29_BCNT_BCNT)	32	RW	see description
7a8h	Transfer Loop Offset Disable Mapping (CTS29_BCNT_LODIS)	32	RW	see description
7a8h	Transfer Loop Offset Enable Mapping (CTS29_BCNT_LOEN)	32	RW	see description
7ach	Source Trigger Loop Offset Register (CTS29_STO)	32	RW	see description
7b0h	Destination Address Register (CTS29_DADDR)	32	RW	see description
7b4h	Destination Address Offset Register (CTS29_DOFF)	16	RW	see description
7b6h	Trigger Count with Link Disable (CTS29_TCNT_LKDIS)	16	RW	see description
7b6h	Trigger Count with Link Enable (CTS29_TCNT_LKEN)	16	RW	see description
7b8h	Destination Trigger Loop Offset Register (CTS29_DTO_RLD_DTO)	32	RW	see description
7b8h	RAM Reload Address Register (CTS29_DTO_RLD_RLD)	32	RW	see description
7bch	Control and Status Register (CTS29_CSR)	16	RW	see description
7beh	TCNT Reload Value Register (CTS29_TCNTRV)	16	RW	see description
7c0h	Source Address Register (CTS30_SADDR)	32	RW	see description
7c4h	Source Address Offset Register (CTS30_SOFF)	16	RW	see description
7c6h	Transfer Control Register (CTS30_TCR)	16	RW	see description
7c8h	Transfer Loop Byte Count (CTS30_BCNT_BCNT)	32	RW	see description
7c8h	Transfer Loop Offset Disable Mapping (CTS30_BCNT_LODIS)	32	RW	see description
7c8h	Transfer Loop Offset Enable Mapping (CTS30_BCNT_LOEN)	32	RW	see description
7cch	Source Trigger Loop Offset Register (CTS30_STO)	32	RW	see description
7d0h	Destination Address Register (CTS30_DADDR)	32	RW	see description
7d4h	Destination Address Offset Register (CTS30_DOFF)	16	RW	see description
7d6h	Trigger Count with Link Disable (CTS30_TCNT_LKDIS)	16	RW	see description
7d6h	Trigger Count with Link Enable (CTS30_TCNT_LKEN)	16	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
7d8h	Destination Trigger Loop Offset Register (CTS30_DTO_RLD_DTO)	32	RW	see description
7d8h	RAM Reload Address Register (CTS30_DTO_RLD_RLD)	32	RW	see description
7dch	Control and Status Register (CTS30_CSR)	16	RW	see description
7deh	TCNT Reload Value Register (CTS30_TCNTRV)	16	RW	see description
7e0h	Source Address Register (CTS31_SADDR)	32	RW	see description
7e4h	Source Address Offset Register (CTS31_SOFF)	16	RW	see description
7e6h	Transfer Control Register (CTS31_TCR)	16	RW	see description
7e8h	Transfer Loop Byte Count (CTS31_BCNT_BCNT)	32	RW	see description
7e8h	Transfer Loop Offset Disable Mapping (CTS31_BCNT_LODIS)	32	RW	see description
7e8h	Transfer Loop Offset Enable Mapping (CTS31_BCNT_LOEN)	32	RW	see description
7ech	Source Trigger Loop Offset Register (CTS31_STO)	32	RW	see description
7f0h	Destination Address Register (CTS31_DADDR)	32	RW	see description
7f4h	Destination Address Offset Register (CTS31_DOFF)	16	RW	see description
7f6h	Trigger Count with Link Disable (CTS31_TCNT_LKDIS)	16	RW	see description
7f6h	Trigger Count with Link Enable (CTS31_TCNT_LKEN)	16	RW	see description
7f8h	Destination Trigger Loop Offset Register (CTS31_DTO_RLD_DTO)	32	RW	see description
7f8h	RAM Reload Address Register (CTS31_DTO_RLD_RLD)	32	RW	see description
7fch	Control and Status Register (CTS31_CSR)	16	RW	see description
7feh	TCNT Reload Value Register (CTS31_TCNTRV)	16	RW	see description

11.3.1.1 DMA CTRL Register

Offset: 0h



Yuntu Microelectronics

Table 11.5: DMA CTRL Register Description

Field	Function
31 ACTIVE	**DMA Active** Shows current DMA active state 0b - DMA is idle 1b - DMA is active
28 - 24 ACTCH	**DMA Active Channel** Shows current DMA active channel. This field only valid when DMA active, when DMA is idle ACTCH = 0
19 CX	**Cancel DMA Transfer** Setting this bit will terminate DMA data transfer after DMA finished current write operation. This bit is cleared by hardware after DMA data transfer terminated. 0b - DMA in normal operation 1b - Cancel DMA data transfer
18 ECX	**Error Cancel Transfer** Setting this bit will terminate DMA data transfer after DMA finished current write operation and set error flag. This bit is cleared by hardware after DMA data transfer terminated. 0b - DMA in normal operation 1b - Cancel DMA data transfer and set error flag
17 PAUSE	**Pause DMA Transfer** Pause DMA following channel transfer, active channel transfer is allowed to complete, following channel transfer can be resumed after this bit cleared. 0b - DMA in normal operation 1b - Pause DMA transfer
16 POE	**Pause DMA Transfer on Error** Pause DMA following channel transfer when any DMA error happened 0b - DMA in normal operation 1b - DMA transfer is paused as DMA error happens, software need to handle the error and clear this bit to continue following data transfer
10 LOEN	**Loop Offset Enable** Enable loop offset adjustment, this bit will affect all DMA channels transfer loop setting. 0b - DMA loop offset is disabled 1b - DMA loop is enabled and CTS[BCNT] split into SLOEN, DLOEN, OFFSET and BCNT
9	**Continuous Link Mode**

Yuntu Microelectronics

Field	Function
CLM	**NOTE**. Only set this bit when more than one DMA channel enabled 0b - DMA will serve other channel even if DMA channel linking to itself. 1b - DMA will restart current channel transfer if DMA channel linking to itself.
2 DBGDIS	**Debug Mode Disable** Setting this bit will pause DMA transfer when CPU is halted 0b - DMA in normal operation 1b - DMA transfer is paused when CPU is halted
1 CLRAD	**Clear All DMA Done** Setting this bit will clear all DMA channel done flags 0b - No effects 1b - Clear all DMA channel done flags
0 STACH	**Start All DMA Channel** Setting this bit will start all DMA channel 0b - No effects 1b - Start all DMA channels

11.3.1.2 DMA CHTLDIF Register

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.6: DMA CHTLDIF Register Description

Field	Function
31 - 0 CH	**DMA Channel Trigger Loop Done Interrupt Flag** CH[n] is channel n trigger loop done interrupt flag, write 1 to clear channel flag.

11.3.1.3 DMA CHTLHDIF Register

Offset: 8h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.7: DMA CHTLHDIF Register Description

Field	Function
31 - 0 CH	**DMA Channel Trigger Loop Half Done Interrupt Flag, write 1 to clear channel flag.** CH[n] is channel n trigger loop half done interrupt flag

11.3.1.4 DMA CHEIF Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.8: DMA CHEIF Register Description

Field	Function
31 - 0	**DMA Channel Error Interrupt Flag**

Yuntu Microelectronics

Field	Function
CH	CH[n] is channel n error interrupt flag

11.3.1.5 DMA CHEIE Register

Offset: 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.9: DMA CHEIE Register Description

Field	Function
31 - 0 CH	**DMA Channel Error Interrupt Enable** CH[n] is channel n error interrupt enable control bit

11.3.1.6 DMA ERS Register

Offset: 14h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VALID													ECX		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SAE	SOE	DAE	DOE	BTCE	RLDE	SBE	DBE				ECH				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 11.10: DMA ERS Register Description

Field	Function
31 VALID	**DMA Configuration Valid** All DMA channel configurations are valid. 0b - No DMA error happens 1b - DMA error happens on at least one DMA channel
18 ECX	**Error Cancel Transfer** 0b - No DMA error happens 1b - DMA error happens because and data transfer error happened.
15 SAE	**Source Address Error** DMA channel source address configuration error 0b - No DMA source address error happens 1b - At least one DMA channel has source address configuration error. Check CTSn_SADDR and CTSn_TCR configuration.
14 SOE	**Source Offset Error** DMA source offset configuration error 0b - No DMA source offset error happens 1b - At least one DMA channel has source offset configuration error. Check CTSn_SOFF and CTSn_TCR configuration.
13 DAE	**Destination Address Error** DMA channel destination address configuration error 0b - No DMA destination address error happens 1b - At least one DMA channel has destination address configuration error. Check CTSn_DADDR and CTSn_TCR configuration.
12 DOE	**Destination Offset Error** DMA channel destination offset configuration error 0b - No DMA destination offset error happens 1b - At least one DMA channel has destination offset configuration error. Check CTSn_DOFF and CTSn_TCR configuration.
11 BTCE	**Byte/Trigger Count Configuration Error** Byte count or trigger count configuration not match DMA transfer size 0b - No DMA byte/trigger count error happens 1b - At least one DMA channel has byte/trigger count configuration error. Check CTSn_BCNT, CTSn_TCNT, CTSn_TCR configuration.
10	**Configuration RAM Reload Error**

Yuntu Microelectronics

Field	Function
RLDE	Target reload address is not aligned with 32 bytes 0b - No DMA ram reload error happens 1b - At least one DMA channel has ram reload configuration error. Check CTSn_RLDADDR(should be 32 byte align)
9 SBE	**Source Bus Error** Bus error happened when DMA reading data from source address. 0b - No DMA source bus error happens 1b - At least one DMA channel has source bus data access error. System bus error happens when DMA reading data from source address.
8 DBE	**Destination Bus Error** Bus error happened when DMA writing data to destination address. 0b - No DMA destination bus error happens 1b - At least one DMA channel has destination bus data access error. System bus error happened when dma write data to destination address.
4 - 0 ECH	**Error Channel Index** Channel index of error captured **NOTE**. This field is valid only when DMA error happened.

11.3.1.7 DMA REQEN Register

Offset: 18h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.11: DMA REQEN Register Description

Field	Function
31 - 0 CH	**DMA Channel Request Enable** Hardware DMA request for channel [n] enable control bit, one bit for one channel.

Yuntu Microelectronics

11.3.1.8 DMA REQS Register

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.12: DMA REQS Register Description

Field	Function
31 - 0 CH	**DMA Channel Request Status** Hardware DMA request status for channel [n], one bit for one channel.

11.3.1.9 DMA START Register

Offset: 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.13: DMA START Register Description

Field	Function
31 - 0 CH	**DMA Channel Start** Start DMA channel [n], one bit for one channel.

11.3.1.10 DMA DONE Register

Offset: 24h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 11.14: DMA DONE Register Description

Field	Function
31 - 0 CH	**DMA Channel Done** Trigger loop done for channel [n], one bit for one channel.

11.3.1.11 DMA CHMUX Register

Offset: 40h + n * 1h

Bits	7	6	5	4	3	2	1	0
R		CH						
W								
Reset	0	0	0	0	0	0	0	0

Table 11.15: DMA CHMUX Register Description

Field	Function
6 - 0 CH	**Hardware Channel** DMA hardware request input index, please refer to DMA MUX table for MUX detail

11.3.1.12 DMA CTS_SADDR Register

Offset: 400h + n * 20h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.16: DMA CTS_SADDR Register Description

Field	Function
31 - 0 SADDR	**Source Address** Source address of DMA transfer.

11.3.1.13 DMA CTS_SOFF Register

Offset: 404h + n * 20h

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.17: DMA CTS_SOFF Register Description

Field	Function
15 - 0 SOFF	**Source Address Offset** Source address will add offset value(support signed-short type) after each DMA transfer size completed.

11.3.1.14 DMA CTS_TCR Register

Offset: 406h + n * 20h

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SMOD					SSIZE			DMOD					DSIZE		
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Yuntu Microelectronics

Table 11.18: DMA CTS_TCR Register Description

Field	Function
15 - 11 SMOD	**Source Address Modulo Value** Source address will add offset value(support signed-short type) and modulo SMOD after each DMA transfer size completed.
10 - 8 SSIZE	**Source Data Transfer Size** Source data transfer size is DMA access source data width. 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit
7 - 3 DMOD	**Destination Address Modulo** Destination address will add offset value(support signed-short type) and modulo DMOD after each DMA transfer size completed.
2 - 0 DSIZE	**Destination Data Transfer Size** Destination data transfer size is DMA write destination data width. 000b - 8-bit 001b - 16-bit 010b - 32-bit 011b - 64-bit

11.3.1.15 DMA CTS_BCNT_BCNT Register

Transfer loop byte count is the number of bytes to be transferred in each trigger loop.

Offset: 408h + n * 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BCNT															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BCNT															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.19: DMA CTS_BCNT_BCNT Register Description

Field	Function
31 - 0	**Transfer Loop Byte Count**

Yuntu Microelectronics

Field	Function
BCNT	Valid for CTRL[LOEN] = 0, Transfer loop byte count is the number of bytes to be transferred in each trigger loop.

11.3.1.16 DMA CTS_BCNT_LODIS Register

Offset: 408h + n * 20h

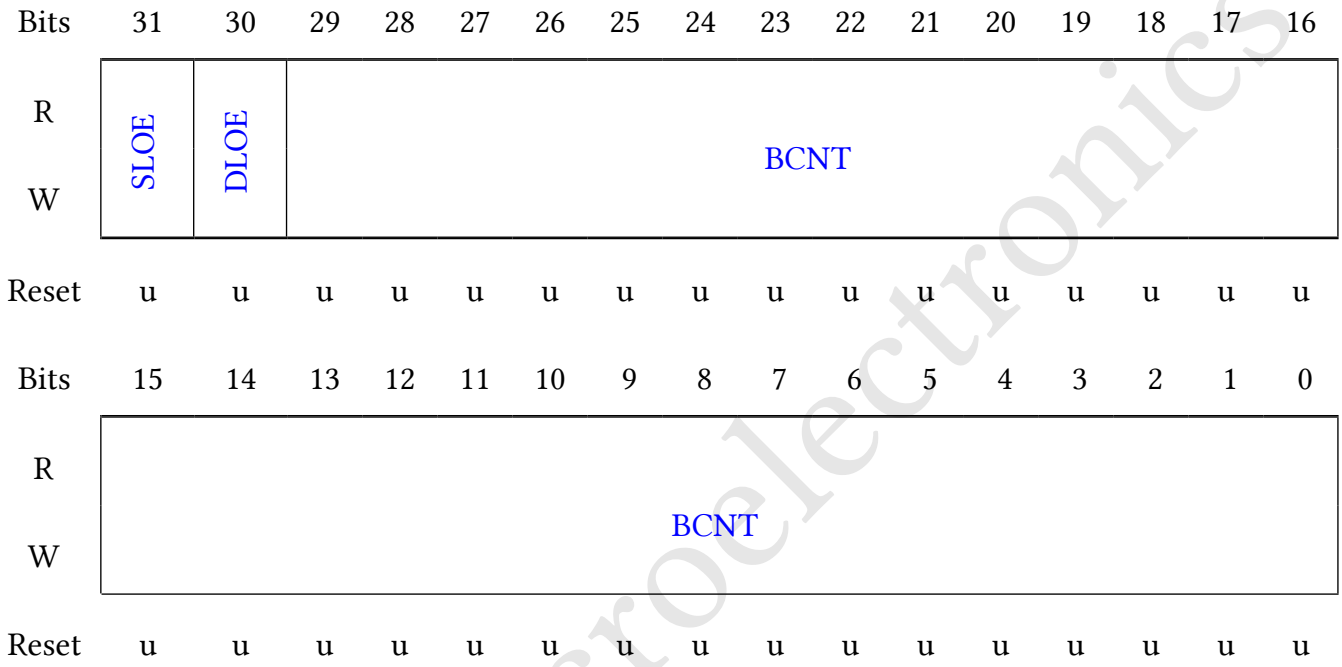


Table 11.20: DMA CTS_BCNT_LODIS Register Description

Field	Function
31 SLOE	**Source Loop Offset Enable** 0b - Source loop offset disabled 1b - Source loop offset enable, refer LOEN for register information
30 DLOE	**Destination Minor Loop Offset Enable** 0b - Destination minor loop offset disabled 1b - Destination minor loop offset enable, refer LOEN for register information
29 - 0 BCNT	**Byte Transfer Count**

11.3.1.17 DMA CTS_BCNT_LOEN Register

Offset: 408h + n * 20h

Yuntu Microelectronics

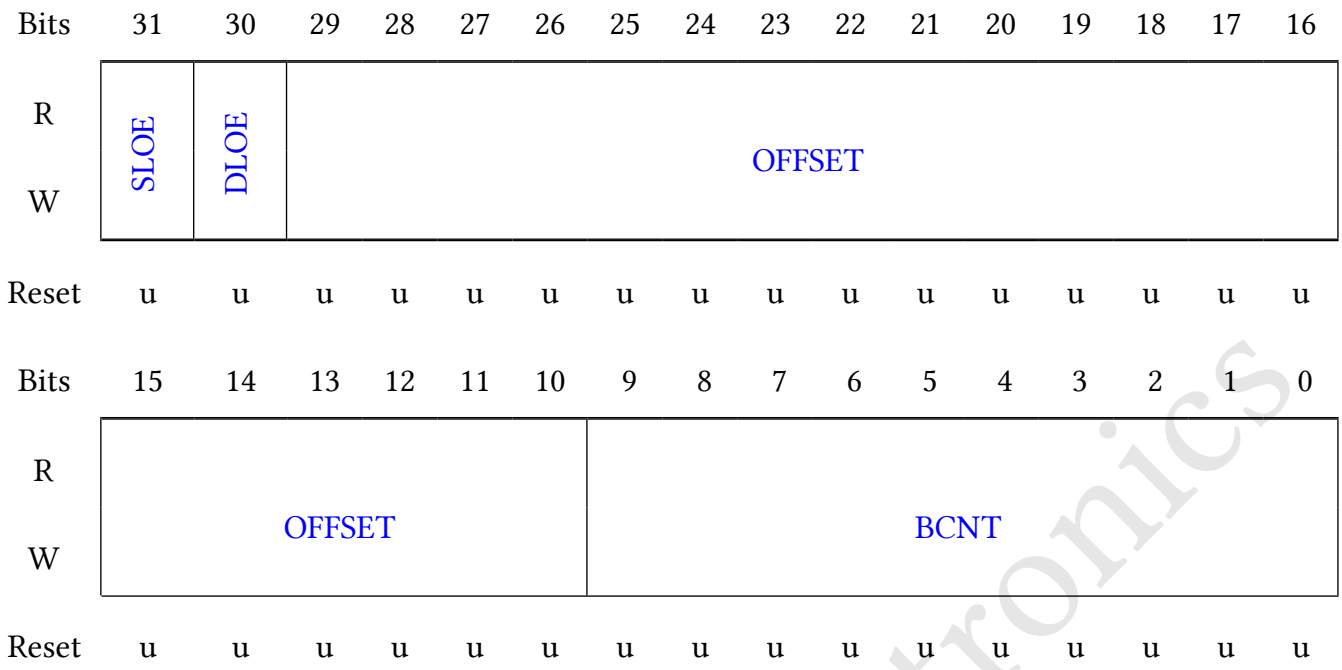


Table 11.21: DMA CTS_BCNT_LOEN Register Description

Field	Function
31 SLOE	**Source Loop Offset Enable** 0b - Source loop offset disabled, refer LODIS for register information 1b - Source loop offset enable
30 DLOE	**Destination Loop Offset Enable** 0b - Destination loop offset disabled, refer LODIS for register information 1b - Destination loop offset enable
29 - 10 OFFSET	**Loop Offset** Source or/and destination address will add offset value(support signed-short type) after each DMA loop completed.
9 - 0 BCNT	**Byte Transfer Count** Total byte count to be transferred in each trigger loop.

Yuntu Microelectronics

11.3.1.18 DMA CTS_STO Register

Offset: 40ch + n * 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	STLOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	STLOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.22: DMA CTS_STO Register Description

Field	Function
31 - 0 STLOFF	Source address will add offset value(support signed-short type) after DMA channel trigger loop completed.

11.3.1.19 DMA CTS_DADDR Register

Offset: 410h + n * 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DADDR															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.23: DMA CTS_DADDR Register Description

Field	Function
31 - 0 DADDR	**Destination Address** Destination address where DMA will transfer data to.

11.3.1.20 DMA CTS_DOFF Register

Offset: 414h + n * 20h

Yuntu Microelectronics

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DOFF															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.24: DMA CTS_DOFF Register Description

Field	Function
15 - 0 DOFF	**Destination Address Offset** Destination address will add offset value(support signed-short type) after each DMA transfer (write operation with DSIZE) completed.

11.3.1.21 DMA CTS_TCNT_LKDIS Register

Offset: 416h + n * 20h

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LKEN	TCNT														
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.25: DMA CTS_TCNT_LKDIS Register Description

Field	Function
15 LKEN	**Link Enable** Enable channel to channel linking after DMA one trigger loop complete. This bit should be 0 when channel to channel link mode is disabled. 0b - Channel to channel linking is disabled. 1b - Channel to channel linking is enabled, refer LKEN for register information
14 - 0 TCNT	**Trigger Count** Trigger counter decrease 1 for every DMA trigger

11.3.1.22 DMA CTS_TCNT_LKEN Register

Offset: 416h + n * 20h

Yuntu Microelectronics

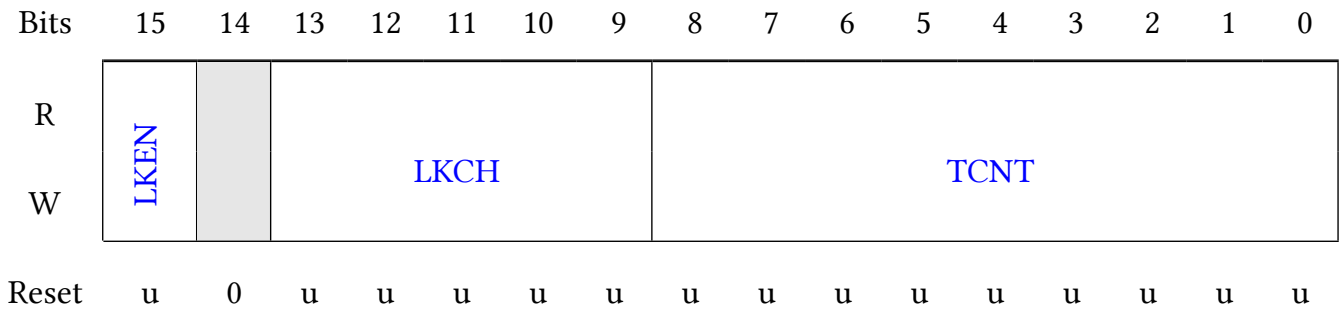


Table 11.26: DMA CTS_TCNT_LKEN Register Description

Field	Function
15 LKEN	**Link Enable** 0b - Channel to channel linking is disabled, refer LKDIS for register information 1b - Channel to channel linking is enabled.
13 - 9 LKCH	**Link Channel** DMA next trigger channel index after current dma trigger loop is done(one trigger)
8 - 0 TCNT	**Trigger Count** Trigger counter decrease 1 for every DMA trigger, when TCNT is 0, channel done flag set. **NOTE**. If new trigger happens and hardware request is enabled, TCNT will reload TCNT and serve new trigger request.

11.3.1.23 DMA CTS_DTO_RLD_DTO Register

Offset: 418h + n * 20h

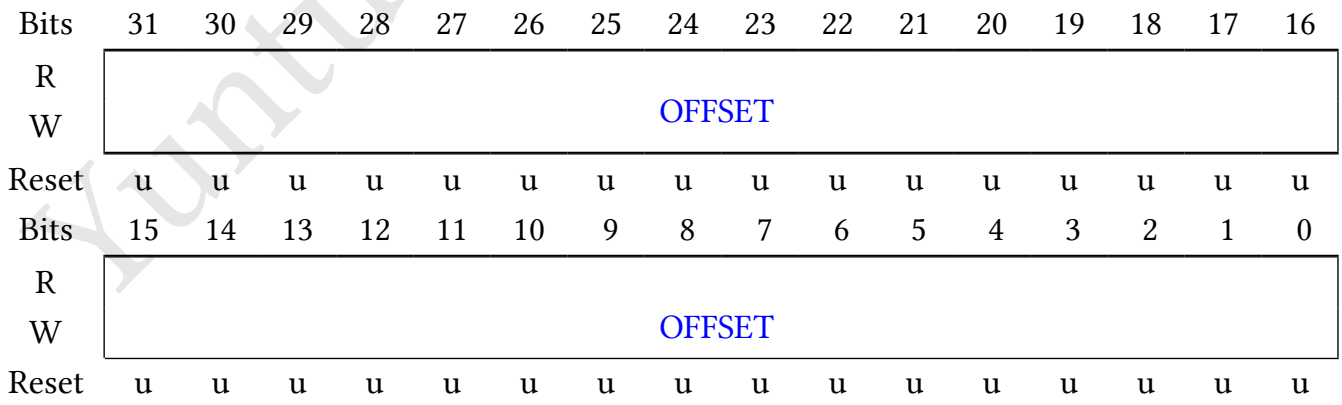


Table 11.27: DMA CTS_DTO_RLD_DTO Register Description

Field	Function
31 - 0	**Destination Trigger Loop Offset**

Yuntu Microelectronics

Field	Function
OFFSET	Destination address will add offset value(support signed-short type) after DMA channel trigger loop completed (trigger count reaches zero).

11.3.1.24 DMA CTS_DTO_RLD_RLD Register

Offset: 418h + n * 20h

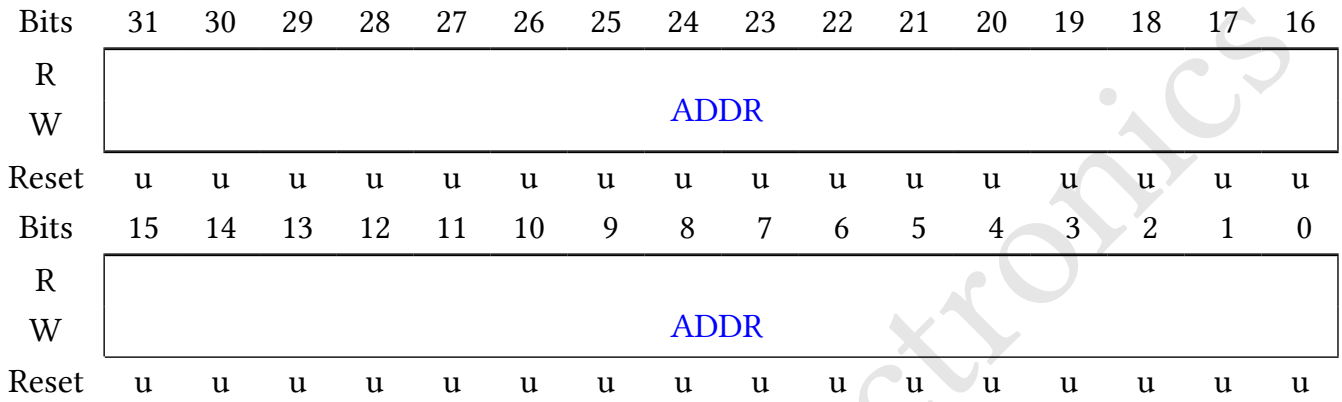


Table 11.28: DMA CTS_DTO_RLD_RLD Register Description

Field	Function
31 - 0 ADDR	**RAM Reload Address** DMA will load new channel configuration structure(CTS) from system memory after trigger loop completed (trigger count reaches zero). **NOTE**. RAM reload address should be 32 bytes aligned. **NOTE**. CTSn_CR[RLEDN] should be 1 when RAM reload function is used.

11.3.1.25 DMA CTS_CSR Register

Offset: 41ch + n * 20h

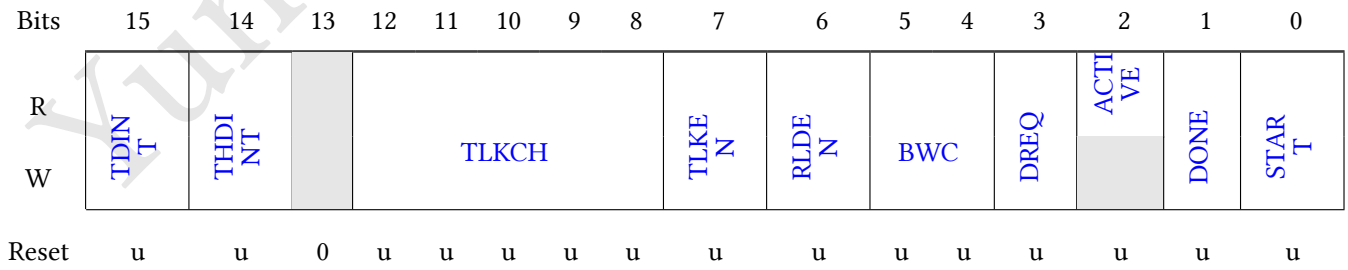


Table 11.29: DMA CTS_CSR Register Description

Field	Function
15	**Trigger Loop Done Interrupt Enable**

Yuntu Microelectronics

Field	Function
TDINT	Enable an interrupt when channel all trigger loop done 0b - Channel trigger loop done interrupt not enabled 1b - Channel trigger loop done interrupt enabled
14 THDINT	**Trigger Half Done Interrupt Enable** Enable an interrupt when half trigger loop done 0b - Channel trigger loop half done interrupt not enabled 1b - Channel trigger loop half done interrupt enabled
12 - 8 TLKCH	**Trigger Loop Link Channel Index** Setting this bit will let current DMA channel trigger linked DMA channel after current DMA channel all trigger loop completed
7 TLKEN	**Trigger Loop Link Enable** Enable channel to channel linking on all trigger loop complete 0b - Channel trigger loop done link not enabled 1b - Channel trigger loop done link enabled
6 RLDEN	**RAM Reload Enable** Setting this bit will current DMA channel load new configurations from system memory map after trigger loop completed **NOTE**. RAM reload feature is not available in YTM32B1HA0. 0b - Channel trigger loop done ram reload not enabled 1b - Channel trigger loop done ram reload enabled
5 - 4 BWC	**Bandwidth Control** Bandwidth control let DMA insert wait cycles between each R/W operation, this allows user to balance bus loadings. **NOTE**. If DMA need to transfer large amount of data, it is recommended to set BWC to 2. otherwise, CPU may need to wait DMA transfer for bus access. 00b - DMA transfer data continuous 01b - Reserved 10b - DMA engine wait 4 cycles after each R/W 11b - DMA engine wait 8 cycles after each R/W
3 DREQ	**Disable Request** Setting this bit will let DMA clear channel hardware request enable bit after trigger loop completed (TCNT decrease to zero). 0b - Channel trigger loop done won't clear channel hardware request enable bit, DMA will continue serve hardware request until even if TCNT already decrease down to zero. 1b - Channel trigger loop done will clear channel hardware request enable bit
2 ACTIVE	**Channel Active** 0b - DMA channel is not active 1b - DMA channel is active
1	**Trigger Loop Done**

Yuntu Microelectronics

Field	Function
DONE	0b - DMA channel trigger loop not done 1b - DMA channel trigger loop done
0 START	**Channel Start** 0b - DMA channel is not started 1b - DMA channel is started and waiting for trigger or active

11.3.1.26 DMA CTS_TCNTRV Register

Offset: 41eh + n * 20h

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RV															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 11.30: DMA CTS_TCNTRV Register Description

Field	Function
15 - 0 RV	**Reload Value** DMA will reload channel's TCNT with TCNTRV after DMA TCNT decreased to zero. This register share same register map with TCNT register.

11.4 Functional Description

Direct Memory Access (DMA) is a memory-mapped peripheral that provides transfer data from memory to memory, or from peripheral to memory, or from memory to peripheral. CPU only have to configure the source and destination address, and the length of data to be transferred. DMA module will automatically start the data transfer with software or hardware trigger.

DMA module support multiple channel triggers, user can configure the bytes count with each trigger and the total trigger count.

11.4.1 Transfer Loop

Transfer loop means data transferred after one DMA channel trigger. Transfer loop use BCNT as data transfer byte counter, BCNT is a counting down counter, when BCNT reaches zero, one transfer loop completed. So BCNT is total transfer data size in bytes of one DMA channel trigger. For example, if DMA need to transfer SPI received data to RAM memory, the transfer loop is should be configured as SPI fifo rx watermark. As SPI with trigger DMA when received data more than SPI fifo rx watermark, DMA need to transfer all received data to RAM memory. If transfer loop is less than rx watermark, system will still work, but DMA will generate more trigger loop.

Yuntu Microelectronics

11.4.2 Trigger Loop

Trigger loop means DMA channel could accept how many DMA channel triggers(include software and hardware trigger). Trigger count is define in TCNT field, TCNT is also a counting down counter, when TCNT reaches zero, one trigger loop completed. For example, if DMA need to transfer SPI received data to RAM memory, the trigger loop is should be configured as SPI total receive bytes divided by rx watermark.

11.4.3 Channel Arbitration

DMA module use round-robin arbitration when multi-channel trigger comes simultaneously, Round-robin arbitration is a fair algorithm, it lets every DMA channel have a chance to be served.

11.4.4 DMA Channel Initialize

1. Flush DMA CTS memory with channel default, for example clear CTS memory with 0.
2. Set DMA trigger source if using hardware trigger source.
3. Enable DMA channel error interrupt if needed.
4. Update DMA channel configuration structure with DMA channel configuration
5. Enable DMA hardware request

11.5 Glossary

BCNT	Byte Count
CTS	Channel Transfer Structure
TCNT	Trigger Count

Yuntu Microelectronics

Chapter 12

Chip Integration Module (CIM)

12.1 Introduction

The Chip Integration Module (CIM) provides system control and chip configuration registers.

12.1.1 Features

- System clocking configuration
- ADC and ACMP trigger selection
- Software trigger generate
- Software fault generate
- eTMR external clock, fault and channel input selection
- eTMR output modulation configuration
- FPU interrupt enable
- TCM backdoor access enable
- System boot configuration
- System unique device identification (UID)
- Flash memory and system RAM size configuration
- Package configuration
- FlexCAN FD feature configuration

12.2 Register Definition

The CIM includes registers to control and configure the chip.

12.2.1 CIM Memory Map

CIM base address 401ec000h

Table 12.1: CIM Memory Map

Offset	Register	Width	Access	Reset Value
0h	CIM Control Register 0 (CTRL0)	32	RW	78000000h
4h	CIM Control Register 1 (CTRL1)	32	RW	00000009h
10h	FPU Interrupt Enable Register (FPUIE)	32	RW	00000000h
14h	Software Trigger Register (SWTRIG)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
18h	eTMR Option register 0 (ETMROPT0)	32	RW	00000000h
1ch	eTMR Option register 1 (ETMROPT1)	32	RW	00000000h
20h	eTMR Option register 2 (ETMROPT2)	32	RW	00000000h
24h	eTMR Option register 3 (ETMROPT3)	32	RW	00000000h
28h	eTMR Option register 4 (ETMROPT4)	32	RW	00000000h
2ch	eTMR Option register 5 (ETMROPT5)	32	RW	00000000h
3ch	eTMR Global Option Register (ETMR_GLB_OPT)	32	RW	00000000h
40h	Software Fault Generate register (SWFAULT)	32	RW	00000000h
44h	Dual Core Compare Error Injection (DCCM_EI)	32	RW	00000000h
80h	CIM Unique Identification Register 0 (UID0)	32	RO	see description
84h	CIM Unique Identification Register 1 (UID1)	32	RO	see description
88h	CIM Unique Identification Register 2 (UID2)	32	RO	see description
8ch	CIM Unique Identification Register 3 (UID3)	32	RO	see description
90h	Die Information Register (DIEINFO)	32	RO	01200000h
94h	Chip Information Register 0 (CHIPINFO0)	32	RO	00CA8B02h
98h	Chip Information Register 1 (CHIPINFO1)	32	RO	see description
a0h	CPU Information Register (CPUINFO)	32	RO	see description
c0h	Chip Boot Configuration Register (BOOT_CFG)	32	RO	see description

12.2.1.1 CIM CTRL0 Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W													ADC1_NORMT_SEL		ADC0_NORMT_SEL	
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W					ACMP1_TRIG_SEL		ACMP0_TRIG_SEL									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 12.2: CIM CTRL0 Register Description

Field	Function
19 - 18	**ADC1 Normal Trigger Select**

Field	Function
ADC1_NORMT_SEL	Select trigger source for ADC1 normal mode. 00b - PTU1 output 01b - TMU output synced by bus clock 10b - TMU output synced by ADC1 func clock 11b - reserved
17 - 16 ADC0_NORMT_SEL	**ADC0 Normal Trigger Select** Select trigger source for ADC0 Normal mode. 00b - PTU0 output 01b - TMU output synced by bus clock 10b - TMU output synced by ADC0 func clock 11b - reserved
11 - 10 ACMP1_TRIG_SEL	**ACMP1 Trigger Select** ACMP1 Trigger Select 00b - Select TMU 01b - Select TMU synced by bus clock 10b - Select TMU synced by ACMP1 func clock 11b - reserved
9 - 8 ACMP0_TRIG_SEL	**ACMP0 Trigger Select** ACMP0 Trigger Select 00b - Select TMU output 01b - Select TMU output synced by bus clock 10b - Select TMU synced by ACMP0 func clock 11b - reserved

12.2.1.2 CIM CTRL1 Register

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VTORLK	AHBPCRLK	DTCMCRLK	ITCMCRLK									TCMBKDEN	LOCKUPEN	NMIEN	WICENREQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Yuntu Microelectronics

Table 12.3: CIM CTRL1 Register Description

Field	Function
15 VTORLK	**VTOR Write Lock** Lock CM7 VTOR Write 0b - VTOR Write not Locked 1b - VTOR Write Locked
14 AHBPCRLK	**AHBPCR Write Lock** Lock AHBPCR Write 0b - AHBPCR Write not Locked 1b - AHBPCR Write Locked
13 DTCMCRLK	**DTCMCR Write Lock** Lock DTCMCR Write 0b - DTCMCR Write not Locked 1b - DTCMCR Write Locked
12 ITCMCRLK	**ITCMCR Write Lock** Lock ITCMCR Write 0b - ITCMCR Write not Locked 1b - ITCMCR Write Locked
3 TCMBKDEN	**TCM Backdoor Access enable** Enable TCM Backdoor Access through CM7 AHBS bus 0b - TCM Backdoor Access is disabled 1b - TCM Backdoor Access is enabled
2 LOCKUPEN	**CPU Lockup Reset Enable** Enable CPU Lockup Reset 0b - CPU Lockup reset is disabled 1b - CPU Lockup reset is enabled
1 NMIEN	**Non-mask Interrupt Enable** Enable NMI function 0b - NMI is disabled 1b - NMI is enabled
0 WICENREQ	**WIC enable** Enable WIC to wakeup low power mode. 0b - Disable WIC to wakeup low power mode. 1b - Enable WIC to wakeup low power mode.

12.2.1.3 CIM FPUIE Register

Offset: 10h

Yuntu Microelectronics

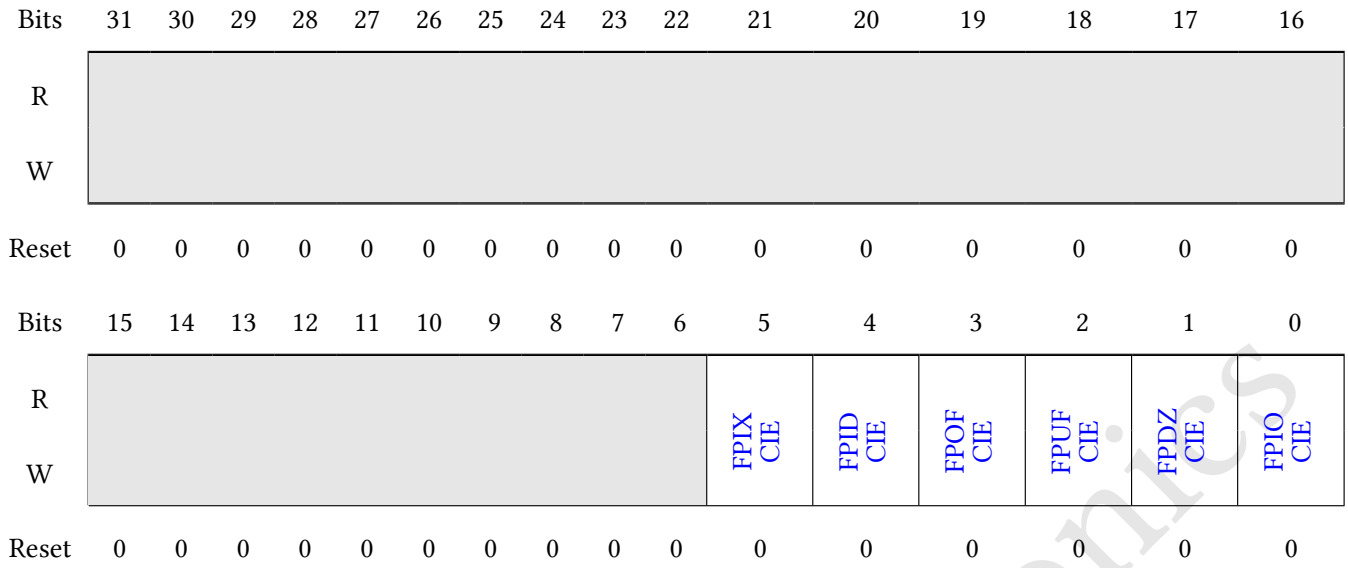


Table 12.4: CIM FPUIE Register Description

Field	Function
5 FPIXCIE	**FPU Inexact Interrupt Enable** 0b - Disable interrupt 1b - Enable interrupt
4 FPIDCIE	**FPU Input Denormal Interrupt Enable** 0b - Disable interrupt 1b - Enable interrupt
3 FPOFCIE	**FPU Overflow Interrupt Enable** 0b - Disable interrupt 1b - Enable interrupt
2 FPUFCIE	**FPU Underflow Interrupt Enable** 0b - Disable interrupt 1b - Enable interrupt
1 FPDZCIE	**FPU Divide-by-Zero Interrupt Enable** 0b - Disable interrupt 1b - Enable interrupt
0 FPIOCIE	**FPU Invalid Operation Interrupt Enable** 0b - Disable interrupt 1b - Enable interrupt

12.2.1.4 CIM SWTRIG Register

Offset: 14h

Yuntu Microelectronics

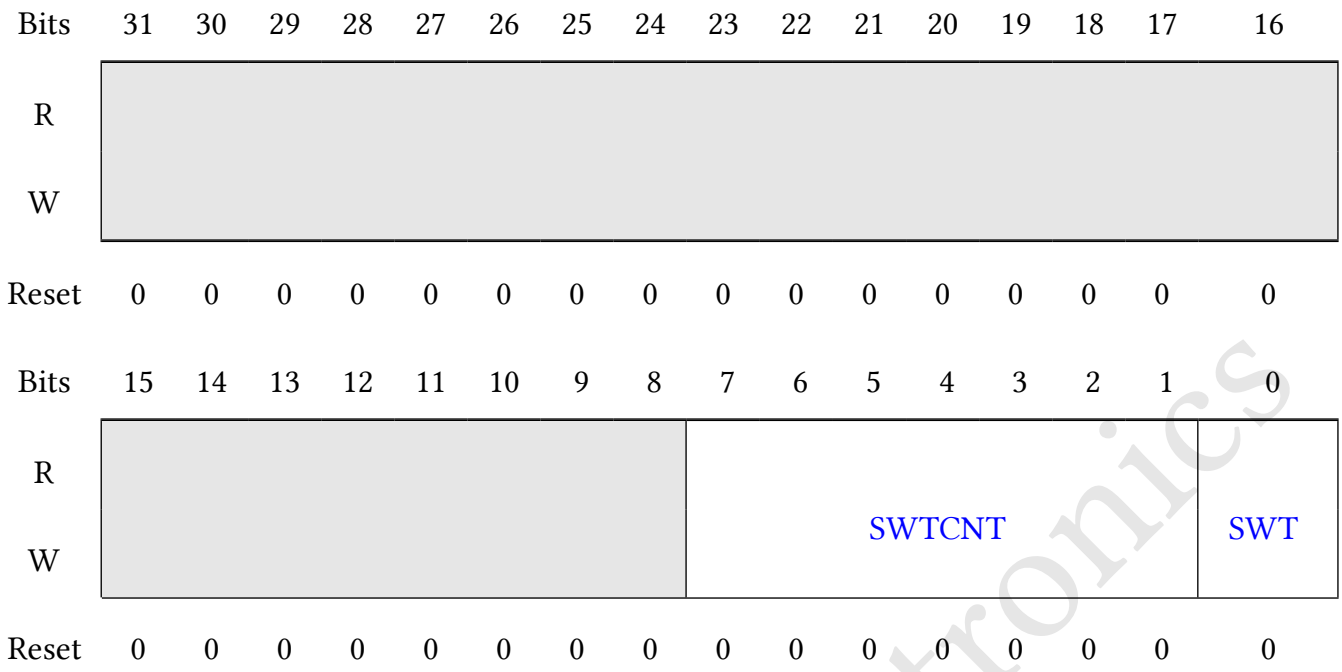
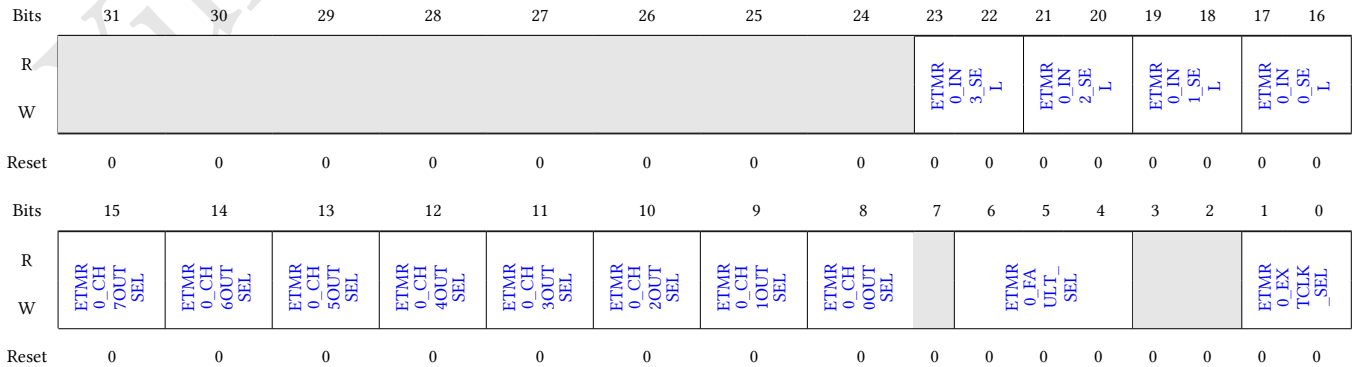


Table 12.5: CIM SWTRIG Register Description

Field	Function
7 - 1 SWTCNT	**Software Trigger Count Register** Define software trigger valid length (multiplied with bus clock cycle), SWTCNT value will decrease until equal to zero when SWT set.
0 SWT	**Software Trigger** Generate software trigger. 0b - Software trigger is disabled. 1b - Generate software trigger and it keeps until SWTCNT equals to zero.

12.2.1.5 CIM ETMROPT0 Register

Offset: 18h



Yuntu Microelectronics

Table 12.6: CIM ETMROPT0 Register Description

Field	Function
23 - 22 ETMR0_IN 3_SEL	**eTMR0 Channel3 Input Select** Select eTMR0 channel3 input sources. 00b - Select ETMR0_CH3 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR0_CH0/1/2) 11b - Reserved
21 - 20 ETMR0_IN 2_SEL	**eTMR0 Channel2 Input Select** Select eTMR0 channel2 input sources. 00b - Select ETMR0_CH2 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR0_CH0/1/2) 11b - Reserved
19 - 18 ETMR0_IN 1_SEL	**eTMR0 Channel1 Input Select** Select eTMR0 channel1 input sources. 00b - Select ETMR0_CH1 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR0_CH0/1/2) 11b - Reserved
17 - 16 ETMR0_IN 0_SEL	**eTMR0 Channel0 Input Select** Select eTMR0 channel0 input sources. 00b - Select ETMR0_CH0 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR0_CH0/1/2) 11b - Reserved
15 ETMR0_CH 7OUTSEL	**eTMR0 Channel 7 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 7. 0b - eTMR0 channel 7 output to pad directly. 1b - eTMR0 channel 7 output modulation with eTMR1_CH1 output.
14 ETMR0_CH 6OUTSEL	**eTMR0 Channel 6 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 6. 0b - eTMR0 channel 6 output to pad directly. 1b - eTMR0 channel 6 output modulation with eTMR1_CH1 output.
13 ETMR0_CH 5OUTSEL	**eTMR0 Channel 5 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 5. 0b - eTMR0 channel 5 output to pad directly. 1b - eTMR0 channel 5 output modulation with eTMR1_CH1 output.

Yuntu Microelectronics

Field	Function
12 ETMR0_CH 4OUTSEL	**eTMR0 Channel 4 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 4. 0b - eTMR0 channel 4 output to pad directly. 1b - eTMR0 channel 4 output modulation with eTMR1_CH1 output.
11 ETMR0_CH 3OUTSEL	**eTMR0 Channel 3 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 3. 0b - eTMR0 channel 3 output to pad directly. 1b - eTMR0 channel 3 output modulation with eTMR1_CH1 output.
10 ETMR0_CH 2OUTSEL	**eTMR0 Channel 2 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 2. 0b - eTMR0 channel 2 output to pad directly. 1b - eTMR0 channel 2 output modulation with eTMR1_CH1 output.
9 ETMR0_CH 1OUTSEL	**eTMR0 Channel 1 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 1. 0b - eTMR0 channel 1 output to pad directly. 1b - eTMR0 channel 1 output modulation with eTMR1_CH1 output.
8 ETMR0_CH 0OUTSEL	**eTMR0 Channel 0 Output Modulation Select** Select modulation with eTMR1_CH1 for eTMR0 channel 0. 0b - eTMR0 channel 0 output to pad directly. 1b - eTMR0 channel 0 output modulation with eTMR1_CH1 output.
6 - 4 ETMR0_FA ULT_SEL	**eTMR0 Fault Select** Select fault source for eTMR0. 000b - eTMR0 fault 0 input from pad. 001b - eTMR0 fault 0 input from TMU. 000b - eTMR0 fault 1 input from pad. 010b - eTMR0 fault 1 input from TMU. 000b - eTMR0 fault 2 input from pad. 100b - eTMR0 fault 2 input from TMU.
1 - 0 ETMR0_EX TCLK_SEL	**eTMR0 External Clock Select** Select external clock source for eTMR0. 00b - Select TCLK_IN0 input as external clock for eTMR0. 01b - Select TCLK_IN1 input as external clock for eTMR0. 10b - Select TCLK_IN2 input as external clock for eTMR0. 11b - Select ETMR0_FUNC_CLK from IPC as external clock for eTMR0.

12.2.1.6 CIM ETMROPT1 Register

Offset: 1ch

Yuntu Microelectronics

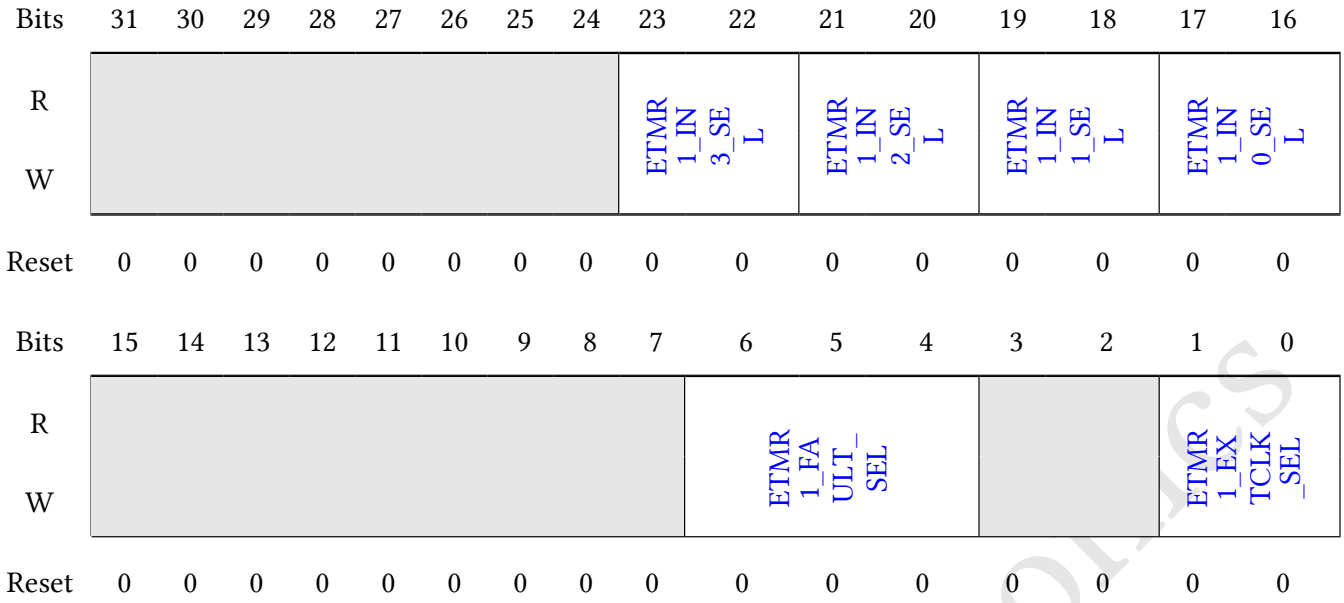


Table 12.7: CIM ETMROPT1 Register Description

Field	Function
23 - 22 ETMR1_IN 3_SEL	**eTMR1 Channel3 Input Select** Select eTMR1 channel3 input sources. 00b - Select ETMR1_CH3 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR1_CH0/1/2) 11b - Reserved
21 - 20 ETMR1_IN 2_SEL	**eTMR1 Channel2 Input Select** Select eTMR1 channel2 input sources. 00b - Select ETMR1_CH2 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR1_CH0/1/2) 11b - Reserved
19 - 18 ETMR1_IN 1_SEL	**eTMR1 Channel1 Input Select** Select eTMR1 channel1 input sources. 00b - Select ETMR1_CH1 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR1_CH0/1/2) 11b - Reserved
17 - 16	**eTMR1 Channel0 Input Select**

Yuntu Microelectronics

Field	Function
ETMR1_IN 0_SEL	Select eTMR1 channel0 input sources. 00b - Select ETMR1_CH0 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR1_CH0/1/2) 11b - Reserved
6 - 4 ETMR1_FA ULT_SEL	**eTMR1 Fault Select** Select fault source for eTMR1. 000b - eTMR1 fault 0 input from pad. 001b - eTMR1 fault 0 input from TMU. 000b - eTMR1 fault 1 input from pad. 010b - eTMR1 fault 1 input from TMU. 000b - eTMR1 fault 2 input from pad. 100b - eTMR1 fault 2 input from TMU.
1 - 0 ETMR1_EX TCLK_SEL	**eTMR1 External Clock Select** Select external clock source for eTMR1. 00b - Select TCLK_IN0 input as external clock for eTMR1. 01b - Select TCLK_IN1 input as external clock for eTMR1. 10b - Select TCLK_IN2 input as external clock for eTMR1. 11b - Select ETMR1_FUNC_CLK from IPC as external clock for eTMR1.

12.2.1.7 CIM ETMROPT2 Register

Offset: 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W									ETMR 2_IN 3_SE L		ETMR 2_IN 2_SE L		ETMR 2_IN 1_SE L		ETMR 2_IN 0_SE L	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W										ETMR 2_FA ULT_ SEL					ETMR 2_EX TCLK _SEL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 12.8: CIM ETMROPT2 Register Description

Field	Function
23 - 22 ETMR2_IN 3_SEL	**eTMR2 Channel3 Input Select** Select eTMR2 channel3 input sources. 00b - Select ETMR2_CH3 input from pad 01b - Select TMU output 10b - Hall sensor XOR input 1 (XOR of ETMR2_CH0/1/2) 11b - Hall sensor XOR input 2 (XOR of ETMR2_CH0,ETMR2_CH1,ETMR1_CH1)
21 - 20 ETMR2_IN 2_SEL	**eTMR2 Channel2 Input Select** Select eTMR2 channel2 input sources. 00b - Select ETMR2_CH2 input from pad 01b - Select TMU output 10b - Hall sensor XOR input 1 (XOR of ETMR2_CH0/1/2) 11b - Hall sensor XOR input 2 (XOR of ETMR2_CH0,ETMR2_CH1,ETMR1_CH1)
19 - 18 ETMR2_IN 1_SEL	**eTMR2 Channel1 Input Select** Select eTMR2 channel1 input sources. 00b - Select ETMR2_CH1 input from pad 01b - Select TMU output 10b - Hall sensor XOR input 1 (XOR of ETMR2_CH0/1/2) 11b - Hall sensor XOR input 2 (XOR of ETMR2_CH0,ETMR2_CH1,ETMR1_CH1)
17 - 16 ETMR2_IN 0_SEL	**eTMR2 Channel0 Input Select** Select eTMR2 channel0 input sources. 00b - Select ETMR2_CH0 input from pad 01b - Select TMU output 10b - Hall sensor XOR input 1 (XOR of ETMR2_CH0/1/2) 11b - Hall sensor XOR input 2 (XOR of ETMR2_CH0,ETMR2_CH1,ETMR1_CH1)
6 - 4 ETMR2_FA ULT_SEL	**eTMR2 Fault Select** Select fault source for eTMR2. 000b - eTMR2 fault 0 input from pad. 001b - eTMR2 fault 0 input from TMU. 000b - eTMR2 fault 1 input from pad. 010b - eTMR2 fault 1 input from TMU. 000b - eTMR2 fault 2 input from pad. 100b - eTMR2 fault 2 input from TMU.
1 - 0	**eTMR2 External Clock Select**

Yuntu Microelectronics

Field	Function
ETMR2_EX TCLK_SEL	Select external clock source for eTMR2. 00b - Select TCLK_IN0 input as external clock for eTMR2. 01b - Select TCLK_IN1 input as external clock for eTMR2. 10b - Select TCLK_IN2 input as external clock for eTMR2. 11b - Select ETMR2_FUNC_CLK from IPC as external clock for eTMR2.

12.2.1.8 CIM ETMROPT3 Register

Offset: 24h

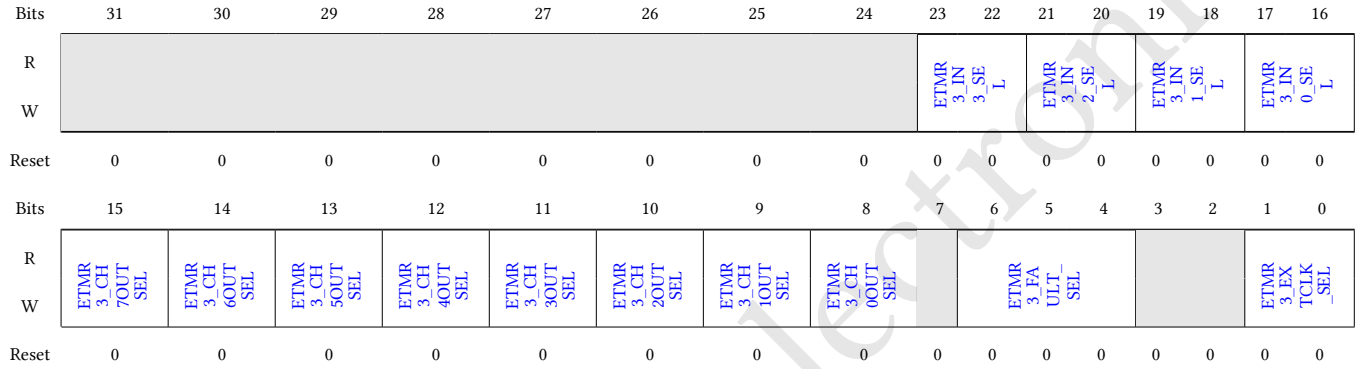


Table 12.9: CIM ETMROPT3 Register Description

Field	Function
23 - 22 ETMR3_IN 3_SEL	**eTMR3 Channel3 Input Select** Select eTMR3 channel3 input sources. 00b - Select ETMR3_CH3 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR3_CH0/1/2) 11b - Reserved
21 - 20 ETMR3_IN 2_SEL	**eTMR3 Channel2 Input Select** Select eTMR3 channel2 input sources. 00b - Select ETMR3_CH2 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR3_CH0/1/2) 11b - Reserved
19 - 18 ETMR3_IN 1_SEL	**eTMR3 Channel1 Input Select** Select eTMR3 channel1 input sources. 00b - Select ETMR3_CH1 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR3_CH0/1/2) 11b - Reserved

Yuntu Microelectronics

Field	Function
17 - 16 ETMR3_IN 0_SEL	**eTMR3 Channel0 Input Select** Select eTMR3 channel0 input sources. 00b - Select ETMR3_CH0 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR3_CH0/1/2) 11b - Reserved
15 ETMR3_CH 7OUTSEL	**eTMR3 Channel 7 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 7. 0b - eTMR3 channel 7 output to pad directly. 1b - eTMR3 channel 7 output modulation with eTMR2_CH1 output.
14 ETMR3_CH 6OUTSEL	**eTMR3 Channel 6 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 6. 0b - eTMR3 channel 6 output to pad directly. 1b - eTMR3 channel 6 output modulation with eTMR2_CH1 output.
13 ETMR3_CH 5OUTSEL	**eTMR3 Channel 5 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 5. 0b - eTMR3 channel 5 output to pad directly. 1b - eTMR3 channel 5 output modulation with eTMR2_CH1 output.
12 ETMR3_CH 4OUTSEL	**eTMR3 Channel 4 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 4. 0b - eTMR3 channel 4 output to pad directly. 1b - eTMR3 channel 4 output modulation with eTMR2_CH1 output.
11 ETMR3_CH 3OUTSEL	**eTMR3 Channel 3 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 3. 0b - eTMR3 channel 3 output to pad directly. 1b - eTMR3 channel 3 output modulation with eTMR2_CH1 output.
10 ETMR3_CH 2OUTSEL	**eTMR3 Channel 2 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 2. 0b - eTMR3 channel 2 output to pad directly. 1b - eTMR3 channel 2 output modulation with eTMR2_CH1 output.
9 ETMR3_CH 1OUTSEL	**eTMR3 Channel 1 Output Modulation Select** Select modulation with eTMR2_CH1 for eTMR3 channel 1. 0b - eTMR3 channel 1 output to pad directly. 1b - eTMR3 channel 1 output modulation with eTMR2_CH1 output.
8	**eTMR3 Channel 0 Output Modulation Select**

Yuntu Microelectronics

Field	Function
ETMR3_CH 0OUTSEL	Select modulation with eTMR2_CH1 for eTMR3 channel 0. 0b - eTMR3 channel 0 output to pad directly. 1b - eTMR3 channel 0 output modulation with eTMR2_CH1 output.
6 - 4 ETMR3_FA ULT_SEL	**eTMR3 Fault Select** Select fault source for eTMR3. 000b - eTMR3 fault 0 input from pad. 001b - eTMR3 fault 0 input from TMU. 000b - eTMR3 fault 1 input from pad. 010b - eTMR3 fault 1 input from TMU. 000b - eTMR3 fault 2 input from pad. 100b - eTMR3 fault 2 input from TMU.
1 - 0 ETMR3_EX TCLK_SEL	**eTMR3 External Clock Select** Select external clock source for eTMR3. 00b - Select TCLK_IN0 input as external clock for eTMR3. 01b - Select TCLK_IN1 input as external clock for eTMR3. 10b - Select TCLK_IN2 input as external clock for eTMR3. 11b - Select ETMR3_FUNC_CLK from IPC as external clock for eTMR3.

12.2.1.9 CIM ETMROPT4 Register

Offset: 28h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R									ETMR4_IN3_SEL		ETMR4_IN2_SEL		ETMR4_IN1_SEL		ETMR4_IN0_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R															ETMR4_EXTCLK_SEL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 12.10: CIM ETMROPT4 Register Description

Field	Function
23 - 22 ETMR4_IN 3_SEL	**eTMR4 Channel3 Input Select** Select eTMR4 channel3 input sources. 00b - Select ETMR4_CH3 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR4_CH0/1/2) 11b - Reserved
21 - 20 ETMR4_IN 2_SEL	**eTMR4 Channel2 Input Select** Select eTMR4 channel2 input sources. 00b - Select ETMR4_CH2 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR4_CH0/1/2) 11b - Reserved
19 - 18 ETMR4_IN 1_SEL	**eTMR4 Channel1 Input Select** Select eTMR4 channel1 input sources. 00b - Select ETMR4_CH1 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR4_CH0/1/2) 11b - Reserved
17 - 16 ETMR4_IN 0_SEL	**eTMR4 Channel0 Input Select** Select eTMR4 channel0 input sources. 00b - Select ETMR4_CH0 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR4_CH0/1/2) 11b - Reserved
1 - 0 ETMR4_EX TCLK_SEL	**eTMR4 External Clock Select** Select external clock source for eTMR4. 00b - Select TCLK_IN0 input as external clock for eTMR4. 01b - Select TCLK_IN1 input as external clock for eTMR4. 10b - Select TCLK_IN2 input as external clock for eTMR4. 11b - Select ETMR4_FUNC_CLK from IPC as external clock for eTMR4.

Yuntu Microelectronics

12.2.1.10 CIM ETMROPT5 Register

Offset: 2ch

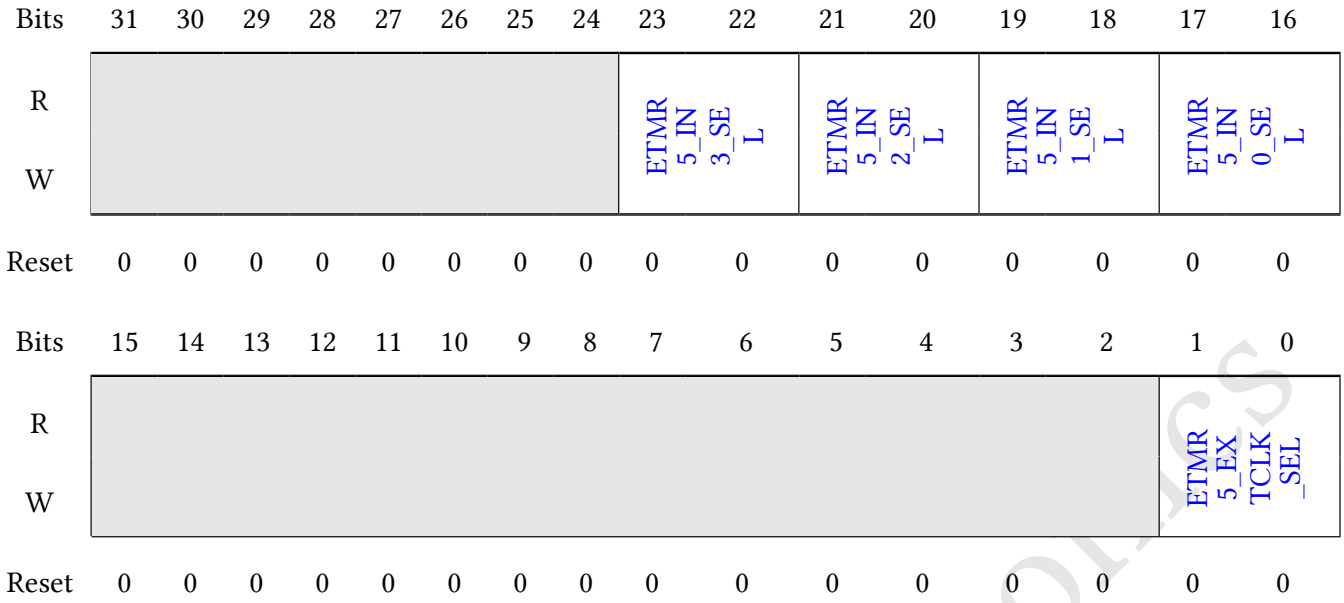


Table 12.11: CIM ETMROPT5 Register Description

Field	Function
23 - 22 ETMR5_IN 3_SEL	**eTMR5 Channel3 Input Select** Select eTMR5 channel3 input sources. 00b - Select ETMR5_CH3 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR5_CH0/1/2) 11b - Reserved
21 - 20 ETMR5_IN 2_SEL	**eTMR5 Channel2 Input Select** Select eTMR5 channel2 input sources. 00b - Select ETMR5_CH2 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR5_CH0/1/2) 11b - Reserved
19 - 18 ETMR5_IN 1_SEL	**eTMR5 Channel1 Input Select** Select eTMR5 channel1 input sources. 00b - Select ETMR5_CH1 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR5_CH0/1/2) 11b - Reserved
17 - 16	**eTMR5 Channel0 Input Select**

Yuntu Microelectronics

Field	Function
ETMR5_IN 0_SEL	Select eTMR5 channel0 input sources. 00b - Select ETMR5_CH0 input from pad 01b - Select TMU output 10b - Hall sensor XOR input (XOR of ETMR5_CH0/1/2) 11b - Reserved
1 - 0 ETMR5_EX TCLK_SEL	**eTMR5 External Clock Select** Select external clock source for eTMR5. 00b - Select TCLK_IN0 input as external clock for eTMR5. 01b - Select TCLK_IN1 input as external clock for eTMR5. 10b - Select TCLK_IN2 input as external clock for eTMR5. 11b - Select ETMR5_FUNC_CLK from IPC as external clock for eTMR5.

12.2.1.11 CIM ETMR_GLB_OPT Register

Offset: 3ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		ETMR_GTB_E_SEL									ETMR5SSC	ETMR4SSC	ETMR3SSC	ETMR2SSC	ETMR1SSC	ETMR0SSC
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 12.12: CIM ETMR_GLB_OPT Register Description

Field	Function
14 ETMR_GTB E_SEL	**eTMR GTB split enable/disable bit** Split time base enable/disable 0b - All eTMRs have a single global timebase 1b - eTMR0~3 have a common timebase and others have a different common timebase.
5 ETMR5SS1 C	**eTMR5 Synchronize Source 1 Control** Generate synchronize signal for eTMR5 trigger 1. 0b - Don't generate synchronize signal for eTMR5 trigger 1. 1b - Generate synchronize signal for eTMR5 trigger 1.

Yuntu Microelectronics

Field	Function
4 ETMR4SS1C	**eTMR4 Synchronize Source 1 Control** Generate synchronize signal for eTMR4 trigger 1. 0b - Don't generate synchronize signal for eTMR4 trigger 1. 1b - Generate synchronize signal for eTMR4 trigger 1.
3 ETMR3SS1C	**eTMR3 Synchronize Source 1 Control** Generate synchronize signal for eTMR3 trigger 1. 0b - Don't generate synchronize signal for eTMR3 trigger 1. 1b - Generate synchronize signal for eTMR3 trigger 1.
2 ETMR2SS1C	**eTMR2 Synchronize Source 1 Control** Generate synchronize signal for eTMR2 trigger 1. 0b - Don't generate synchronize signal for eTMR2 trigger 1. 1b - Generate synchronize signal for eTMR2 trigger 1.
1 ETMR1SS1C	**eTMR1 Synchronize Source 1 Control** Generate synchronize signal for eTMR1 trigger 1. 0b - Don't generate synchronize signal for eTMR1 trigger 1. 1b - Generate synchronize signal for eTMR1 trigger 1.
0 ETMR0SS1C	**eTMR0 Synchronize Source 1 Control** Generate synchronize signal for eTMR0 trigger 1. 0b - Don't generate synchronize signal for eTMR0 trigger 1. 1b - Generate synchronize signal for eTMR0 trigger 1.

Yuntu Microelectronics

12.2.1.12 CIM SWFAULT Register

Offset: 40h

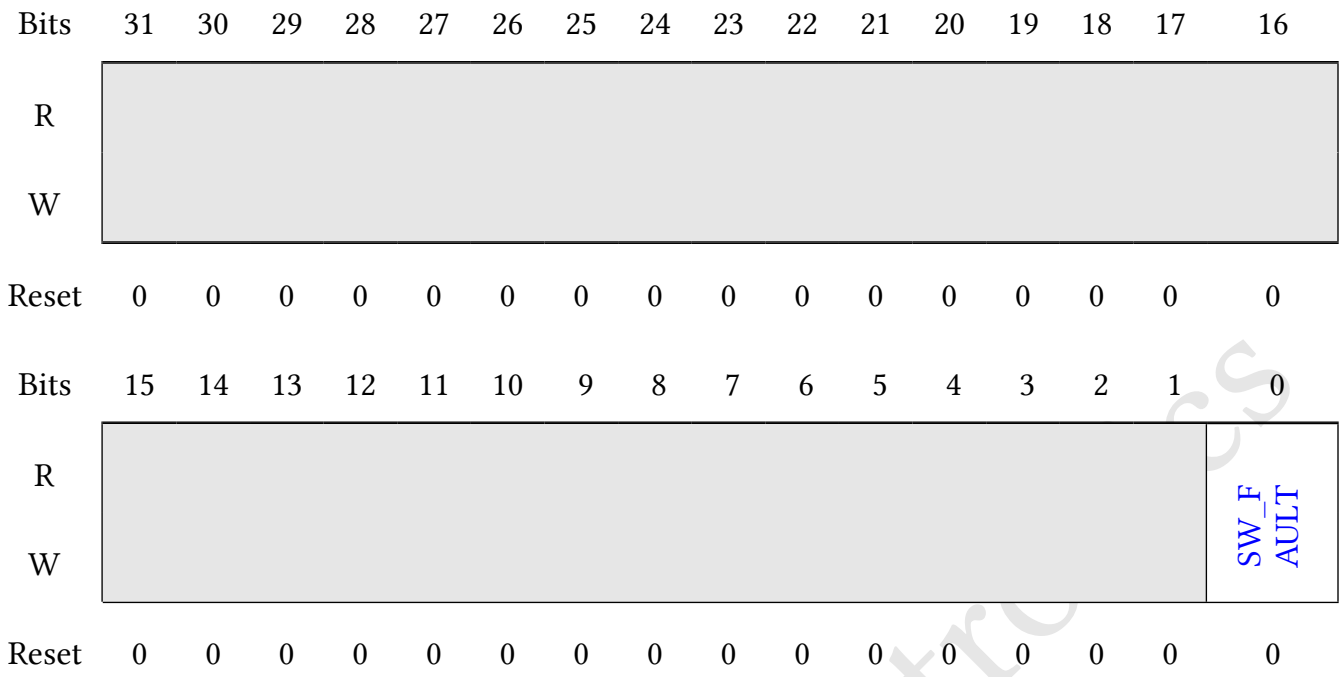
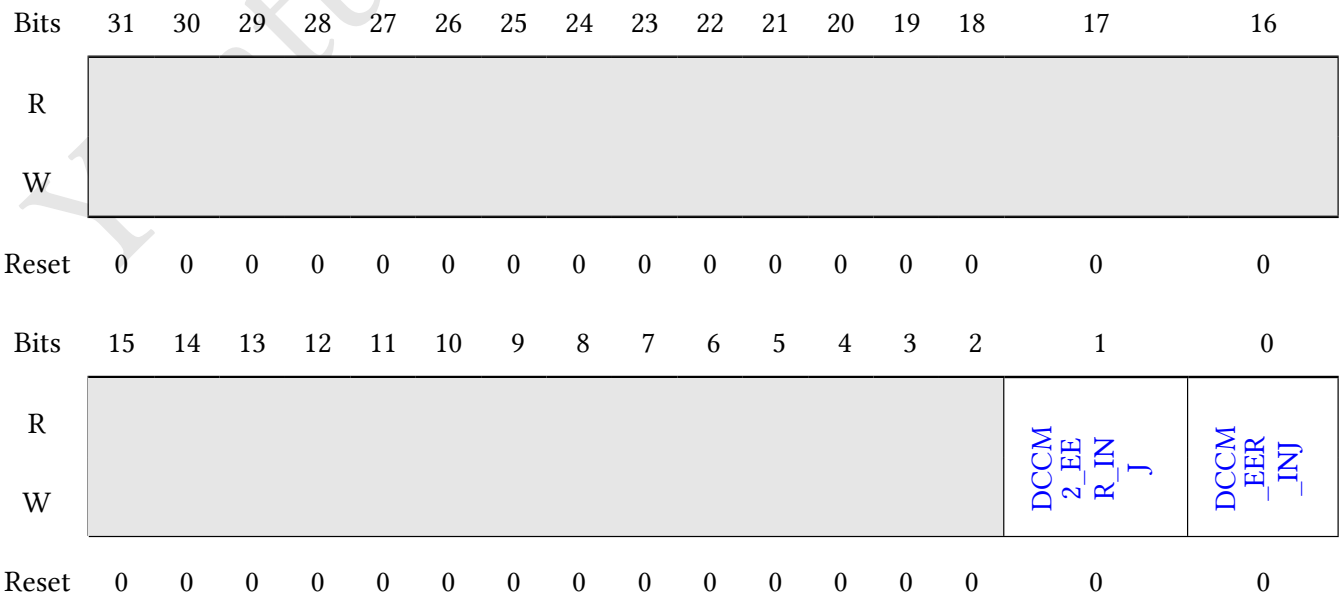


Table 12.13: CIM SWFAULT Register Description

Field	Function
0 SW_FAULT	**Software fault generate bit** Writing 1 to this bit will generate a software fault to FMU.Writing 0 to clear. 0b - No software fault 1b - Software fault assert.

12.2.1.13 CIM DCCM_EI Register

Offset: 44h



Yuntu Microelectronics

Table 12.14: CIM DCCM_EI Register Description

Field	Function
1 DCCM2_EER_INJ	**Lockstep Core Compare Error Injection** Writing 1 to this bit will generate a lockstep fault. It is auto cleared after one pclk cycle.
0 DCCM_EER_INJ	**Main Core Compare Error Injection** Writing 1 to this bit will generate a lockstep fault. It is auto cleared after one pclk cycle.

12.2.1.14 CIM UID0 Register

Offset: 80h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UID0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 12.15: CIM UID0 Register Description

Field	Function
31 - 0 UID0	**Unique Identification** Unique identification for the chip.

12.2.1.15 CIM UID1 Register

Offset: 84h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UID1															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID1															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 12.16: CIM UID1 Register Description

Field	Function
31 - 0 UID1	**Unique Identification** Unique identification for the chip.

12.2.1.16 CIM UID2 Register

Offset: 88h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UID2															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID2															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Yuntu Microelectronics

Table 12.17: CIM UID2 Register Description

Field	Function
31 - 0 UID2	**Unique Identification** Unique identification for the chip.

12.2.1.17 CIM UID3 Register

Offset: 8ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	UID3															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID3															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 12.18: CIM UID3 Register Description

Field	Function
31 - 0 UID3	**Unique Identification** Unique identification for the chip.

12.2.1.18 CIM DIEINFO Register

Offset: 90h

Yuntu Microelectronics

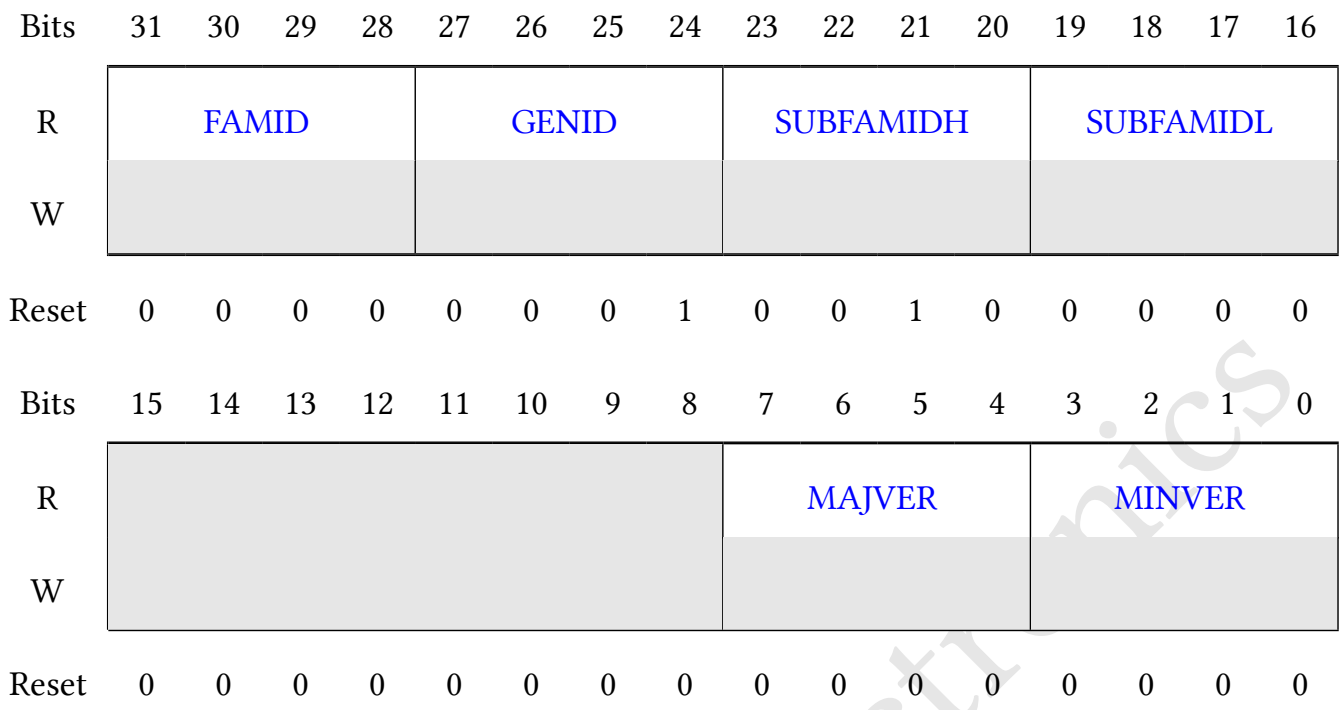


Table 12.19: CIM DIEINFO Register Description

Field	Function
31 - 28 FAMID	**Family ID** Describe the product family information. 0000b - B - Basic(General) 0001b - D - Dashboard 0010b - P - Powertrain 0011b - V - Vision 0100b - N - Networks
27 - 24 GENID	**Generation ID** Describe the product generation information. 0001b - 1st generation production 0010b - 2nd generation production 0011b - 3rd generation production
23 - 20 SUBFAMID H	**Sub Family ID High** Describe the product sub family information. 0000b - L - Low end 0001b - M - Middle end 0010b - H - High end
19 - 16	**Sub Family ID Low**

Yuntu Microelectronics

Field	Function
SUBFAMID L	Describe the product max Flash size information. 0000b - A - 2M 0001b - B - 4M 0010b - C - 6M 0011b - D - 8M 0100b - E - 16MB
7 - 4 MAJVER	**Major Version Information** Describe the major version information.
3 - 0 MINVER	**Minor Version Information** Describe the minor version information.

12.2.1.19 CIM CHIPINFO0 Register

Offset: 94h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R									PKGS				OCRAMS			
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DFLASHS				PFLASHS											
W																
Reset	1	0	0	0	1	0	1	1	0	0	0	0	0	0	1	0

Table 12.20: CIM CHIPINFO0 Register Description

Field	Function
23 - 20	**Package Size**

Yuntu Microelectronics

Field	Function
PKGS	Describe chip package type information. 0000b - 16 TSOP 0001b - 32 LQFP 0010b - 32 QFN 0011b - 48 LQFP 0100b - 64 LQFP 1000b - 100 LQFP 1010b - 144 LQFP 1100b - 176 LQFP 1101b - 257 BGA 1111b - 289 BGA
19 - 16 OCRAMS	**OCRAM Size** Describe ocram size information. 0010b - 1KB 0011b - 2KB 0100b - 4KB 0101b - 8KB 0110b - 16KB 0111b - 32KB 1000b - 64KB 1001b - 128KB 1010b - 256KB 1011b - 512KB 1100b - 1MB 1101b - 2MB 1110b - 4MB 1111b - 8MB
15 - 12 DFLASHS	**DFLASH Size** Describe data flash size information. 0000b - 1KB 0001b - 2KB 0010b - 4KB 0011b - 8KB 0100b - 16KB 0101b - 32KB 0110b - 64KB 0111b - 128KB 1000b - 256KB 1001b - 512KB
11 - 8	**PFLASH Size**

Yuntu Microelectronics

Field	Function
PFLASHS	Describe program flash size information. 0000b - 1KB 0001b - 2KB 0010b - 4KB 0011b - 8KB 0100b - 16KB 0101b - 32KB 0110b - 64KB 0111b - 128KB 1000b - 256KB 1001b - 512KB 1010b - 1MB 1011b - 2MB 1100b - 4MB 1101b - 8MB 1110b - 16MB 1111b - 32MB

12.2.1.20 CIM CHIPINFO1 Register

Offset: 98h

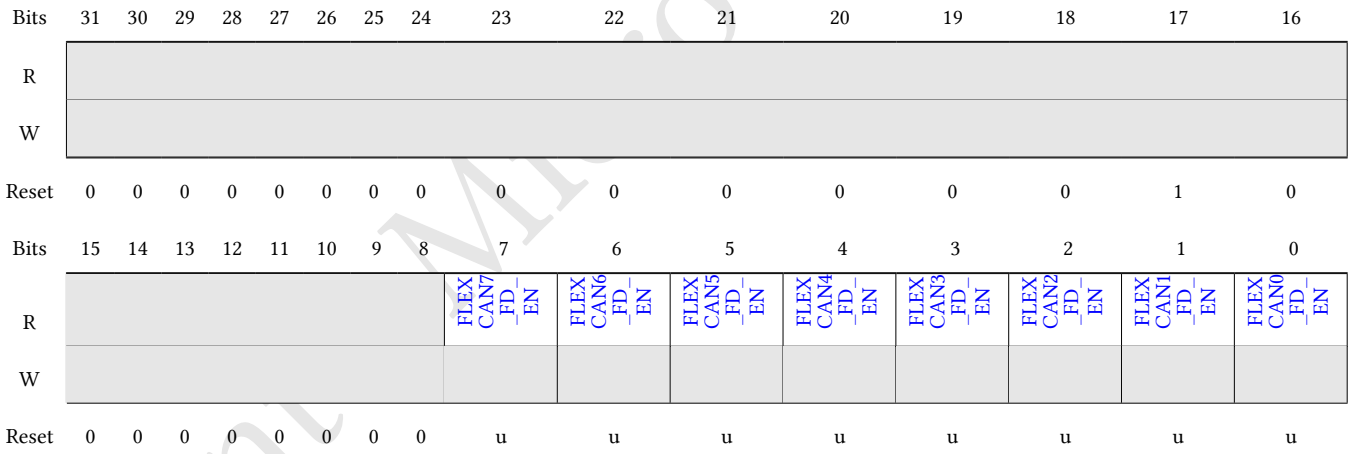


Table 12.21: CIM CHIPINFO1 Register Description

Field	Function
7 FLEXCAN7_FD_EN	**FlexCAN7 FD Feature Enable** FlexCAN7 FD Feature Enable/Disable 0b - FlexCAN7 FD Feature is disabled 1b - FlexCAN7 FD Feature is enabled
6	**FlexCAN6 FD Feature Enable**

Yuntu Microelectronics

Field	Function
FLEXCAN6 _FD_EN	FlexCAN6 FD Feature Enable/Disable 0b - FlexCAN6 FD Feature is disabled 1b - FlexCAN6 FD Feature is enabled
5 FLEXCAN5 _FD_EN	**FlexCAN5 FD Feature Enable** FlexCAN5 FD Feature Enable/Disable 0b - FlexCAN5 FD Feature is disabled 1b - FlexCAN5 FD Feature is enabled
4 FLEXCAN4 _FD_EN	**FlexCAN4 FD Feature Enable** FlexCAN4 FD Feature Enable/Disable 0b - FlexCAN4 FD Feature is disabled 1b - FlexCAN4 FD Feature is enabled
3 FLEXCAN3 _FD_EN	**FlexCAN3 FD Feature Enable** FlexCAN3 FD Feature Enable/Disable 0b - FlexCAN3 FD Feature is disabled 1b - FlexCAN3 FD Feature is enabled
2 FLEXCAN2 _FD_EN	**FlexCAN2 FD Feature Enable** FlexCAN2 FD Feature Enable/Disable 0b - FlexCAN2 FD Feature is disabled 1b - FlexCAN2 FD Feature is enabled
1 FLEXCAN1 _FD_EN	**FlexCAN1 FD Feature Enable** FlexCAN1 FD Feature Enable/Disable 0b - FlexCAN1 FD Feature is disabled 1b - FlexCAN1 FD Feature is enabled
0 FLEXCAN0 _FD_EN	**FlexCAN0 FD Feature Enable** FlexCAN0 FD Feature Enable/Disable 0b - FlexCAN0 FD Feature is disabled 1b - FlexCAN0 FD Feature is enabled

Yuntu Microelectronics

12.2.1.21 CIM CPUINFO Register

Offset: a0h

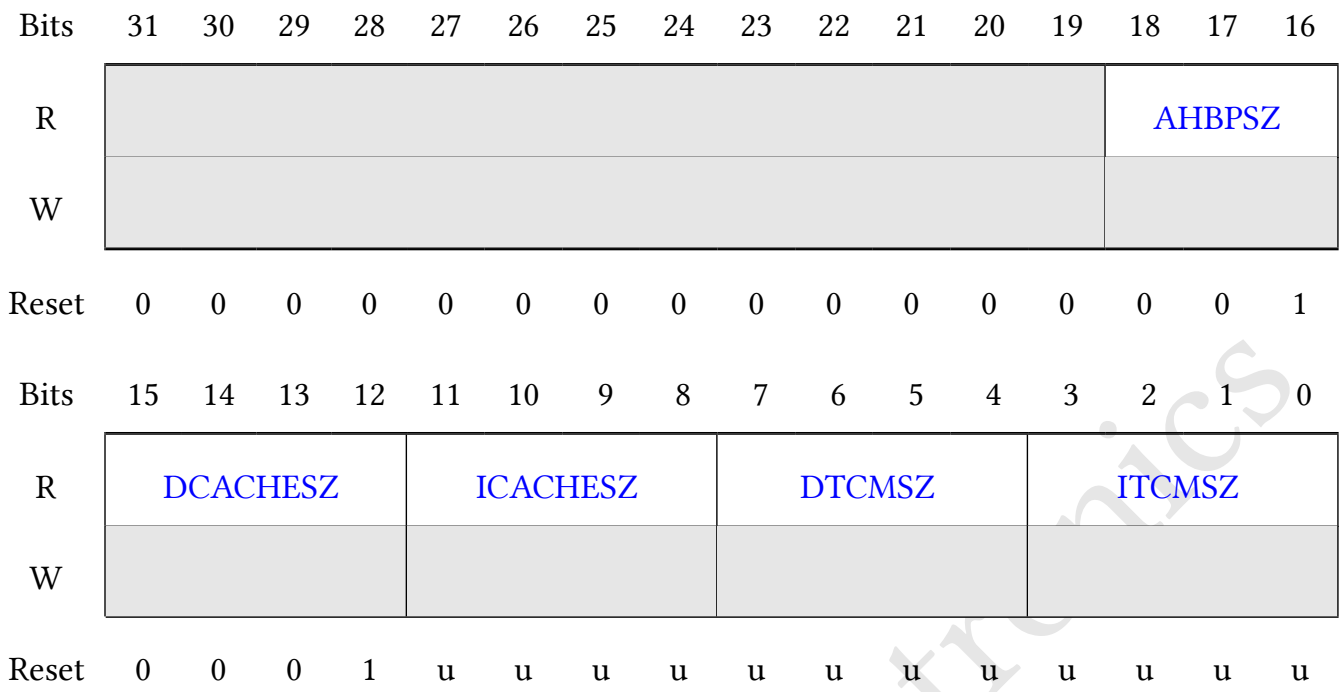


Table 12.22: CIM CPUINFO Register Description

Field	Function
18 - 16 AHBPSZ	**AHBP Size** Describe AHBP size information. 000b - AHBP disabled 001b - 64MB 010b - 128MB 011b - 256MB 100b - 512MB
15 - 12 DCACHESZ	**Data Cache Size** Describe Data Cache size information. 0000b - 4KB 0001b - 8KB 0011b - 16KB 0111b - 32KB 1111b - 64KB
11 - 8 ICACHESZ	**Instructin Cache Size** Describe Instruction Cache size information 0000b - 4KB 0001b - 8KB 0011b - 16KB 0111b - 32KB 1111b - 64KB
7 - 4	**DTCM Size**

Yuntu Microelectronics

Field	Function
DTCMSZ	Describe DTCM size information 0000b - No TCM implemented 0001b - Reserved 0010b - Reserved 0011b - 4KB 0100b - 8KB 0101b - 16KB 0110b - 32KB 0111b - 64KB 1000b - 128KB 1001b - 256KB 1010b - 512KB 1011b - 1MB 1100b - 2MB 1101b - 4MB 1110b - 8MB 1111b - 16MB
3 - 0 ITCMSZ	**ITCM Size** Describe ITCM size information 0000b - No TCM implemented 0001b - Reserved 0010b - Reserved 0011b - 4KB 0100b - 8KB 0101b - 16KB 0110b - 32KB 0111b - 64KB 1000b - 128KB 1001b - 256KB 1010b - 512KB 1011b - 1MB 1100b - 2MB 1101b - 4MB 1110b - 8MB 1111b - 16MB

Yuntu Microelectronics

12.2.1.22 CIM BOOT_CFG Register

Write to BOOT_CFG is forbidden, as the value loaded from IFR will be changed.

Offset: c0h

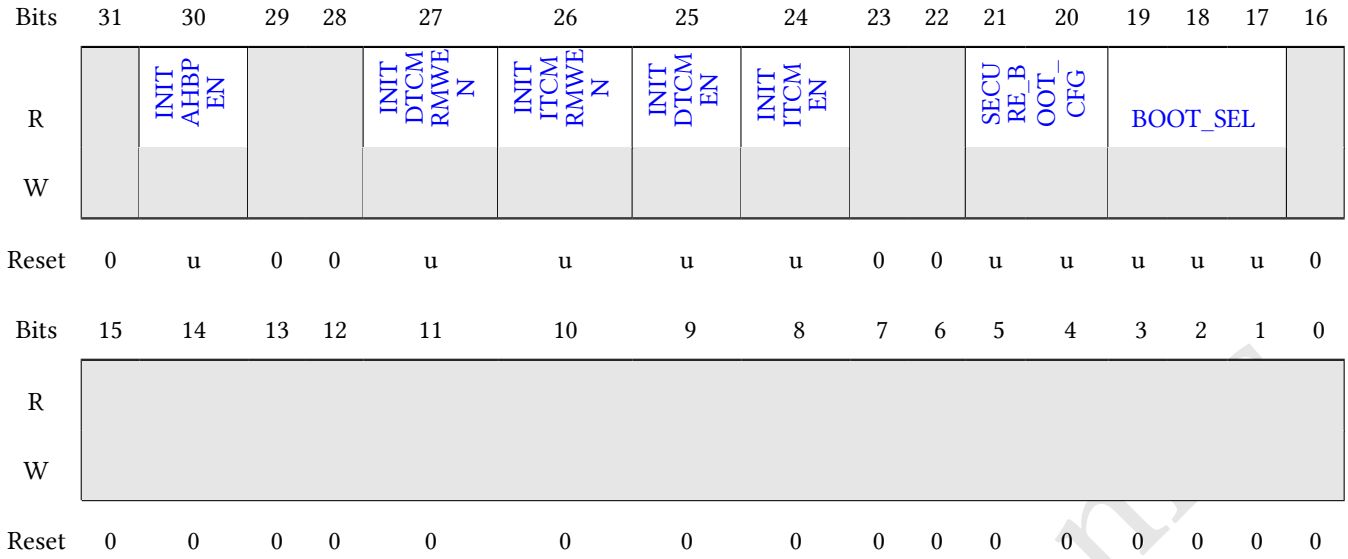


Table 12.23: CIM BOOT_CFG Register Description

Field	Function
30 INITAHBP EN	**AHBP enable initialization out of reset** Configure AHBP enable or disable out of reset 0b - AHBP disable out of reset 1b - AHBP enable out of reset
27 INITDTCM RMWEN	**DTCM read modify write enable initialization out of reset** Configure DTCM read modify write enable or disable out of reset 0b - DTCM read modify write disable out of reset 1b - DTCM read modify write enable out of reset
26 INITITCM RMWEN	**ITCM read modify write enable initialization out of reset** Configure ITCM read modify write enable or disable out of reset 0b - ITCM read modify write disable out of reset 1b - ITCM read modify write enable out of reset
25 INITDTCM EN	**DTCM enable initialization out of reset** Configure DTCM enable or disable out of reset 0b - DTCM disable out of reset 1b - DTCM enable out of reset
24 INITITCM EN	**ITCM enable initialization out of reset** Configure ITCM enable or disable out of reset 0b - ITCM disable out of reset 1b - ITCM enable out of reset
21 - 20	**Secure boot configuration**

Yuntu Microelectronics

Field	Function
SECURE_BOOT_CFG	Configure secure boot from ROM 00b - Nonsecure boot 01b - Nonsecure boot 10b - Secure boot from ROM 11b - Nonsecure boot
19 - 17 BOOT_SEL	**Chip Boot Select** Select the base address chip boot from 000b - Flash base 001b - Reserved 010b - Reserved 011b - ROM base 100b - Reserved 101b - DFLASH base 110b - Reserved 111b - Reserved

Yuntu Microelectronics

Chapter 13

Trigger Multiplexer Module (TMU)

13.1 Introduction

The Trigger Multiplexer Unit (TMU) provides an extremely flexible mechanism for connecting various trigger sources to multiple pins/peripherals.

With the TMU, each peripheral that accepts external triggers usually has one specific 8-bit trigger control register. Each control register supports up to 128 triggers, and each trigger can be selected from the available input triggers.

13.2 Trigger Interconnectivity Table

Table 13.1: Trigger MUX Table

Slot	Trigger input	Trigger output
0	VSS	-
1	VDD	-
2	TMU_IN0	-
3	TMU_IN1	-
4	TMU_IN2	TMU_OUT0
5	TMU_IN3	TMU_OUT1
6	TMU_IN4	TMU_OUT2
7	TMU_IN5	TMU_OUT3
8	TMU_IN6	TMU_OUT4
9	TMU_IN7	TMU_OUT5
10	TMU_IN8	TMU_OUT6
11	TMU_IN9	TMU_OUT7
12	TMU_IN10	ADC0_EXT_TRIG0
13	TMU_IN11	ADC0_EXT_TRIG1
14	ACMP0_OUT	ADC0_EXT_TRIG2
15	ACMP1_OUT	ADC0_EXT_TRIG3
16	-	ADC1_EXT_TRIG0
17	pTMR_CH0	ADC1_EXT_TRIG1
18	pTMR_CH1	ADC1_EXT_TRIG2
19	pTMR_CH2	ADC1_EXT_TRIG3

Yuntu Microelectronics

Table 13.1 continued from previous page

Slot	Trigger input	Trigger output
20	pTMR_CH3	-
21	lpTMR0	-
22	eTMR0_INIT_TRIG	-
23	eTMR0_EXT_TRIG	-
24	eTMR1_INIT_TRIG	TMU_OUT8
25	eTMR1_EXT_TRIG	TMU_OUT9
26	eTMR2_INIT_TRIG	TMU_OUT10
27	eTMR2_EXT_TRIG	TMU_OUT11
28	eTMR3_INIT_TRIG	ACMP0_SAMPLE_INPUT
29	eTMR3_EXT_TRIG	ACMP1_SAMPLE_INPUT
30	ADC0_EOC	MPWM_TRIG0
31	ADC0_AWD	MPWM_TRIG1
32	ADC1_EOC	MPWM_TRIG2
33	ADC1_AWD	MPWM_TRIG3
34	PTU0_TRIG	MPWM_TRIG4
35	-	MPWM_TRIG5
36	PTU0_PULSE_OUT	MPWM_TRIG6
37	PTU1_TRIG	MPWM_TRIG7
38	-	-
39	PTU1_PULSE_OUT	-
40	-	eTMR0_HWTRIG0
41	-	eTMR0_FAULT0
42	-	eTMR0_FAULT1
43	RTC_ALARM	eTMR0_FAULT2
44	RTC_SECOND	eTMR1_HWTRIG1
45	-	eTMR1_FAULT0
46	-	eTMR1_FAULT1
47	-	eTMR1_FAULT2
48	-	eTMR2_HWTRIG2
49	-	eTMR2_FAULT0
50	-	eTMR2_FAULT1
51	-	eTMR2_FAULT2
52	-	eTMR3_HWTRIG3
53	-	eTMR3_FAULT0
54	-	eTMR3_FAULT1
55	I2C0_MASTER_TRIG	eTMR3_FAULT2
56	I2C0_SLAVE_TRIG	PTU0_TRIG_IN
57	-	-
58	-	-
59	SPI0_FRAME	-
60	SPI0_RX_DATA	PTU1_TRIG_IN
61	SPI1_FRAME	-
62	SPI1_RX_DATA	-
63	CIM_SW_TRIG	-
64	-	eTMR0_CH0_IN

Yuntu Microelectronics

Table 13.1 continued from previous page

Slot	Trigger input	Trigger output
65	-	eTMR0_CH1_IN
66	-	eTMR0_CH2_IN
67	I2C1_MASTER_TRIG	eTMR0_CH3_IN
68	I2C1_SLAVE_TRIG	eTMR1_CH0_IN
69	eTMR4_INIT_TRIG	eTMR1_CH1_IN
70	eTMR4_EXT_TRIG	eTMR1_CH2_IN
71	eTMR5_INIT_TRIG	eTMR1_CH3_IN
72	eTMR5_EXT_TRIG	eTMR2_CH0_IN
73	-	eTMR2_CH1_IN
74	-	eTMR2_CH2_IN
75	-	eTMR2_CH3_IN
76	-	LINFlexD0_TRIG
77	SPI2_FRAME	LINFlexD1_TRIG
78	SPI2_RX_DATA	LINFlexD2_TRIG
79	SPI3_FRAME	LINFlexD3_TRIG
80	SPI3_RX_DATA	LINFlexD4_TRIG
81	SPI4_FRAME	LINFlexD5_TRIG
82	SPI4_RX_DATA	LINFlexD6_TRIG
83	SPI5_FRAME	LINFlexD7_TRIG
84	SPI5_RX_DATA	I2C0_TRIG
85	I2C2_MASTER_TRIG	LINFlexD8_TRIG
86	I2C2_SLAVE_TRIG	LINFlexD9_TRIG
87	PTMR1_CH0	-
88	PTMR1_CH1	eTMR3_CH0_IN
89	PTMR1_CH2	eTMR3_CH1_IN
90	PTMR1_CH3	eTMR3_CH2_IN
91	PTMR2_CH0	eTMR3_CH3_IN
92	PTMR2_CH1	-
93	PTMR2_CH2	-
94	PTMR2_CH3	-
95	-	-
96	-	-
97	-	-
98	-	-
99	-	-
100	I2C3_MASTER_TRIG	lpTMR0_ALT0
101	I2C3_SLAVE_TRIG	CM7_RXEV
102	I2C4_MASTER_TRIG	-
103	I2C4_SLAVE_TRIG	-
104	SPI6_FRAME	-
105	SPI6_RX_DATA	-
106	SPI7_FRAME	-
107	SPI7_RX_DATA	-
108	-	I2C1_TRIG
109	-	I2C2_TRIG

Yuntu Microelectronics

Table 13.1 continued from previous page

Slot	Trigger input	Trigger output
110	CM7_TXEV	I2C3_TRIG
111	-	I2C4_TRIG
112	FMU_ERR_DO0	eTMR4_HWTRIG0
113	FMU_ERR_DO1	-
114	-	-
115	-	-
116	MPWM0_TRIG_OUT0_3	eTMR5_HWTRIG0
117	MPWM0_TRIG_OUT4_7	-
118	MPWM0_TRIG_OUT8_11	-
119	MPWM0_TRIG_OUT12_15	MPWM_TRIG8
120	MPWM1_TRIG_OUT0_3	MPWM_TRIG9
121	MPWM1_TRIG_OUT4_7	MPWM_TRIG10
122	MPWM1_TRIG_OUT8_11	MPWM_TRIG11
123	MPWM1_TRIG_OUT12_15	MPWM_TRIG12
124	MPWM2_TRIG_OUT0_3	MPWM_TRIG13
125	MPWM2_TRIG_OUT4_7	MPWM_TRIG14
126	MPWM2_TRIG_OUT8_11	MPWM_TRIG15
127	MPWM2_TRIG_OUT12_15	-

13.3 Register Definition

This subsection provides a description of the memory map and registers.

13.3.1 TMU Memory Map

TMU base address 40120000h

Table 13.2: TMU Memory Map

Offset	Register	Width	Access	Reset Value
0h	MUX (MUX0)	32	RW	00000000h
4h	MUX (MUX1)	32	RW	00000000h
8h	MUX (MUX2)	32	RW	00000000h
ch	MUX (MUX3)	32	RW	00000000h
10h	MUX (MUX4)	32	RW	00000000h
14h	MUX (MUX5)	32	RW	00000000h
18h	MUX (MUX6)	32	RW	00000000h
1ch	MUX (MUX7)	32	RW	00000000h
20h	MUX (MUX8)	32	RW	00000000h
24h	MUX (MUX9)	32	RW	00000000h
28h	MUX (MUX10)	32	RW	00000000h
2ch	MUX (MUX11)	32	RW	00000000h
30h	MUX (MUX12)	32	RW	00000000h
34h	MUX (MUX13)	32	RW	00000000h
38h	MUX (MUX14)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3ch	MUX (MUX15)	32	RW	00000000h
40h	MUX (MUX16)	32	RW	00000000h
44h	MUX (MUX17)	32	RW	00000000h
48h	MUX (MUX18)	32	RW	00000000h
4ch	MUX (MUX19)	32	RW	00000000h
50h	MUX (MUX20)	32	RW	00000000h
54h	MUX (MUX21)	32	RW	00000000h
58h	MUX (MUX22)	32	RW	00000000h
5ch	MUX (MUX23)	32	RW	00000000h
60h	MUX (MUX24)	32	RW	00000000h
64h	MUX (MUX25)	32	RW	00000000h
68h	MUX (MUX26)	32	RW	00000000h
6ch	MUX (MUX27)	32	RW	00000000h
70h	MUX (MUX28)	32	RW	00000000h
74h	MUX (MUX29)	32	RW	00000000h
78h	MUX (MUX30)	32	RW	00000000h
7ch	MUX (MUX31)	32	RW	00000000h

13.3.1.1 TMU MUX Register

Offset: 0h + n * 4h

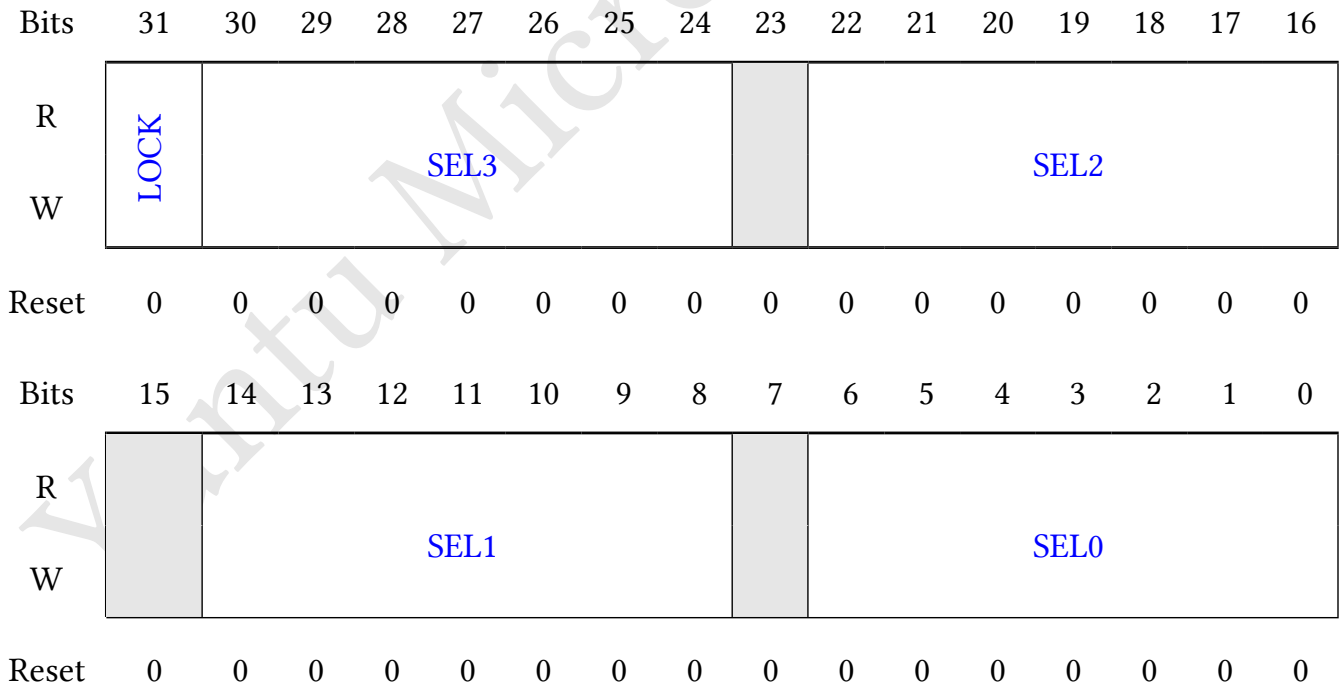


Table 13.3: TMU MUX Register Description

Field	Function
31	**Trigger MUX Locked**

Yuntu Microelectronics

Field	Function
LOCK	Lock the MUX select for peripheral trigger input
30 - 24 SEL3	**Trigger MUX Input Source Select 3** This read/write bit field is used to configure the MUX select for peripheral trigger input.
22 - 16 SEL2	**Trigger MUX Input Source Select 2** This read/write bit field is used to configure the MUX select for peripheral trigger input.
14 - 8 SEL1	**Trigger MUX Input Source Select 1** This read/write bit field is used to configure the MUX select for peripheral trigger input.
6 - 0 SEL0	**Trigger MUX Input Source Select 0** This read/write bit field is used to configure the MUX select for peripheral trigger input.

13.4 Glossary

TMU Trigger Multiplexer

Yuntu Microelectronics

Part V

Clock, Reset and Power Management

Yuntu Microelectronics

Chapter 14

System Clock Unit (SCU)

14.1 Introduction

The system clock unit(SCU) provides multiple clock source choices for the MCU. The module contains Fast Internal RC Oscillator(FIRC), Slow Internal RC Oscillator(SIRC), Fast Crystal Oscillator(FXOSC), Slow Crystal Oscillator(SXOSC) and a Phase-Locked Loop (PLL).

- The FIRC, SIRC, FXOSC and PLL can be selected as system clock source.
- Either FIRC or FXOSC can be selected as PLL reference clock.
- The SIRC is always on unless it is forced to be disabled in Deepsleep, Standby or Powerdown mode.
- The SXOSC is not used as system clock source, it provides accurate clock to some peripherals.
- For safety reason, the system clock unit(SCU) contains clock monitor block.
- Multiple clock sources can be selected to output to chip IO.

14.1.1 Features

- Fast Internal RC Oscillator(FIRC)
  - Up to 96 MHz
  - Default system boot clock source
  - Support trim for temperature and process
  - Can be selected as PLL reference clock
- Slow Internal RC Oscillator(SIRC)
  - Up to 12 MHz
  - Can be selected as system clock source
  - Always on unless it is forced to be disable in Deepsleep, Standby and Powerdown mode
  - Support trim for temperature and process
- Fast Crystal Oscillator(FXOSC)
  - Support 4~40MHz crystal
  - Can be selected as PLL reference clock
  - Can be selected as system clock source
  - Support bypass mode
- Slow Crystal Oscillator(SXOSC)
  - 32.768 KHz real time oscillator
  - Can't be selected as system clock
  - Provides accurate clock as the functional clock sources of some peripherals and Real-Time Clock(RTC)

Yuntu Microelectronics

- Phase-Locked Loop(PLL)
  - Up to 200 MHz
  - Contains Voltage-Controlled Oscillator(VCO)
  - Support selectable reference clock
  - Contain frequency lock detector
  - Can be selected as system clock source
- Clock Monitor Unit(CMU)
  - SCU contains 4 CMU blocks
  - CMU monitors slow bus clock, FIRC clock, PLL clock and FXOSC clock
  - FXOSC or SIRC clock can be selected as reference clock of CMU
  - CMU can detect frequency out of range, loss of checked clock and loss of reference clock
- SCU provides glitch free switcher to select system clock source
- SCU provides system clock dividers to generate core clock, fast bus clock and slow bus clock

14.1.2 Block Diagram

The following figure shows the SCU block diagram.

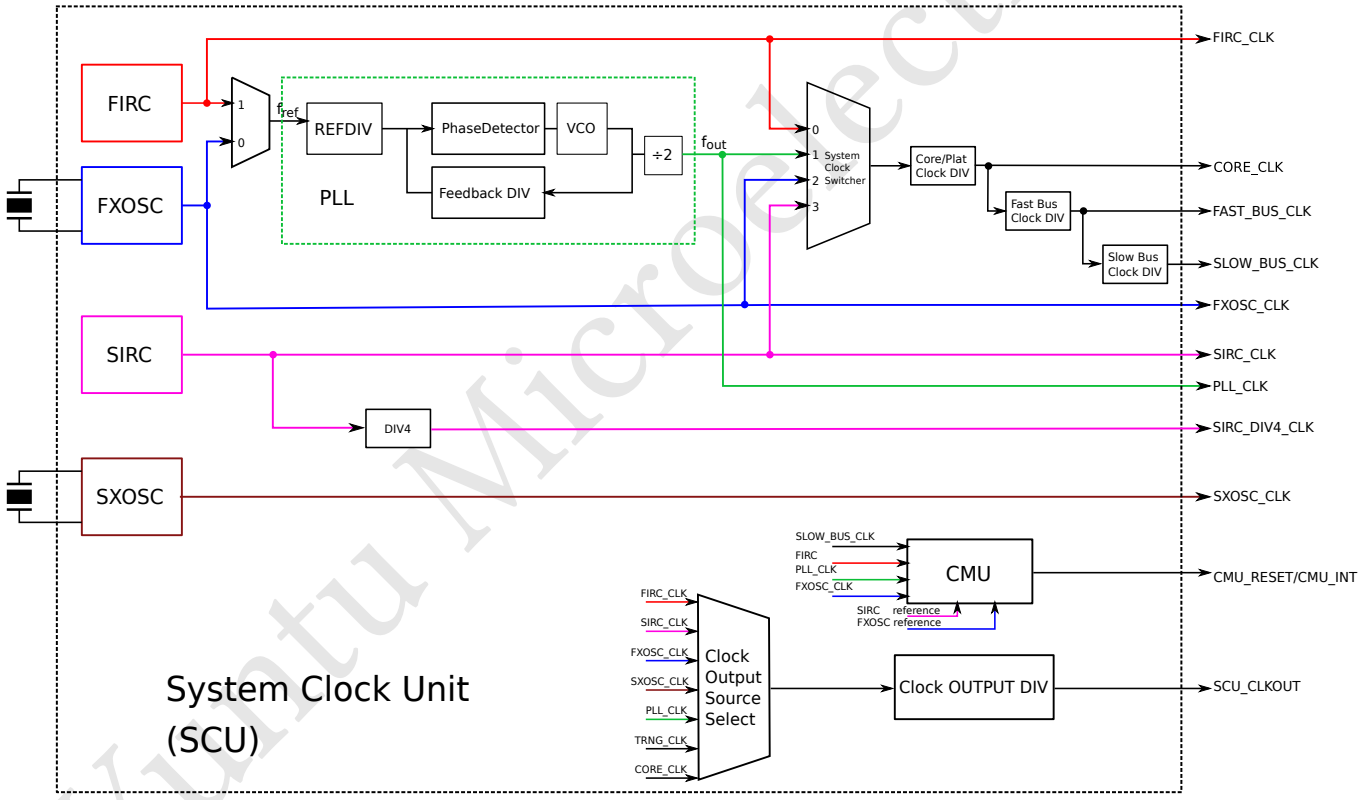


Figure 14.1: SCU Block Diagram

14.2 Register Definition

14.2.1 SCU Memory Map

SCU base address 401f0000h

Yuntu Microelectronics

Table 14.1: SCU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Clock Selection Register (CLKS)	32	RW	00000000h
4h	System Clock Divider (DIV)	32	RW	00010001h
8h	Status Register (STS)	32	RO	00000004h
ch	Clock Divier Status Register (DIVSTS)	32	RO	00010001h
10h	FIRC Control Register (FIRC_CTRL)	32	RW	00000000h
14h	SIRC Control Register (SIRC_CTRL)	32	RW	0000000Fh
18h	FXOSC Control Register (FXOSC_CTRL)	32	RW	00001810h
1ch	SXOSC Control Register (SXOSC_CTRL)	32	RW	00000000h
20h	PLL Control Register (PLL_CTRL)	32	RW	00000000h
24h	PLL Configuration Register (PLL_CFG)	32	RW	00000000h
28h	PLL Spread Spectrum Mode Register (PLL_FM)	32	RW	00000000h
2ch	PLL Fractional Mode Register (PLL_FD)	32	RW	00000000h
30h	Clock Output Control Register (CLKO)	32	RW	00000000h
40h	Clock Monitor Unit Control Register (CMU_CTRL)	32	RW	00000000h
44h	Clock Monitor Unit Status Register (CMUSTS)	32	RW	00000000h
48h	Clock Monitor Unit High Comparison Value Register (CMUCMP0_HIGH)	32	RW	0000FFFFh
4ch	Clock Monitor Unit Low Comparison Value Register (CMUCMP0_LOW)	32	RW	00000000h
50h	Clock Monitor Unit High Comparison Value Register (CMUCMP1_HIGH)	32	RW	0000FFFFh
54h	Clock Monitor Unit Low Comparison Value Register (CMUCMP1_LOW)	32	RW	00000000h
58h	Clock Monitor Unit High Comparison Value Register (CMUCMP2_HIGH)	32	RW	0000FFFFh
5ch	Clock Monitor Unit Low Comparison Value Register (CMUCMP2_LOW)	32	RW	00000000h
60h	Clock Monitor Unit High Comparison Value Register (CMUCMP3_HIGH)	32	RW	0000FFFFh
64h	Clock Monitor Unit Low Comparison Value Register (CMUCMP3_LOW)	32	RW	00000000h

Yuntu Microelectronics

14.2.1.1 SCU CLKS Register

The register contains system clock source selection bit, the system clock source is the root of system clock divider which generates Core/Platform Clock, Fast Bus Clock and Slow Bus Clock.

Offset: 0h

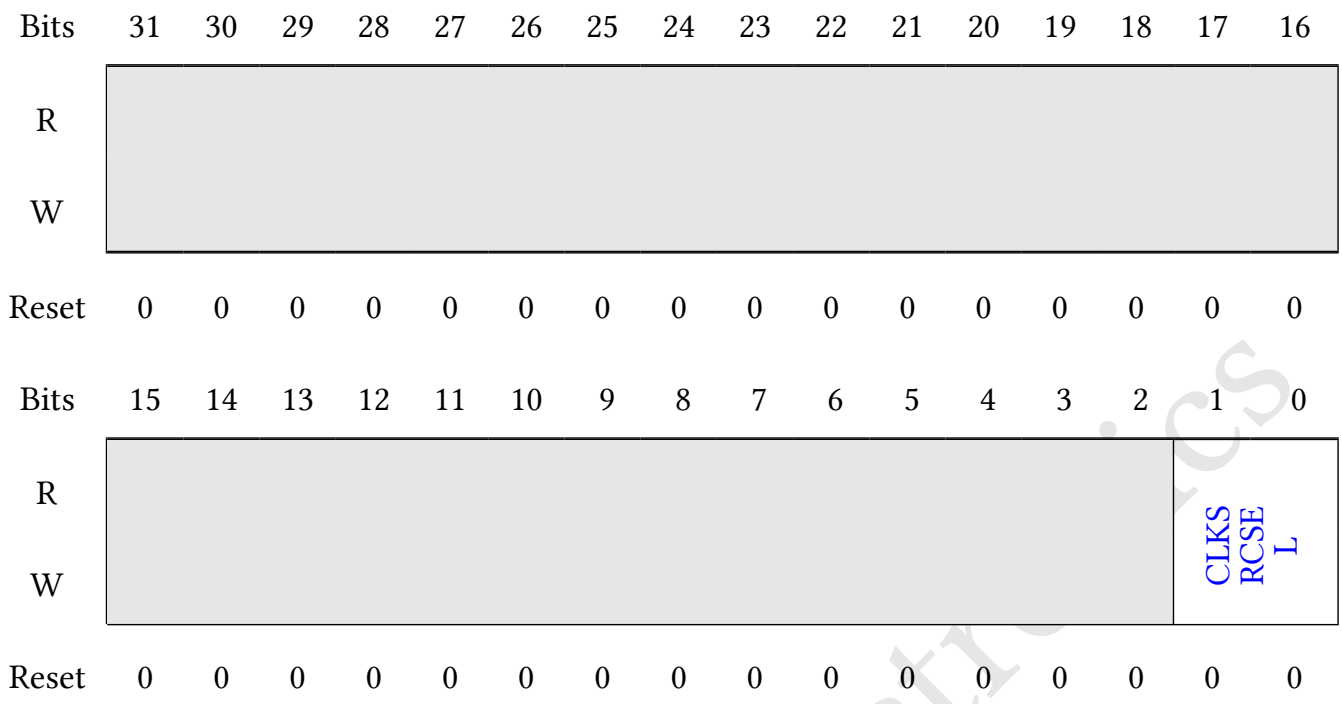


Table 14.2: SCU CLKS Register Description

Field	Function
1 - 0 CLKSRCSEL	**System Clock Source Selection** Select System Clock Source. 00b - Select FIRC as system clock source 01b - Select PLL as system clock source 10b - Select FXOSC as system clock source 11b - Select SIRC as system clock source

Yuntu Microelectronics

14.2.1.2 SCU DIV Register

This register controls the system clock divider which generates Core Platform Clock, Fast Bus Clock and Slow Bus Clock.

Offset: 4h

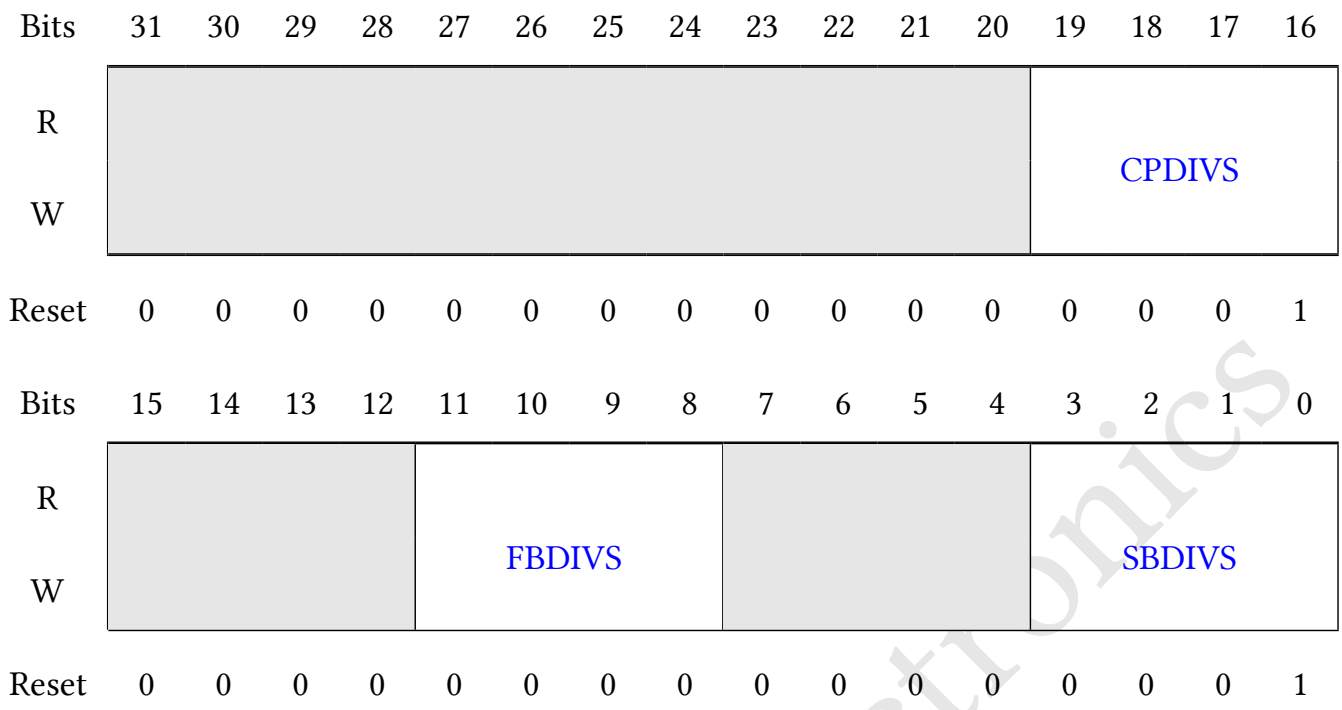


Table 14.3: SCU DIV Register Description

Field	Function
19 - 16 CPDIVS	**Core Platform Clock Divider Value** 0000b - Divide by 1 0001b - Divide by 2 0010b - Divide by 3 0011b - Divide by 4 0100b - Divide by 5 0101b - Divide by 6 0110b - Divide by 7 0111b - Divide by 8 1000b - Divide by 9 1001b - Divide by 10 1010b - Divide by 11 1011b - Divide by 12 1100b - Divide by 13 1101b - Divide by 14 1110b - Divide by 15 1111b - Divide by 16
11 - 8	**Fast Bus Clock Divider Value**

Yuntu Microelectronics

Field	Function
FBDIVS	0000b - Divide by 1 0001b - Divide by 2 0010b - Divide by 3 0011b - Divide by 4 0100b - Divide by 5 0101b - Divide by 6 0110b - Divide by 7 0111b - Divide by 8 1000b - Divide by 9 1001b - Divide by 10 1010b - Divide by 11 1011b - Divide by 12 1100b - Divide by 13 1101b - Divide by 14 1110b - Divide by 15 1111b - Divide by 16
3 - 0 SBDIVS	**Slow Bus Clock Divider Value** 0000b - Divide by 1 0001b - Divide by 2 0010b - Divide by 3 0011b - Divide by 4 0100b - Divide by 5 0101b - Divide by 6 0110b - Divide by 7 0111b - Divide by 8 1000b - Divide by 9 1001b - Divide by 10 1010b - Divide by 11 1011b - Divide by 12 1100b - Divide by 13 1101b - Divide by 14 1110b - Divide by 15 1111b - Divide by 16

Yuntu Microelectronics

14.2.1.3 SCU STS Register

System Clock Unit Status register

Offset: 8h

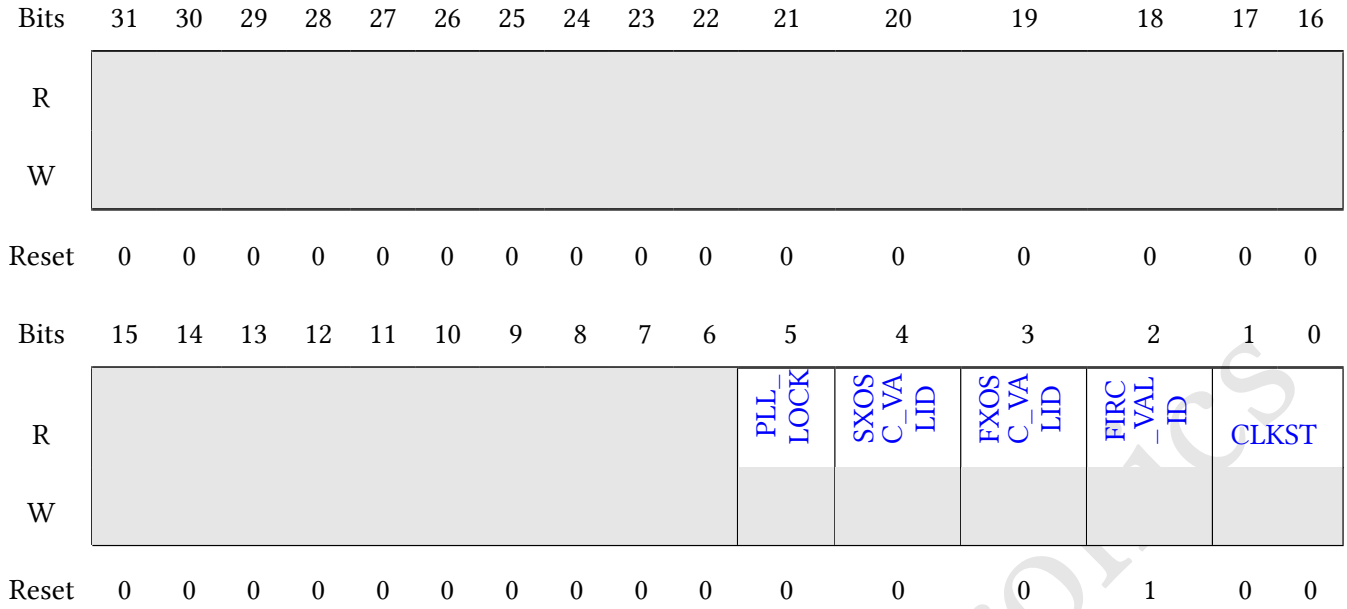


Table 14.4: SCU STS Register Description

Field	Function
5 PLL_LOCK	**PLL Lock Bit** 0b - PLL unlocked 1b - PLL locked
4 SXOSC_VALID	**SXOSC Clock Valid** 0b - SXOSC Clock invalid 1b - SXOSC Clock valid
3 FXOSC_VALID	**FXOSC Clock Valid** 0b - FXOSC Clock invalid 1b - FXOSC Clock valid
2 FIRC_VALID	**FIRC Clock Valid** 0b - FIRC Clock invalid 1b - FIRC Clock valid
1 - 0 CLKST	**System Clock Source Selection Status Bits** 00b - FIRC is selected as system clock source 01b - PLL is selected as system clock source 10b - FXOSC is selected as system clock source 11b - SIRC is selected as system clock source

14.2.1.4 SCU DIVSTS Register

System Clock Unit clock divider status, check this register after any change of divider value.

Yuntu Microelectronics

Offset: ch

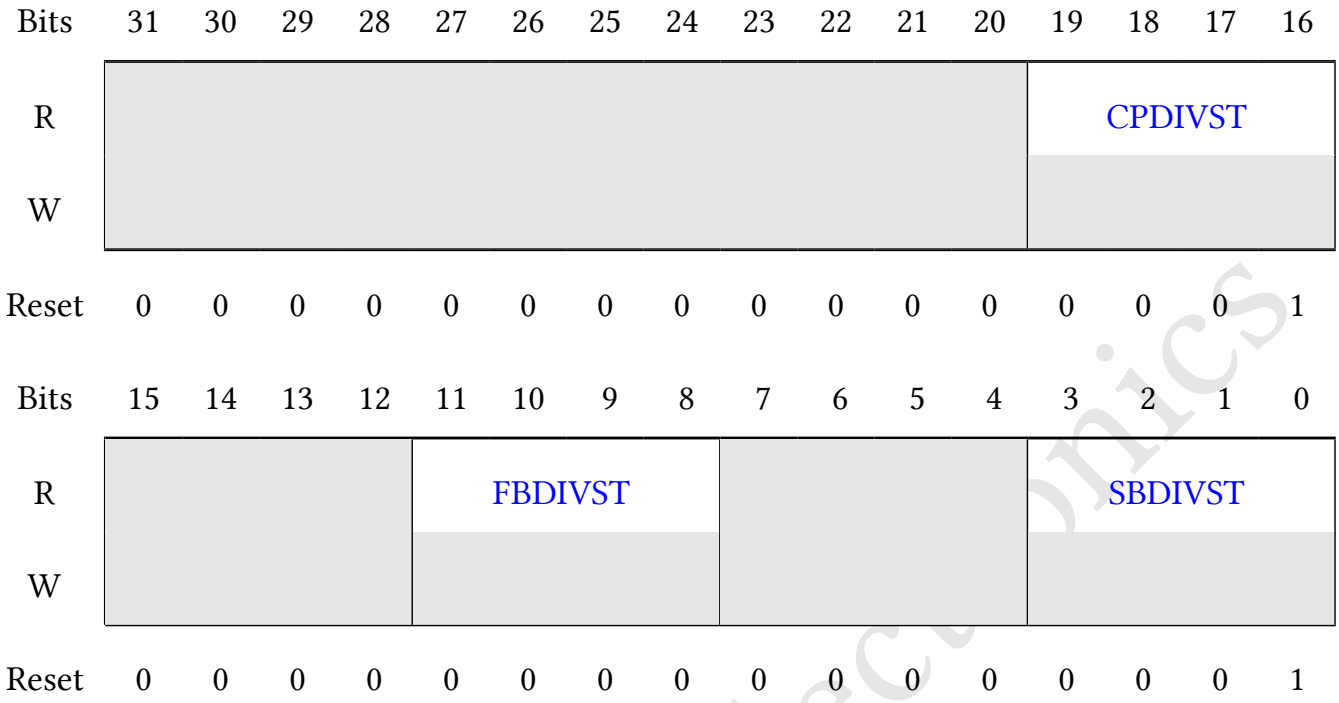


Table 14.5: SCU DIVSTS Register Description

Field	Function
19 - 16 CPDIVST	**Core Platform Clock Divider Value** 0000b - Divide by 1 0001b - Divide by 2 0010b - Divide by 3 0011b - Divide by 4 0100b - Divide by 5 0101b - Divide by 6 0110b - Divide by 7 0111b - Divide by 8 1000b - Divide by 9 1001b - Divide by 10 1010b - Divide by 11 1011b - Divide by 12 1100b - Divide by 13 1101b - Divide by 14 1110b - Divide by 15 1111b - Divide by 16
11 - 8	**Fast Bus Clock Divider Value**

Yuntu Microelectronics

Field	Function
FBDIVST	0000b - Divide by 1 0001b - Divide by 2 0010b - Divide by 3 0011b - Divide by 4 0100b - Divide by 5 0101b - Divide by 6 0110b - Divide by 7 0111b - Divide by 8 1000b - Divide by 9 1001b - Divide by 10 1010b - Divide by 11 1011b - Divide by 12 1100b - Divide by 13 1101b - Divide by 14 1110b - Divide by 15 1111b - Divide by 16
3 - 0 SBDIVST	**Slow Bus Clock Divider Value** 0000b - Divide by 1 0001b - Divide by 2 0010b - Divide by 3 0011b - Divide by 4 0100b - Divide by 5 0101b - Divide by 6 0110b - Divide by 7 0111b - Divide by 8 1000b - Divide by 9 1001b - Divide by 10 1010b - Divide by 11 1011b - Divide by 12 1100b - Divide by 13 1101b - Divide by 14 1110b - Divide by 15 1111b - Divide by 16

Yuntu Microelectronics

14.2.1.5 SCU FIRC_CTRL Register

FIRC control register

Offset: 10h

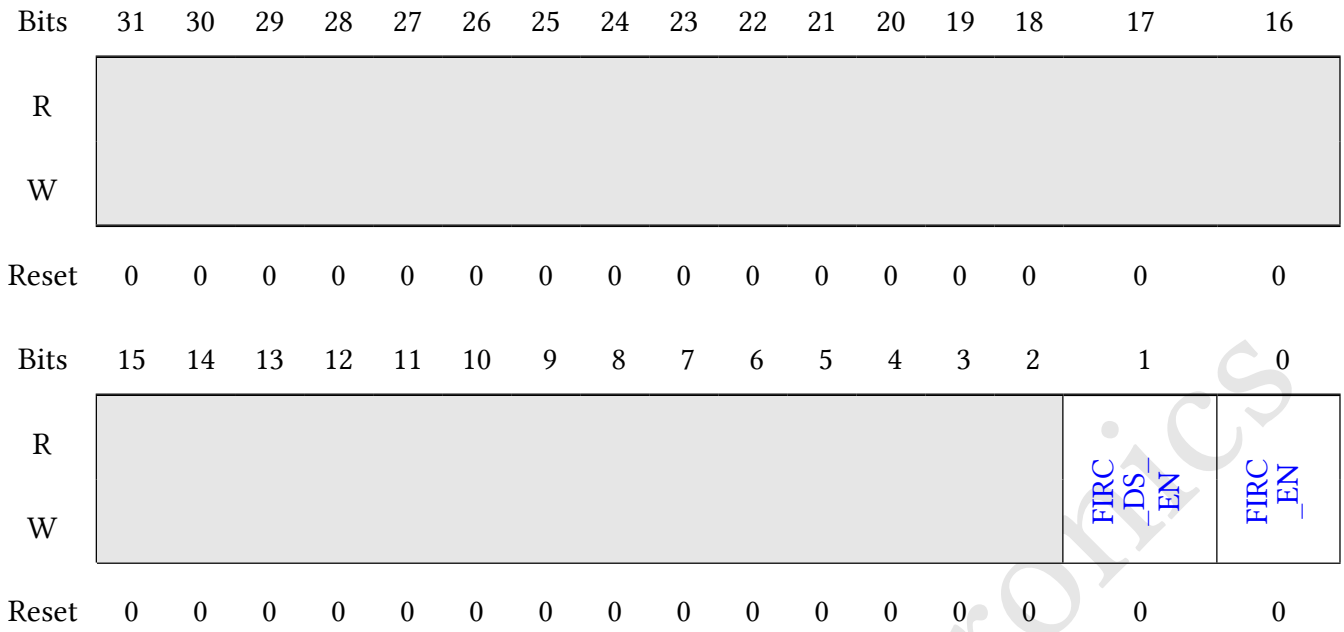


Table 14.6: SCU FIRC_CTRL Register Description

Field	Function
1 FIRC_DS_EN	**FIRC Deepsleep Enable** Enable/Disable FIRC when chip enters Deepsleep mode. 0b - Disable FIRC when chip enters Deepsleep mode 1b - Enable FIRC when chip enters Deepsleep mode
0 FIRC_EN	**FIRC Enable** Enable/Disable FIRC regardless it is selected as system clock source.. 0b - Disable FIRC if it is not selected as system clock source 1b - Enable FIRC regardless it is selected as system clock or not

Yuntu Microelectronics

14.2.1.6 SCU SIRC_CTRL Register

SIRC control register

Offset: 14h

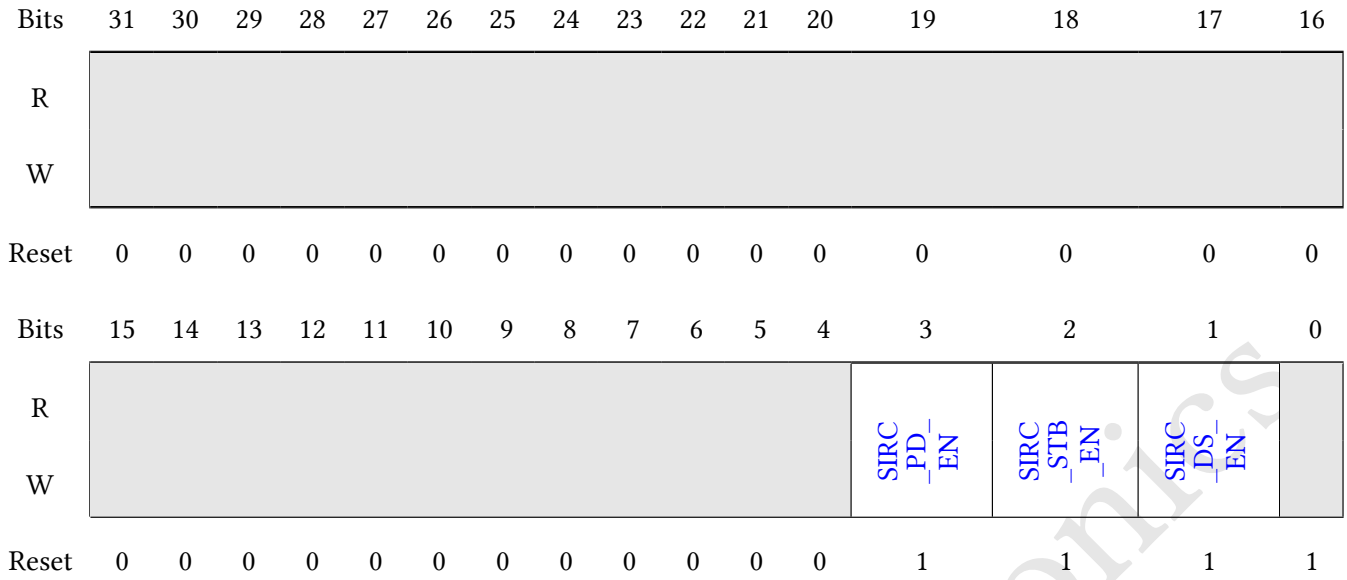


Table 14.7: SCU SIRC_CTRL Register Description

Field	Function
3 SIRC_PD_ EN	**SIRC Powerdown Enable** Enable/Disable SIRC when chip enters Powerdown mode. 0b - Disable SIRC when chip enters Powerdown mode 1b - Enable SIRC when chip enters Powerdown mode
2 SIRC_STB _EN	**SIRC Standby Enable** Enable/Disable SIRC when chip enters Standby mode. 0b - Disable SIRC when chip enters Standby mode 1b - Enable SIRC when chip enters Standby mode
1 SIRC_DS_ EN	**SIRC Deepsleep Enable** Enable/Disable SIRC when chip enters Deepsleep mode. 0b - Disable SIRC when chip enters Deepsleep mode 1b - Enable SIRC when chip enters Deepsleep mode

Yuntu Microelectronics

14.2.1.7 SCU FXOSC_CTRL Register

FXOSC control register

Offset: 18h

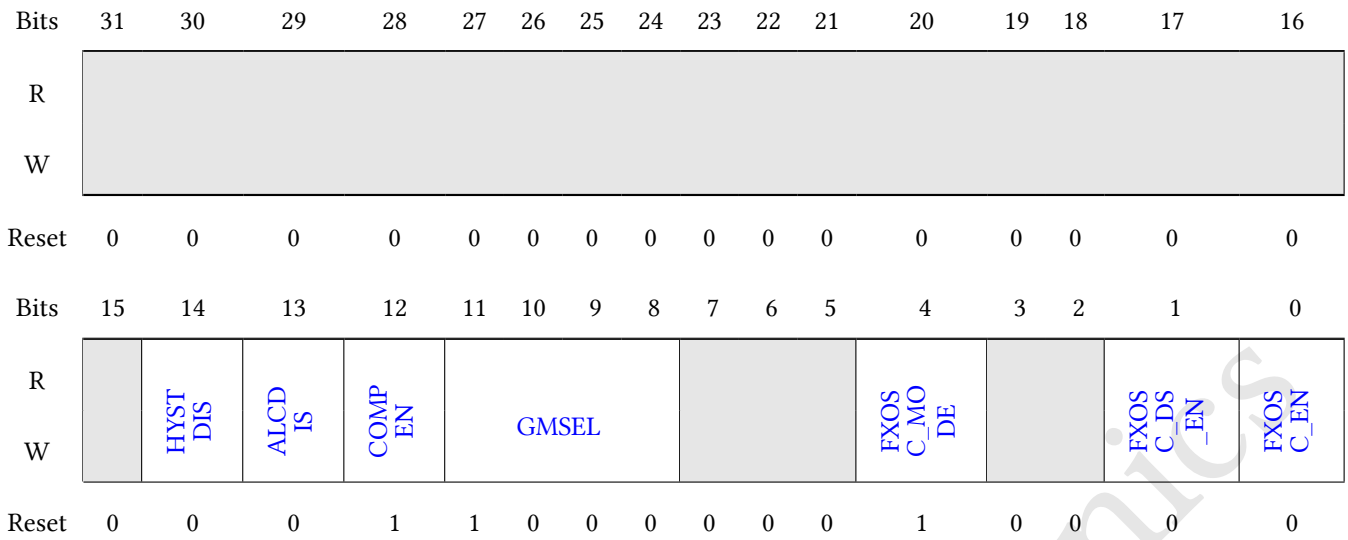


Table 14.8: SCU FXOSC_CTRL Register Description

Field	Function
14 HYSTDIS	**Hys Disable Bit** This bit should not be changed when FXOSC_VALID is high
13 ALCDIS	**ALC Disable Bit** This bit should not be changed when FXOSC_VALID is high
12 COMPEN	**Comp Enable Bit** Comp enable bit, this bit should be high when crystal mode is used, and it should be low when bypass mode is used, and it should not be changed when FXOSC_VALID is high
11 - 8	**Gain Select Bit**

Yuntu Microelectronics

Field	Function
GMSEL	Gain select bit, higher gain value results higher power and better performance, it only impact FXOSC crystal mode, and it should not be changed when FXOSC_VALID is high 0000b - Reserved 0001b - used for 4MHz oscillator 0010b - used for 4MHz-6MHz oscillator 0011b - used for 4MHz-8MHz oscillator 0100b - used for 4MHz-8MHz oscillator 0101b - used for 4MHz-8MHz oscillator 0110b - used for 4MHz-8MHz oscillator 0111b - used for 6MHz-12MHz oscillator 1000b - used for 6MHz-12MHz oscillator 1001b - used for 12MHz-20MHz oscillator 1010b - used for 12MHz-32MHz oscillator 1011b - used for 12MHz-32MHz oscillator 1100b - used for 12MHz-32MHz oscillator 1101b - used for 32MHz-40MHz oscillator 1110b - used for 32MHz-40MHz oscillator 1111b - used for 32MHz-40MHz oscillator NOTE: The above data represents recommended values for typical applications of FXOSC. For more specific details, please refer to the Datasheet, calculate the required Gain Setting, and perform crystal oscillator impedance matching tests.
4 FXOSC_MODE	**FXOSC Mode Selection** Select FXOSC crystal mode or bypass mode, it should not be changed when FXOSC_VALID is high 0b - FXOSC bypass mode 1b - FXOSC crystal mode
1 FXOSC_DS_EN	**FXOSC Deepsleep Enable** Enable/Disable FXOSC when chip enters Deepsleep mode. 0b - Disable FXOSC when chip enters Deepsleep mode 1b - Enable FXOSC when chip enters Deepsleep mode
0 FXOSC_EN	**Fast Crystal Oscillator(FXOSC) Enable** Enable/Disable Fast Crystal Oscillator(FXOSC) regardless it is selected as system clock source.. 0b - Disable FXOSC if it is not selected as system clock source 1b - Enable FXOSC regardless it is selected as system clock or not

14.2.1.8 SCU SXOSC_CTRL Register

SXOSC control register

Offset: 1ch

Yuntu Microelectronics

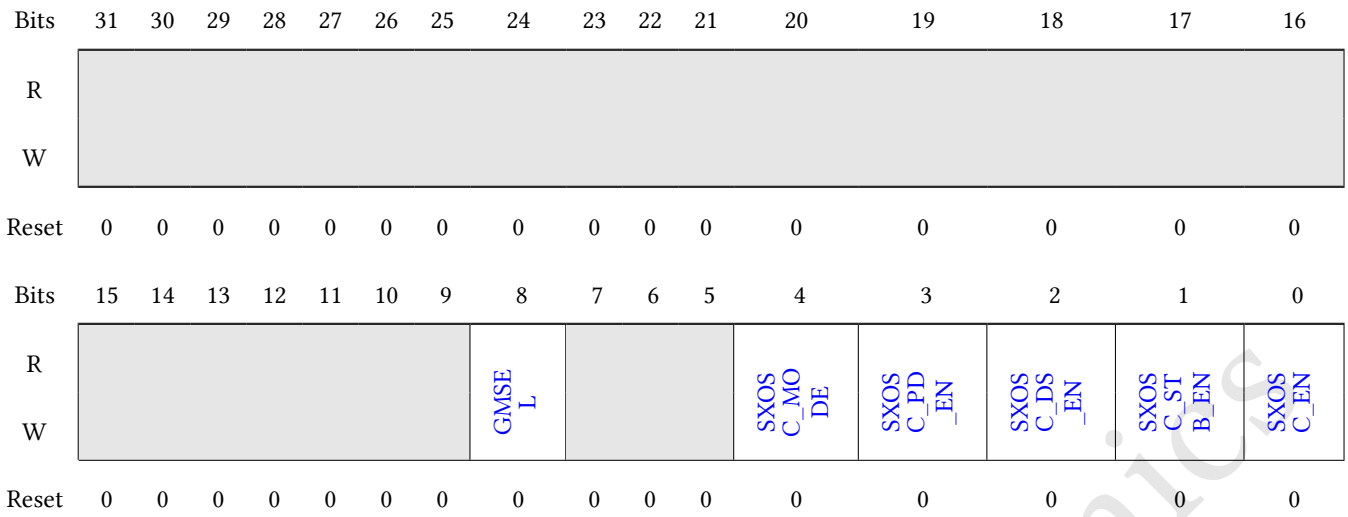


Table 14.9: SCU SXOSC_CTRL Register Description

Field	Function
8 GMSEL	**SXOSC Gain Select Bit** SXOSC Gain select bit
4 SXOSC_MODE	**SXOSC Mode Selection** Select SXOSC oscillator mode or bypass mode 0b - SXOSC bypass mode 1b - SXOSC crystal mode
3 SXOSC_PD_EN	**SXOSC Powerdown Enable** Enable/Disable SXOSC when chip enters Powerdown mode. 0b - Disable SXOSC when chip enters Standby mode 1b - Enable SXOSC when chip enters Standby mode
2 SXOSC_DS_EN	**SXOSC Deepsleep Enable** Enable/Disable SXOSC when chip enters Deepsleep mode. 0b - Disable SXOSC when chip enters Deepsleep mode 1b - Enable SXOSC when chip enters Deepsleep mode
1 SXOSC_STB_EN	**SXOSC Standby Enable** Enable/Disable SXOSC when chip enters Standby mode. 0b - Disable SXOSC when chip enters Standby mode 1b - Enable SXOSC when chip enters Standby mode
0 SXOSC_EN	**SXOSC Enable** Enable/Disable SXOSC 0b - Disable SXOSC 1b - Enable SXOSC

Yuntu Microelectronics

14.2.1.9 SCU PLL_CTRL Register

System Clock Unit PLL control register, changes to this register should be done when PLL is disabled to prevent glitches to system clock source.

Offset: 20h

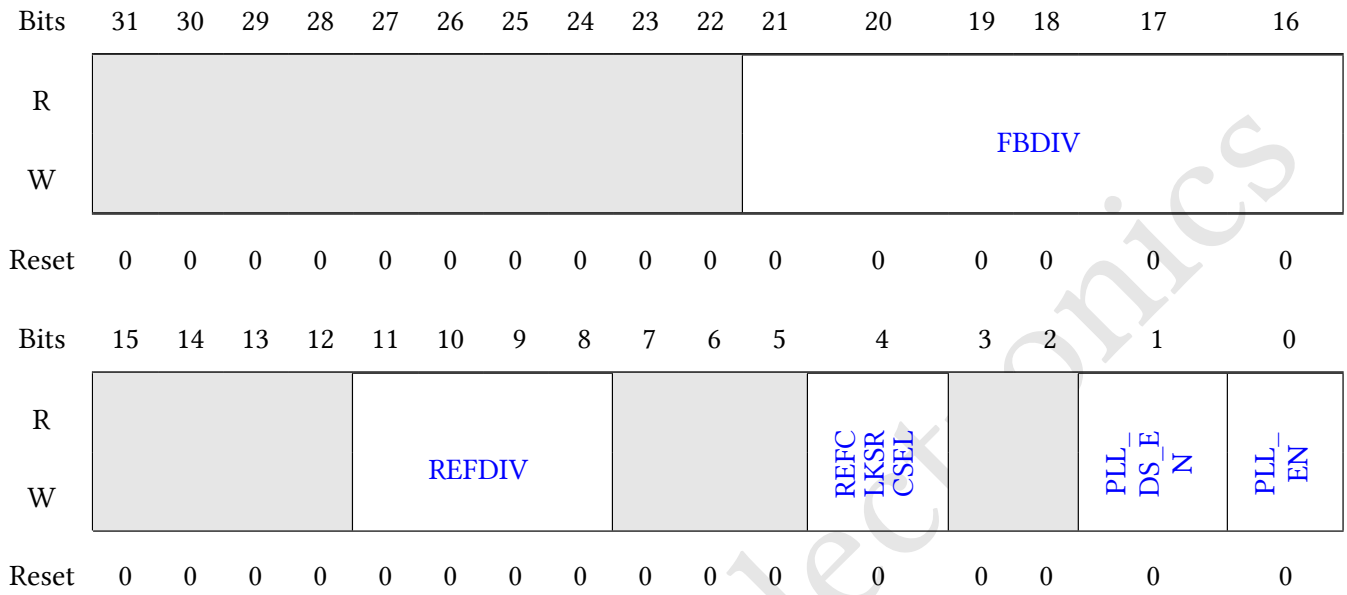


Table 14.10: SCU PLL_CTRL Register Description

Field	Function
21 - 16 FBDIV	**Feedback Divider Value** The FBDIV bits establish the value of the divider in th PLL feedabck loop. Please refer to Phase-Locked Loop Clock(PLL).
11 - 8 REFDIV	**Reference Clock Divider Value** Selects the amount to divide down the reference clock for the PLL. Please refer to Phase-Locked Loop Clock(PLL).
4 REFCLKSRCSEL	**Reference Clock Selection** PLL reference clock source select bit 0b - Select FXOSC as PLL reference clock 1b - Select FIRC as PLL reference clock
1 PLL_DS_EN	**PLL Deepsleep Enable** Enable/Disable PLL when chip enters Deepsleep mode. 0b - Disable PLL when chip enters Deepsleep mode 1b - Enable PLL when chip enters Deepsleep mode
0	**PLL Enable**

Yuntu Microelectronics

Field	Function
PLL_EN	Enable/Disable PLL regardless it is selected as system clock source. 0b - Disable PLL if it is not selected as system clock source 1b - Enable PLL regardless it is selected as system clock or not

14.2.1.10 SCU PLL_CFG Register

Offset: 24h

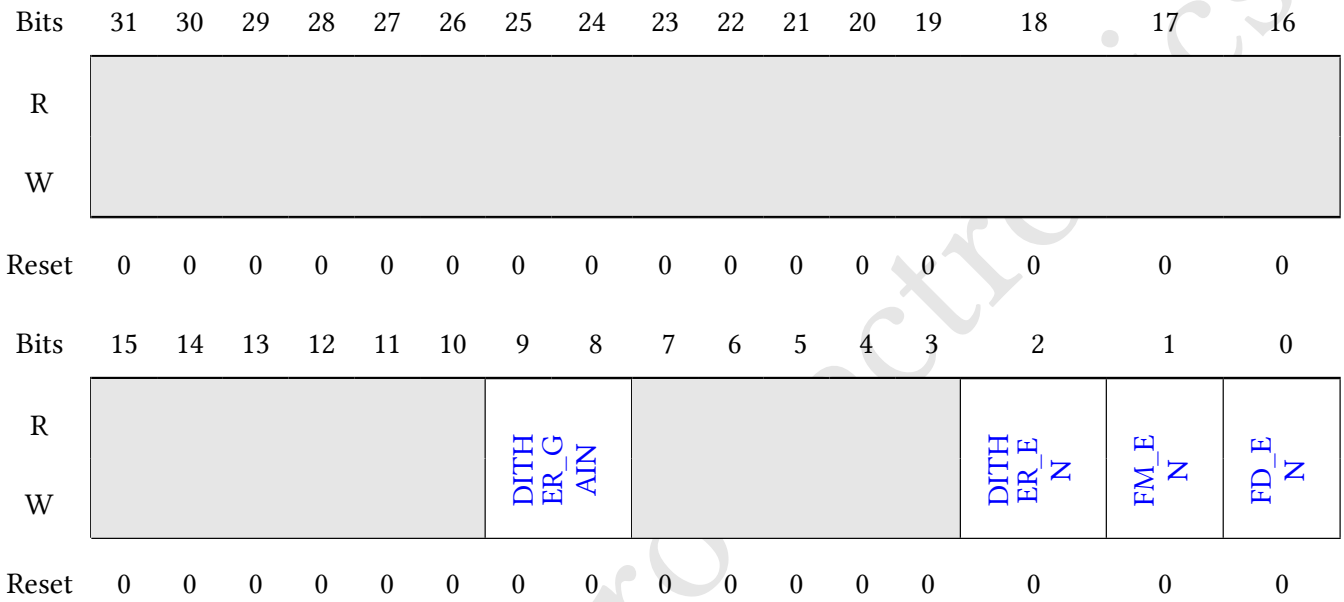


Table 14.11: SCU PLL_CFG Register Description

Field	Function
9 - 8 DITHER_GAIN	**Dither Gain** Dither gain control bit. Replace one bit of the accumulator output SUM_SUB[3:0] with the output D0 of the LFSR. 00b - SUM_SUB[0] is replaced by D0 01b - SUM_SUB[1] is replaced by D0 10b - SUM_SUB[2] is replaced by D0 11b - SUM_SUB[3] is replaced by D0
2 DITHER_EN	**Dither Enable** Enable/Disable dither 0b - Disabled 1b - Enabled
1 FM_EN	**Spread Spectrum Mode Enable** Enable/Disable spread spectrum mode. 0b - Disabled 1b - Enabled

Yuntu Microelectronics

Field	Function
0 FD_EN	**Fractional Mode Enable** Enable/Disable Fractional mode. 0b - Disabled 1b - Enabled

14.2.1.11 SCU PLL_FM Register

Offset: 28h

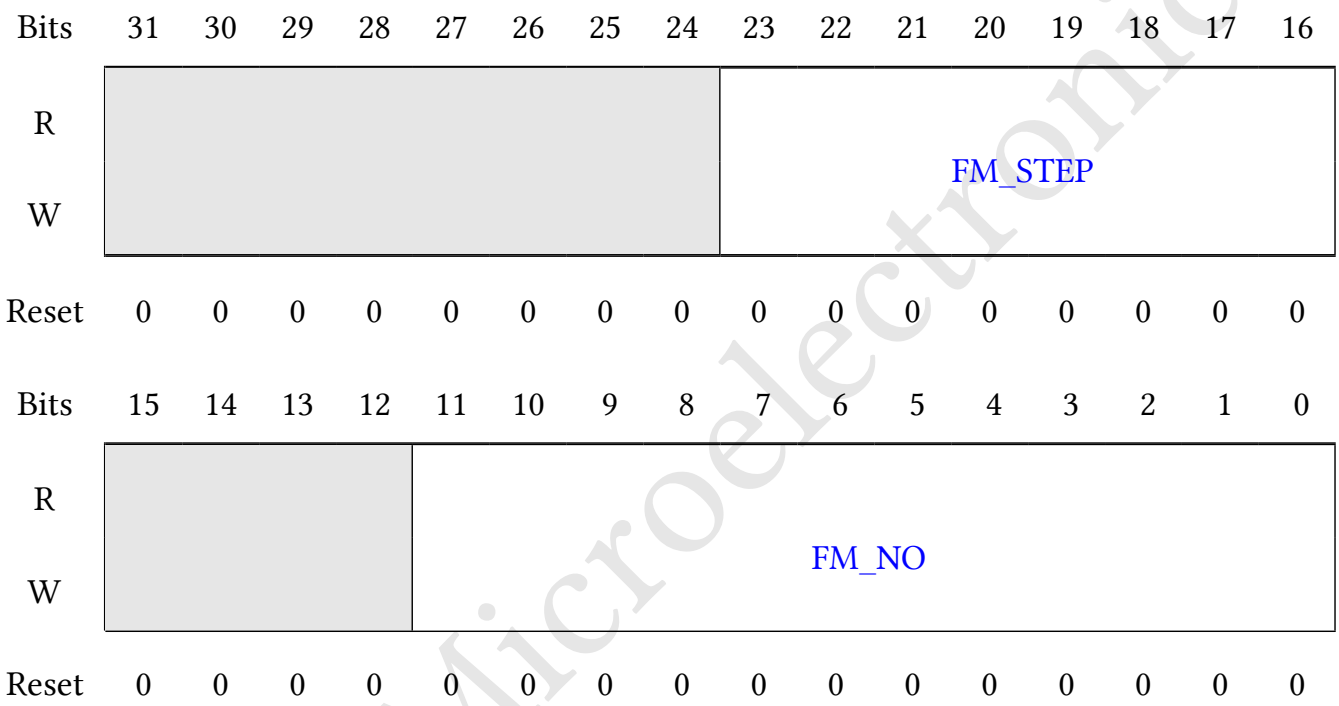


Table 14.12: SCU PLL_FM Register Description

Field	Function
23 - 16 FM_STEP	**Frequency Modulation Step Size** In Frequency Modulation mode, FM_STEP provides the step size for modulation depth and frequency.
11 - 0 FM_NO	**Frequency Modulation Number of Steps** In Frequency Modulation mode, FM_NO provides the number of steps to achieve modulation depth.

14.2.1.12 SCU PLL_FD Register

Offset: 2ch

Yuntu Microelectronics

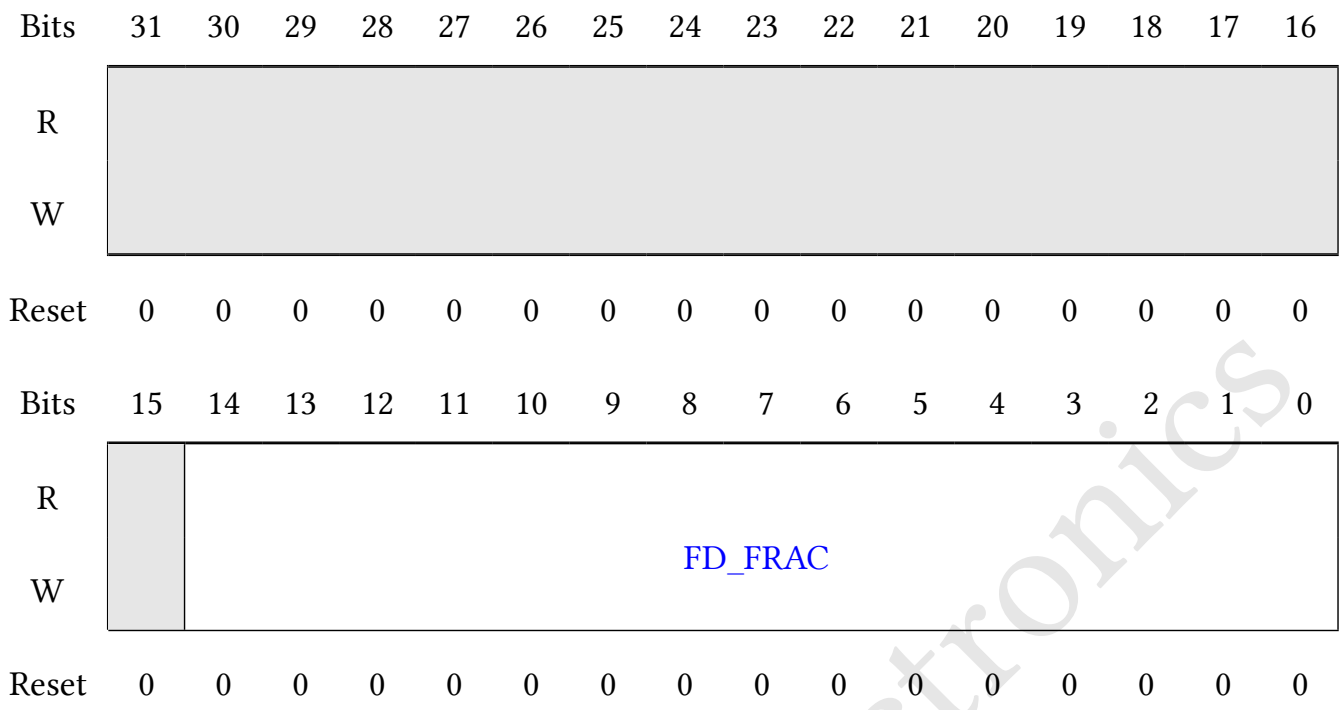


Table 14.13: SCU PLL_FD Register Description

Field	Function
14 - 0 FD_FRAC	**Numerator of Fractional Loop Division Factor** The value of this field must be less than 32768. In integer mode, it must be set to 0.

Yuntu Microelectronics

14.2.1.13 SCU CLKO Register

This register is to select the clock sources to output to chip IOs for external use.

Offset: 30h

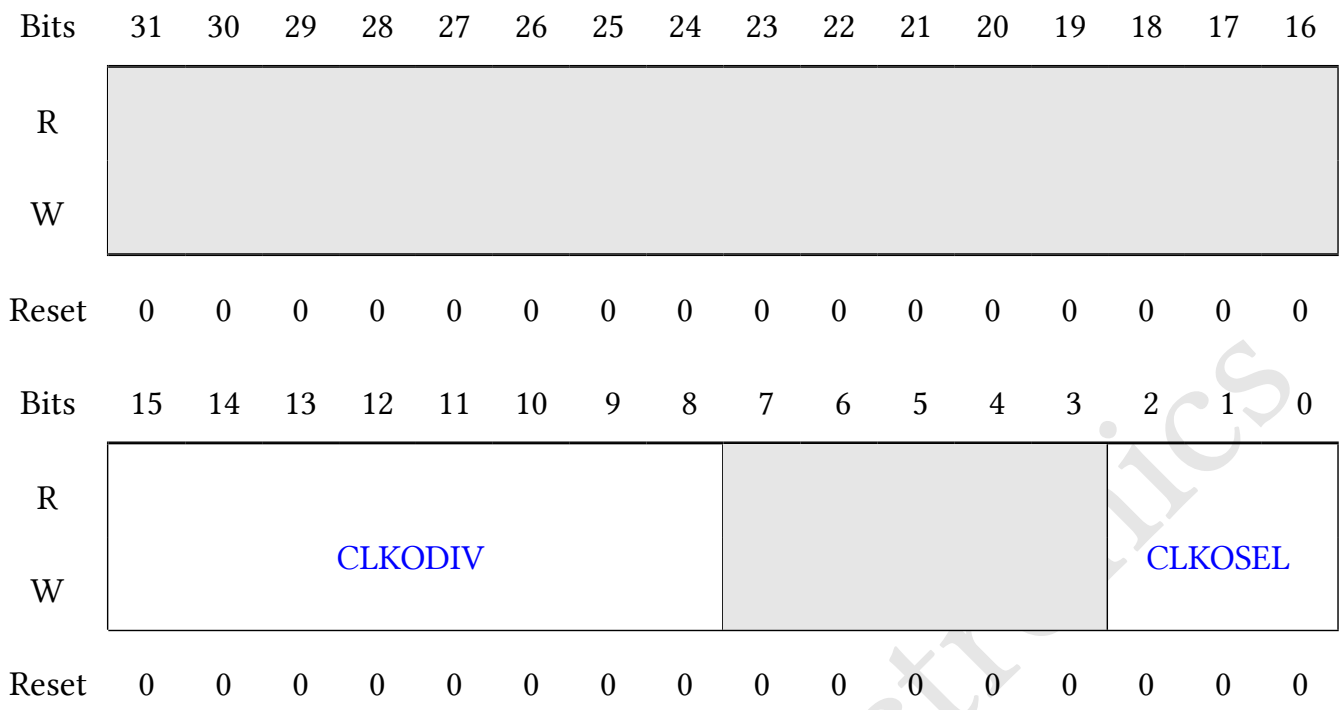


Table 14.14: SCU CLKO Register Description

Field	Function
15 - 8 CLKODIV	**Clock Output Divider** 00000000b - DIV disable 00000001b - DIV disable xxxxxx1xb - DIV as the divider value
2 - 0 CLKOSEL	**Clock Output Source Select** 000b - None clock select to output 001b - FIRC clock output 010b - SIRC clock output 011b - FXOSC clock output 100b - SXOSC clock output 101b - PLL clock output 110b - TRNG ring oscillator otuput 111b - Core clock output

Yuntu Microelectronics

14.2.1.14 SCU CMU_CTRL Register

System Clock Unit Clock Monitor Unit control register

Offset: 40h

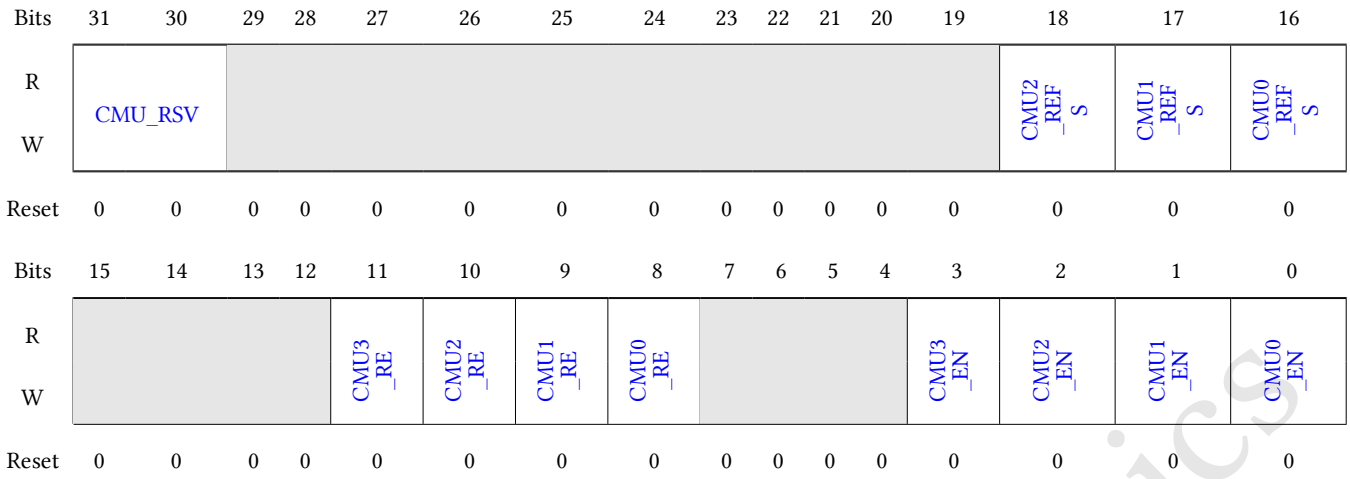


Table 14.15: SCU CMU_CTRL Register Description

Field	Function
31 - 30 CMU_RSV	**Reserved register bits for future internal use only.** The register bits are reseted by global reset.
18 CMU2_REFS	**CMU2 Reference Clock Selected** Select reference clock of CMU2 0b - SIRC selected as reference clock of CMU 1b - FXOSC selected as reference clock of CMU
17 CMU1_REFS	**CMU1 Reference Clock Selected** Select reference clock of CMU1 0b - SIRC selected as reference clock of CMU 1b - FXOSC selected as reference clock of CMU
16 CMU0_REFS	**CMU0 Reference Clock Selected** Select reference clock of CMU0 0b - SIRC selected as reference clock of CMU 1b - FXOSC selected as reference clock of CMU
11 CMU3_RE	**CMU3 Reset Enable** Enable/Disable reset when CMU3 event happens 0b - Reset disable when CMU event happens 1b - Reset enable when CMU event happens
10 CMU2_RE	**CMU2 Reset Enable** Enable/Disable reset when CMU2 event happens 0b - Reset disable when CMU event happens 1b - Reset enable when CMU event happens
9	**CMU1 Reset Enable**

Yuntu Microelectronics

Field	Function
CMU1_RE	Enable/Disable reset when CMU1 event happens 0b - Reset disable when CMU event happens 1b - Reset enable when CMU event happens
8 CMU0_RE	**CMU0 Reset Enable** Enable/Disable reset when CMU0 event happens 0b - Reset disable when CMU event happens 1b - Reset enable when CMU event happens
3 CMU3_EN	**CMU3 Enable** Enable/Disable CMU3 0b - CMU disable 1b - CMU enable
2 CMU2_EN	**CMU2 Enable** Enable/Disable CMU2 0b - CMU disable 1b - CMU enable
1 CMU1_EN	**CMU1 Enable** Enable/Disable CMU1 0b - CMU disable 1b - CMU enable
0 CMU0_EN	**CMU0 Enable** Enable/Disable CMU0 0b - CMU disable 1b - CMU enable

Yuntu Microelectronics

14.2.1.15 SCU CMUSTS Register

System Clock Unit Clock Monitor Unit status register

Offset: 44h

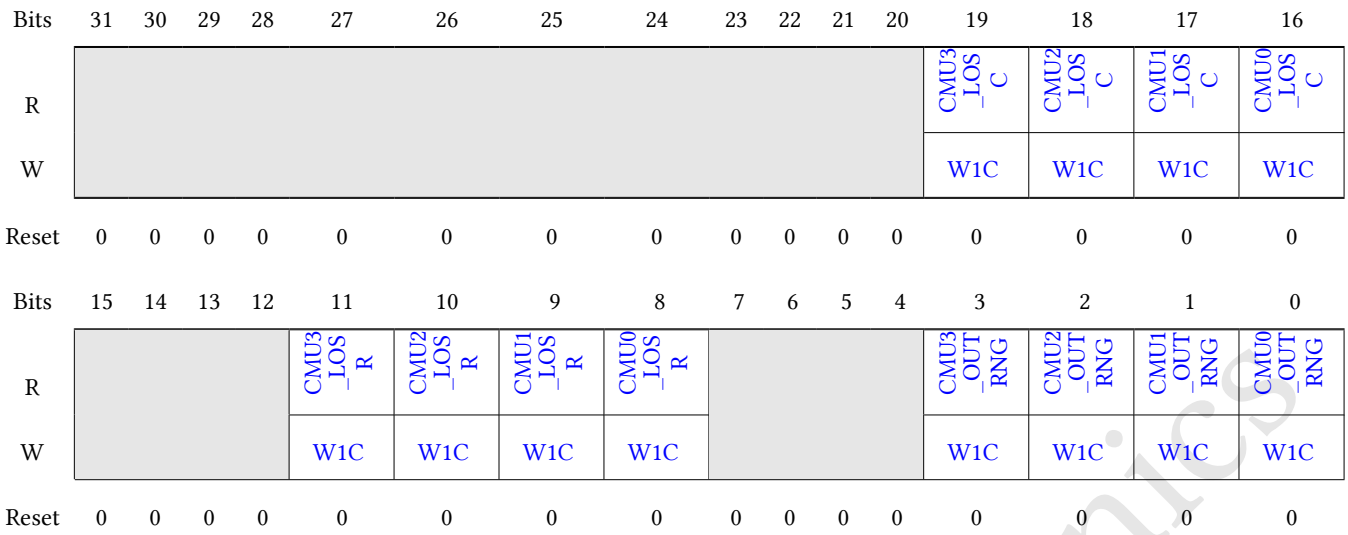


Table 14.16: SCU CMUSTS Register Description

Field	Function
19 CMU3_LOSC	**CMU3 Loss of Clock** Indicates CMU3 loss of clock event 0b - Checked Clock of CMU3 is available 1b - Checked Clock of CMU3 is lost
18 CMU2_LOSC	**CMU2 Loss of Clock** Indicates CMU2 loss of clock event 0b - Checked Clock of CMU2 is available 1b - Checked Clock of CMU2 is lost
17 CMU1_LOSC	**CMU1 Loss of Clock** Indicates CMU1 loss of clock event 0b - Checked Clock of CMU1 is available 1b - Checked Clock of CMU1 is lost
16 CMU0_LOSC	**CMU0 Loss of Clock** Indicates CMU0 loss of clock event 0b - Checked Clock of CMU0 is available 1b - Checked Clock of CMU0 is lost
11 CMU3_LOSR	**CMU3 Loss of Reference Clock** Indicates CMU3 loss of reference clock event 0b - Reference Clock of CMU3 is available 1b - Reference Clock of CMU3 is lost
10	**CMU2 Loss of Reference Clock**

Yuntu Microelectronics

Field	Function
CMU2_LOSR	Indicates CMU2 loss of reference clock event 0b - Reference Clock of CMU2 is available 1b - Reference Clock of CMU2 is lost
9 CMU1_LOSR	**CMU1 Loss of Reference Clock** Indicates CMU1 loss of reference clock event 0b - Reference Clock of CMU1 is available 1b - Reference Clock of CMU1 is lost
8 CMU0_LOSR	**CMU0 Loss of Reference Clock** Indicates CMU0 loss of reference clock event 0b - Reference Clock of CMU0 is available 1b - Reference Clock of CMU0 is lost
3 CMU3_OUTRNG	**CMU3 Clock Out of Range** Indicates Checked Clock of CMU3 is out of range 0b - Checked Clock of CMU3 is out of range 1b - Reference Clock of CMU3 is lost
2 CMU2_OUTRNG	**CMU2 Clock Out of Range** Indicates Checked Clock of CMU2 is out of range 0b - Checked Clock of CMU2 is out of range 1b - Reference Clock of CMU2 is lost
1 CMU1_OUTRNG	**CMU1 Clock Out of Range** Indicates Checked Clock of CMU1 is out of range 0b - Checked Clock of CMU1 is out of range 1b - Reference Clock of CMU1 is lost
0 CMU0_OUTRNG	**CMU0 Lock Out of Range** Indicates Checked Clock of CMU0 is out of range 0b - Checked Clock of CMU0 is out of range 1b - Reference Clock of CMU0 is lost

Yuntu Microelectronics

14.2.1.16 SCU CMUCMP_HIGH Register

System Clock Unit Clock Monitor High Comparison Value Register

Offset: 48h + n * 8h

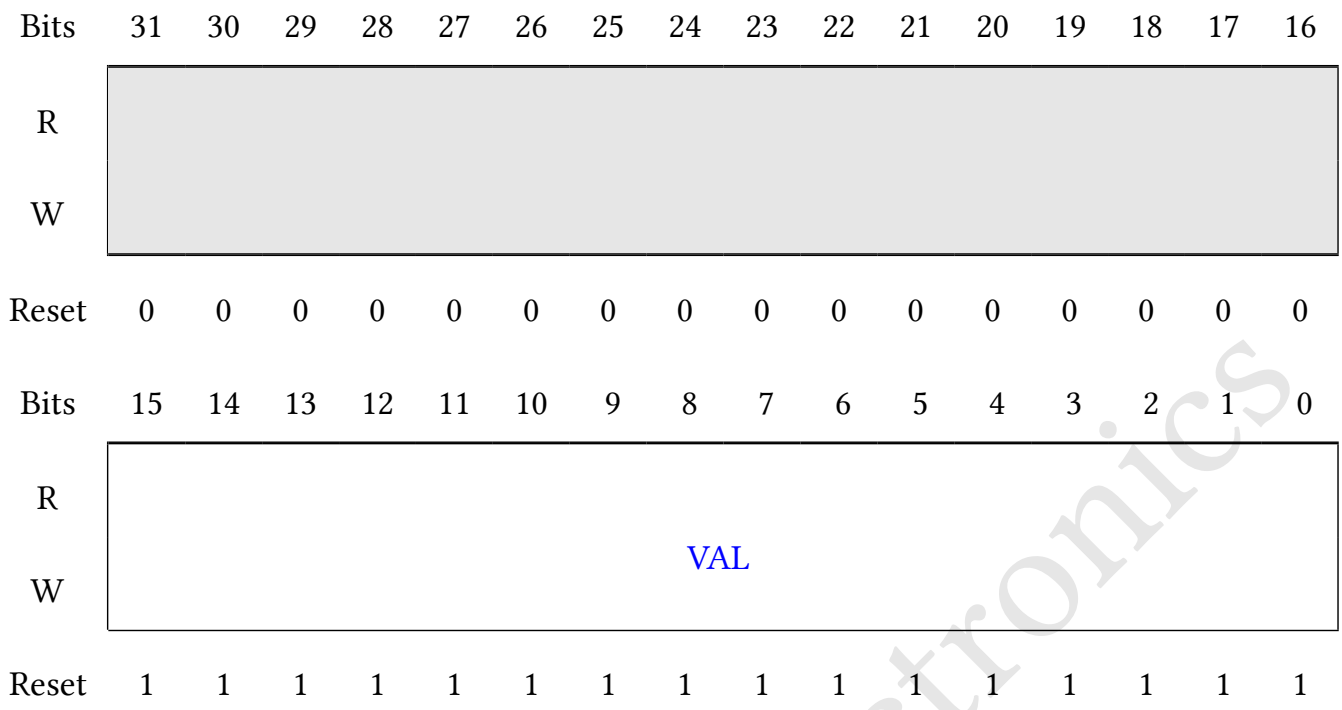


Table 14.17: SCU CMUCMP_HIGH Register Description

Field	Function
15 - 0 VAL	**High Comparison Value**

Yuntu Microelectronics

14.2.1.17 SCU CMUCMP_LOW Register

System Clock Unit Clock Monitor Low Comparison Value Register

Offset: 4ch + n * 8h

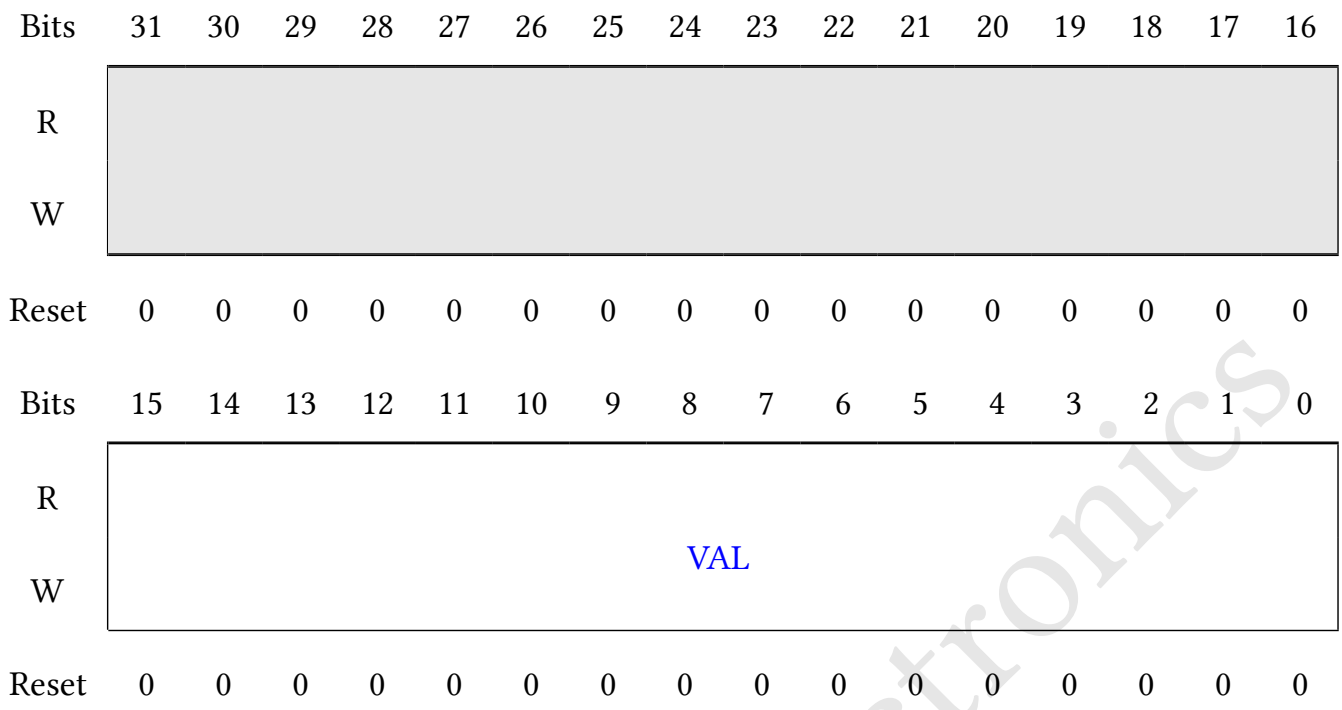


Table 14.18: SCU CMUCMP_LOW Register Description

Field	Function
15 - 0 VAL	**Low Comparison Value**

14.3 Functional Description

14.3.1 Clock Switcher

The system clock unit(SCU) contains a glitch free clock switcher. User can switch the clock source when last clock switching is completed. User can configure the SCU CLKS[CLKSRCSEL] bit to change the system clock source, but the actual switch to the newly selected clock only happens when SCU STS[CLKST] change to be same as SCU CLKS[CLKSRCSEL]. If the newly selected clock source is not available, the previous clock source will remain selected and SCU STS[CLKST] will not change.

14.3.2 Phase-Locked Loop Clock(PLL)

The SCU PLL Clock is available depending on the device's configuration of the SCU module. When PLL clock is enabled by setting PLL_CTRL[PLL_EN] and STS[PLL_LOCK] will be set.

PLL reference clock can be selected from FIRC or FXOSC by configuring PLL_CTRL register REFCLKSRCSEL bit. The reference should be pre-divided before sending to VCO by PLL_CTRL[REFDIV]. The PLL output clock is divided before feedback to VCO by PLL_CTRL[FBDIV].

$$f_{out} = \frac{f_{ref}}{2 * (REFDIV + 1)} * (FBDIV + 1)$$

Yuntu Microelectronics

Please disable PLL before changing any PLL configuration bits.

PLL supports integer mode, fractional mode and spread spectrum mode.

14.3.2.1 Integer Mode

When PLL_CFG[FD_EN]=0 and PLL_CFG[FM_EN]=0, PLL works in integer mode.

$$f_{vco} = f_{ref} * (REFDIV + 1)$$

14.3.2.2 Fractional Mode

When PLL_CFG[FD_EN]=1 and PLL_CFG[FM_EN]=0, PLL works in fractional mode.

$$f_{vco} = f_{ref} * (REFDIV + 1 + FD_FRAC/2^15)$$

14.3.2.3 Spread Spectrum Mode

When PLL_CFG[FD_EN]=1 and PLL_CFG[FM_EN]=0, PLL works in spread spectrum mode. The PLL only support downward spread spectrum mode.

The modulation frequency:

$$f_m = f_{ref}/(2 * FM_{NO})$$

If the f_m range supports [10KHz,100KHz], then the FM_NO range is [200,2000].

The modulation depth:

$$MD = (1 - FD_FRAC/2^15)f_{ref}/f_{vco}$$

The range of MD is [-f_{ref}/f_{vco},0]

$$FM_{step} = (2^15 - FD_FRAC)/FM_{NO}$$

$$FM_{step_max} = 32768/200=164

Notes: When PLL operates in decimal mode and spread spectrum mode, f_{ref} should be set to the maximum value, that is, f_{ref}=40MHz.

14.3.3 System Clock Divider

The system clock unit(SCU) contains 3 dividers to generate core, fast bus and slow bus clock from system clock source selected by clock switcher. The core clock is divided from system clock source directly and fast bus clock is divided from core clock, the slow bus clock is divided from fast bus clock. The 3 dividers support divide-value from 1 to 16 by configure DIV[CPDIVS], DIV[FBDIVS] and DIV[SBDIVS]. When clock dividers work with newly divider values, the DIVSTS[CPDIVST], DIVSTS[FBDIVST] and DIVSTS[SBDIVST] reflect the divide ratios.

The setting of clock divider and system clock source selected by clock switcher should ensure that the max frequency of core/platform clock should be lower than 200MHz, the max frequency of fast bus clock should be lower than 100MHz, and the max frequency of slow bus clock should be lower than 50MHz.

Yuntu Microelectronics

14.3.4 Clock Monitor Unit (CMU)

The SCU CMU block provides functions to monitor the abnormalities in the monitored clocks(checked clocks) and reference clocks, including loss of clock of checked clock, loss of clock of reference clock and out of frequency of checked clock. CMU supports to select SIRC or FXOSC as reference clock.

CMU is used to monitor various checked clocks

- CMU0: check Slow Bus Clock
- CMU1: check FIRC Clock
- CMU2: check PLL Clock
- CMU3: use SIRC Clock as reference clock(fixed) to check FXOSC Clock.

14.3.5 Clocks After Reset

When out of any of Reset(including Power on and Powerdown wakeup), the FIRC and SIRC are enabled. PLL and FXOSC are disabled.

SXOSC is disabled by Reset and Power on Reset. It will keep active if SXOSC is enabled in Powerdown mode when waking up from Powerdown mode.

14.3.6 Operation Modes

- Active: SCU is full function.
- Sleep: SCU is full function.
- Deepsleep: Core clock, plat clock, fast bus clock and slow bus clock are gated, but all clock sources(FIRC, SIRC, FXOSC, SXOSC, and PLL) can be configured active.
- Standby: Only SIRC and SXOSC can be configured as active.
- Powerdown: Only SIRC and SXOSC can be configured as active.

14.4 Glossary

FIRC	Fast Internal RC Oscillator
SIRC	Slow Internal RC Oscillator
FXOSC	Fast Crystal Oscillator
SXOSC	Slow Crystal Oscillator
PLL	Phase-locked Loop
VCO	Voltage-controlled oscillator
CMU	Clock Monitor Unit

Yuntu Microelectronics

Chapter 15

IP Control (IPC)

15.1 Introduction

The IP Control (IPC) module provides clock control and configuration for on-chip peripherals. Each peripheral has its clock control and configuration register.

The peripheral's operational (bus interface) clock must be enabled in the IPC register before the peripheral can be used.

A peripheral may also require an asynchronous (functional) clock, so the IPC also provides some clock inputs options for a peripheral which are software selectable.

Any peripheral can choose its asynchronous clock by setting the Clock Source Select (SRCSEL) bits.

Optionally, any peripheral can also have a clock divider for its asynchronous clock, selected by setting the Clock Divider (DIV) bits.

The Software Reset (SWREN) is available for each peripheral to generate its own reset signal.

15.2 Features

- Peripheral Bus clock enable
- IPC clock source selection as follow
  - FIRC 96MHz
  - SIRC_DIV4_CLK 3MHz
  - FXOSC 4~40MHz
  - SXOSC 32.768KHz
  - FAST_BUS_CLK
  - PLL 200MHz
- IPC clock divide values from 1 to 16
- Peripheral software reset

15.3 Block diagram

Below is a block diagram of the IPC module:

Yuntu Microelectronics

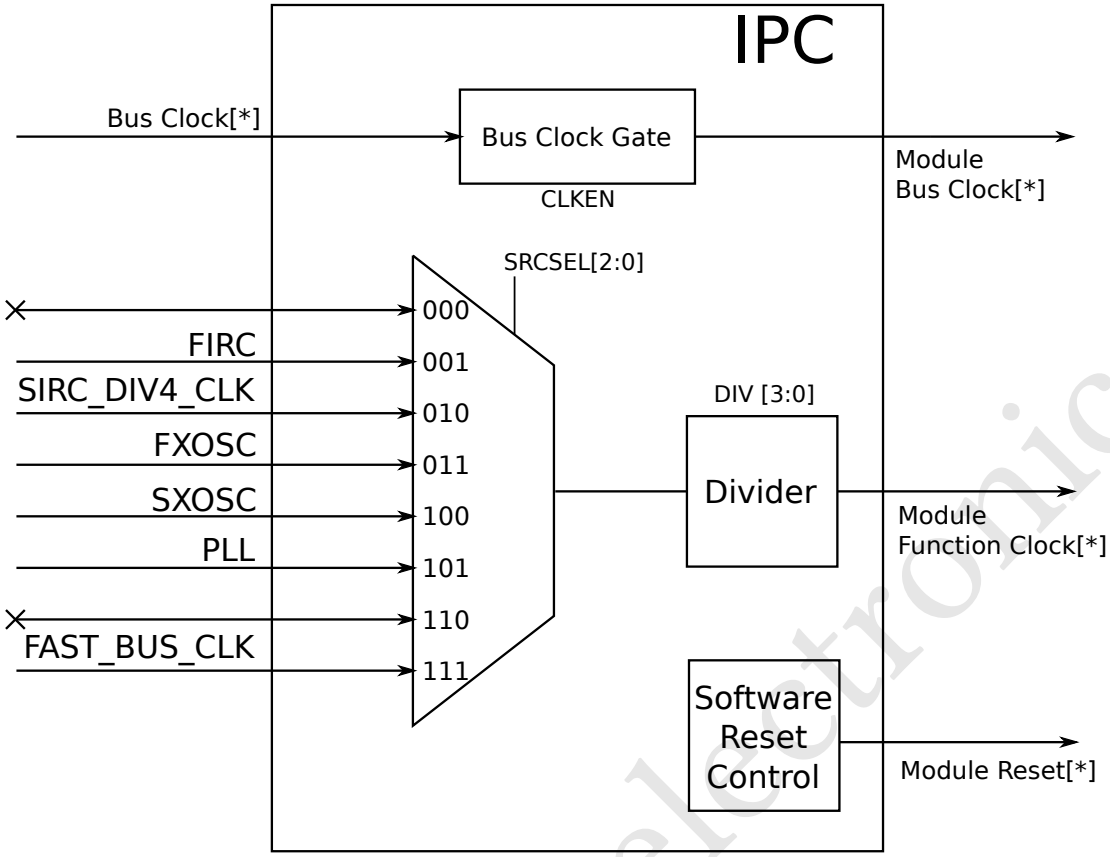


Figure 15.1: IPC Block Diagram

15.4 Peripheral Clock Table

Table 15.1: IP Clock Control Table

No.	Module Name	Offset	Bus Clock	Bus Clock Enable	Peripheral Functional Clock	Software Reset	Additional Clock	MAX Functional Clock Frequency
0	-	0x000	-	-	-	-	-	-
1	-	0x004	-	-	-	-	-	-
2	-	0x008	-	-	-	-	-	-
3	-	0x00C	-	-	-	-	-	-
4	-	0x010	-	-	-	-	-	-
5	-	0x014	-	-	-	-	-	-
6	-	0x018	-	-	-	-	-	-
7	-	0x01C	-	-	-	-	-	-
8	DMA	0x020	FAST_BUS_CLK	OFF	-	YES	-	$\leq$ 100MHz
9	-	0x024	-	-	-	-	-	-
10	-	0x028	-	-	-	-	-	-
11	-	0x02C	-	-	-	-	-	-
12	-	0x030	-	-	-	-	-	-
13	-	0x034	-	-	-	-	-	-
14	-	0x038	-	-	-	-	-	-
15	TRACE	0x03C	-	-	YES	-	-	$\leq$ 100MHz
16	-	0x040	-	-	-	-	-	-
17	GPIO	0x044	CORE_CLK	OFF	YES	YES	-	$\leq$ 100MHz
18	PCTRLA	0x048	SLOW_BUS_CLK	-	-	YES	-	-
19	PCTRLB	0x04C	SLOW_BUS_CLK	-	-	YES	-	-
20	PCTRLC	0x050	SLOW_BUS_CLK	-	-	YES	-	-

Yuntu Microelectronics

Table 15.1 continued from previous page

No.	Module Name	Offset	Bus Clock	Bus Clock Enable	Peripheral Functional Clock	Software Reset	Additional Clock	MAX Functional Clock Frequency
21	PCTRLD	0x054	SLOW_BUS_CLK	-	-	YES	-	-
22	PCTRLE	0x058	SLOW_BUS_CLK	-	-	YES	-	-
23	-	0x05C	-	-	-	-	-	-
24	-	0x060	-	-	-	-	-	-
25	-	0x064	-	-	-	-	-	-
26	-	0x068	-	-	-	-	-	-
27	LINFlexD0	0x06C	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
28	LINFlexD1	0x070	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
29	LINFlexD2	0x074	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
30	LINFlexD3	0x078	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
31	LINFlexD4	0x07C	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
32	LINFlexD5	0x080	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
33	LINFlexD6	0x084	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
34	LINFlexD7	0x088	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
35	LINFlexD8	0x08C	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
36	LINFlexD9	0x090	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
37	I2C0	0x094	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
38	I2C1	0x098	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
39	I2C2	0x09C	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
40	I2C3	0x0A0	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
41	I2C4	0x0A4	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
42	SPI0	0x0A8	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
43	SPI1	0x0AC	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
44	SPI2	0x0B0	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
45	SPI3	0x0B4	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
46	SPI4	0x0B8	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
47	SPI5	0x0BC	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
48	SPI6	0x0C0	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
49	SPI7	0x0C4	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
50	-	0x0C8	-	-	-	-	-	-
51	-	0x0CC	-	-	-	-	-	-
52	FlexCAN0	0x0D0	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
53	FlexCAN1	0x0D4	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
54	FlexCAN2	0x0D8	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
55	FlexCAN3	0x0DC	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
56	FlexCAN4	0x0E0	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
57	FlexCAN5	0x0E4	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
58	FlexCAN6	0x0E8	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
59	FlexCAN7	0x0EC	FAST_BUS_CLK	OFF	-	YES	FXOSC_CLK	-
60	-	0x0F0	-	-	-	-	-	-
61	-	0x0F4	-	-	-	-	-	-
62	-	0x0F8	-	-	-	-	-	-
63	-	0x0FC	-	-	-	-	-	-
64	ADC0	0x100	FAST_BUS_CLK	OFF	YES	YES	-	$\leq$ 100MHz
65	ADC1	0x104	FAST_BUS_CLK	OFF	YES	YES	-	$\leq$ 100MHz
66	-	0x108	-	-	-	-	-	-
67	ACMP0	0x10C	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
68	ACMP1	0x110	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
69	-	0x114	-	-	-	-	-	-
70	PTU0	0x118	FAST_BUS_CLK	OFF	-	YES	-	-
71	PTU1	0x11C	FAST_BUS_CLK	OFF	-	YES	-	-
72	TMU	0x120	SLOW_BUS_CLK	-	-	YES	-	-
73	eTMR0	0x124	FAST_BUS_CLK	OFF	YES	YES	TCLK0/1/2	$\leq$ 100MHz
74	eTMR1	0x128	FAST_BUS_CLK	OFF	YES	YES	TCLK0/1/2	$\leq$ 100MHz
75	eTMR2	0x12C	FAST_BUS_CLK	OFF	YES	YES	TCLK0/1/2	$\leq$ 100MHz
76	eTMR3	0x130	FAST_BUS_CLK	OFF	YES	YES	TCLK0/1/2	$\leq$ 100MHz
77	eTMR4	0x134	FAST_BUS_CLK	OFF	YES	YES	TCLK0/1/2	$\leq$ 100MHz
78	eTMR5	0x138	FAST_BUS_CLK	OFF	YES	YES	TCLK0/1/2	$\leq$ 100MHz
79	-	0x13C	-	-	-	-	-	-
80	-	0x140	-	-	-	-	-	-
81	MPWM0	0x144	FAST_BUS_CLK	OFF	-	YES	-	-
82	MPWM1	0x148	FAST_BUS_CLK	OFF	-	YES	-	-
83	MPWM2	0x14C	FAST_BUS_CLK	OFF	-	YES	-	-

Yuntu Microelectronics

Table 15.1 continued from previous page

No.	Module Name	Offset	Bus Clock	Bus Clock Enable	Peripheral Functional Clock	Software Reset	Additional Clock	MAX Functional Clock Frequency
84	TMR0	0x150	SLOW_BUS_CLK	OFF	-	YES	-	-
85	TMR1	0x154	SLOW_BUS_CLK	OFF	-	YES	-	-
86	-	0x158	-	-	-	-	-	-
87	-	0x15C	-	-	-	-	-	-
88	SENT0	0x160	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
89	SENT1	0x164	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
90	pTMR0	0x168	SLOW_BUS_CLK	OFF	-	YES	-	-
91	pTMR1	0x16C	SLOW_BUS_CLK	OFF	-	YES	-	-
92	pTMR2	0x170	SLOW_BUS_CLK	OFF	-	YES	-	-
93	lpTMR0	0x174	SLOW_BUS_CLK	OFF	YES	YES	-	$\leq$ 50MHz
94	-	0x178	-	-	-	-	-	-
95	-	0x17C	-	-	-	-	-	-
96	RTC	0x180	SLOW_BUS_CLK	OFF	-	-	BUS_CLK/CLKIN/SIRC/IPC_CLK	-
97	-	0x184	-	-	-	-	-	-
98	WKU	0x188	SLOW_BUS_CLK	OFF	-	-	-	-
99	-	0x18C	-	-	-	-	-	-
100	-	0x190	-	-	-	-	-	-
101	PCRC	0x194	CORE_CLK	OFF	-	YES	-	-
102	TRNG	0x198	SLOW_BUS_CLK	OFF	-	YES	-	-
103	HCU	0x19C	FAST_BUS_CLK	OFF	-	YES	-	-
104	QSPI	0x1A0	CORE_CLK	OFF	YES	YES	-	$\leq$ 100MHz
105	-	0x1A4	-	-	-	-	-	-
106	WDG	0x1A8	SLOW_BUS_CLK	ON	YES	YES	WDG_FUNC_CLK/PCLK_WDG	$\leq$ 50MHz
107	-	0x1AC	-	-	-	-	-	-
108	-	0x1B0	-	-	-	-	-	-
109	-	0x1B4	-	-	-	-	-	-
110	-	0x1B8	-	-	-	-	-	-
111	-	0x1BC	-	-	-	-	-	-
112	FMU	0x1C0	FAST_BUS_CLK	ON	-	YES	-	-
113	INTM	0x1C4	SLOW_BUS_CLK	OFF	-	YES	-	-
114	EMU	0x1C8	FAST_BUS_CLK	-	-	YES	-	-
115	SAI0	0x1CC	SLOW_BUS_CLK	OFF	-	YES	FXOSC_CLK/PLL_CLK SAI0_MCLK_IND SAI1_MCLK_IND	-
116	SAI1	0x1D0	SLOW_BUS_CLK	OFF	-	YES	FXOSC_CLK/PLL_CLK SAI0_MCLK_IND SAI1_MCLK_IND	-
117	-	0x1D4	-	-	-	-	-	-
118	-	0x1D8	-	-	-	-	-	-
119	-	0x1DC	-	-	-	-	-	-
120	-	0x1E0	-	-	-	-	-	-
121	-	0x1E4	-	-	-	-	-	-
122	-	0x1E8	-	-	-	-	-	-
123	CIM	0x1EC	SLOW_BUS_CLK	-	-	YES	-	-
124	SCU	0x1F0	SLOW_BUS_CLK	ON	-	YES	-	-
125	-	0x1F4	-	-	-	-	-	-
126	PCU	0x1F8	SLOW_BUS_CLK	ON	-	-	-	-
127	RCU	0x1FC	SLOW_BUS_CLK	ON	-	-	SIRC_CLK	-

Yuntu Microelectronics

Note:

1. Bus clock is used to operate registers of IPs, if Bus clock gate of corresponding IP is off, registers will not be required to operate. If operates, hard fault will generate.
2. Peripheral Functional Clock “YES” means its clock source and clock divider can be configured in the IPC register. “-” means can’t.
3. Bus clock Enable “ON” means default IP clock enable, “OFF” means default IP clock is disable.
4. LINFlexD module baudrate is calculated according to FAST_BUS_CLK.
5. FlexCAN module clock source can be selected as FAST_BUS_CLK or FXOSC_CLK, this is set in register of FlexCAN IP.

6. eTMR module clock source is FAST_BUS_CLK default, also can use TCLK through configuring in the eTMR register, and TCLK frequency must be less than the half of FAST_BUS_CLK.
7. lpTMR module clock source can be selected to func_clk, if selected to func_clk, the clock source is configured by IPC clock source selection, if MCU enters into Powerdown mode, IPC will be disable. If lpTMR is required to use, it is necessary to select SIRC or SXOSC in the lpTMR register.
8. RTC module clock source is only configured in the module register, it can be used in the low power mode.
9. WDG module clock source is only configured in the module register.
10. RCU module digital filter function needs SIRC.

15.5 Register Definition

15.5.1 IPC Memory Map

IPC base address 401f4000h

Table 15.2: IPC Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL0)	32	RW	00000000h
4h	Control Register (CTRL1)	32	RW	00000000h
8h	Control Register (CTRL2)	32	RW	00000000h
ch	Control Register (CTRL3)	32	RW	00000000h
10h	Control Register (CTRL4)	32	RW	00000000h
14h	Control Register (CTRL5)	32	RW	00000000h
18h	Control Register (CTRL6)	32	RW	00000000h
1ch	Control Register (CTRL7)	32	RW	00000000h
20h	Control Register (CTRL8)	32	RW	00000000h
24h	Control Register (CTRL9)	32	RW	00000000h
28h	Control Register (CTRL10)	32	RW	00000000h
2ch	Control Register (CTRL11)	32	RW	00000000h
30h	Control Register (CTRL12)	32	RW	00000000h
34h	Control Register (CTRL13)	32	RW	00000000h
38h	Control Register (CTRL14)	32	RW	00000000h
3ch	Control Register (CTRL15)	32	RW	00000000h
40h	Control Register (CTRL16)	32	RW	00000000h
44h	Control Register (CTRL17)	32	RW	00000000h
48h	Control Register (CTRL18)	32	RW	00000000h
4ch	Control Register (CTRL19)	32	RW	00000000h
50h	Control Register (CTRL20)	32	RW	00000000h
54h	Control Register (CTRL21)	32	RW	00000000h
58h	Control Register (CTRL22)	32	RW	00000000h
5ch	Control Register (CTRL23)	32	RW	00000000h
60h	Control Register (CTRL24)	32	RW	00000000h
64h	Control Register (CTRL25)	32	RW	00000000h
68h	Control Register (CTRL26)	32	RW	00000000h
6ch	Control Register (CTRL27)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
70h	Control Register (CTRL28)	32	RW	00000000h
74h	Control Register (CTRL29)	32	RW	00000000h
78h	Control Register (CTRL30)	32	RW	00000000h
7ch	Control Register (CTRL31)	32	RW	00000000h
80h	Control Register (CTRL32)	32	RW	00000000h
84h	Control Register (CTRL33)	32	RW	00000000h
88h	Control Register (CTRL34)	32	RW	00000000h
8ch	Control Register (CTRL35)	32	RW	00000000h
90h	Control Register (CTRL36)	32	RW	00000000h
94h	Control Register (CTRL37)	32	RW	00000000h
98h	Control Register (CTRL38)	32	RW	00000000h
9ch	Control Register (CTRL39)	32	RW	00000000h
a0h	Control Register (CTRL40)	32	RW	00000000h
a4h	Control Register (CTRL41)	32	RW	00000000h
a8h	Control Register (CTRL42)	32	RW	00000000h
ach	Control Register (CTRL43)	32	RW	00000000h
b0h	Control Register (CTRL44)	32	RW	00000000h
b4h	Control Register (CTRL45)	32	RW	00000000h
b8h	Control Register (CTRL46)	32	RW	00000000h
bch	Control Register (CTRL47)	32	RW	00000000h
c0h	Control Register (CTRL48)	32	RW	00000000h
c4h	Control Register (CTRL49)	32	RW	00000000h
c8h	Control Register (CTRL50)	32	RW	00000000h
cch	Control Register (CTRL51)	32	RW	00000000h
d0h	Control Register (CTRL52)	32	RW	00000000h
d4h	Control Register (CTRL53)	32	RW	00000000h
d8h	Control Register (CTRL54)	32	RW	00000000h
dch	Control Register (CTRL55)	32	RW	00000000h
e0h	Control Register (CTRL56)	32	RW	00000000h
e4h	Control Register (CTRL57)	32	RW	00000000h
e8h	Control Register (CTRL58)	32	RW	00000000h
ech	Control Register (CTRL59)	32	RW	00000000h
f0h	Control Register (CTRL60)	32	RW	00000000h
f4h	Control Register (CTRL61)	32	RW	00000000h
f8h	Control Register (CTRL62)	32	RW	00000000h
fch	Control Register (CTRL63)	32	RW	00000000h
100h	Control Register (CTRL64)	32	RW	00000000h
104h	Control Register (CTRL65)	32	RW	00000000h
108h	Control Register (CTRL66)	32	RW	00000000h
10ch	Control Register (CTRL67)	32	RW	00000000h
110h	Control Register (CTRL68)	32	RW	00000000h
114h	Control Register (CTRL69)	32	RW	00000000h
118h	Control Register (CTRL70)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
11ch	Control Register (CTRL71)	32	RW	00000000h
120h	Control Register (CTRL72)	32	RW	00000000h
124h	Control Register (CTRL73)	32	RW	00000000h
128h	Control Register (CTRL74)	32	RW	00000000h
12ch	Control Register (CTRL75)	32	RW	00000000h
130h	Control Register (CTRL76)	32	RW	00000000h
134h	Control Register (CTRL77)	32	RW	00000000h
138h	Control Register (CTRL78)	32	RW	00000000h
13ch	Control Register (CTRL79)	32	RW	00000000h
140h	Control Register (CTRL80)	32	RW	00000000h
144h	Control Register (CTRL81)	32	RW	00000000h
148h	Control Register (CTRL82)	32	RW	00000000h
14ch	Control Register (CTRL83)	32	RW	00000000h
150h	Control Register (CTRL84)	32	RW	00000000h
154h	Control Register (CTRL85)	32	RW	00000000h
158h	Control Register (CTRL86)	32	RW	00000000h
15ch	Control Register (CTRL87)	32	RW	00000000h
160h	Control Register (CTRL88)	32	RW	00000000h
164h	Control Register (CTRL89)	32	RW	00000000h
168h	Control Register (CTRL90)	32	RW	00000000h
16ch	Control Register (CTRL91)	32	RW	00000000h
170h	Control Register (CTRL92)	32	RW	00000000h
174h	Control Register (CTRL93)	32	RW	00000000h
178h	Control Register (CTRL94)	32	RW	00000000h
17ch	Control Register (CTRL95)	32	RW	00000000h
180h	Control Register (CTRL96)	32	RW	00000000h
184h	Control Register (CTRL97)	32	RW	00000000h
188h	Control Register (CTRL98)	32	RW	00000000h
18ch	Control Register (CTRL99)	32	RW	00000000h
190h	Control Register (CTRL100)	32	RW	00000000h
194h	Control Register (CTRL101)	32	RW	00000000h
198h	Control Register (CTRL102)	32	RW	00000000h
19ch	Control Register (CTRL103)	32	RW	00000000h
1a0h	Control Register (CTRL104)	32	RW	00000000h
1a4h	Control Register (CTRL105)	32	RW	00000000h
1a8h	Control Register (CTRL106)	32	RW	00000000h
1ach	Control Register (CTRL107)	32	RW	00000000h
1b0h	Control Register (CTRL108)	32	RW	00000000h
1b4h	Control Register (CTRL109)	32	RW	00000000h
1b8h	Control Register (CTRL110)	32	RW	00000000h
1bch	Control Register (CTRL111)	32	RW	00000000h
1c0h	Control Register (CTRL112)	32	RW	00000000h
1c4h	Control Register (CTRL113)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
1c8h	Control Register (CTRL114)	32	RW	00000000h
1cch	Control Register (CTRL115)	32	RW	00000000h
1d0h	Control Register (CTRL116)	32	RW	00000000h
1d4h	Control Register (CTRL117)	32	RW	00000000h
1d8h	Control Register (CTRL118)	32	RW	00000000h
1dch	Control Register (CTRL119)	32	RW	00000000h
1e0h	Control Register (CTRL120)	32	RW	00000000h
1e4h	Control Register (CTRL121)	32	RW	00000000h
1e8h	Control Register (CTRL122)	32	RW	00000000h
1ech	Control Register (CTRL123)	32	RW	00000000h
1f0h	Control Register (CTRL124)	32	RW	00000000h
1f4h	Control Register (CTRL125)	32	RW	00000000h
1f8h	Control Register (CTRL126)	32	RW	00000000h
1fch	Control Register (CTRL127)	32	RW	00000000h

15.5.1.1 IPC CTRL Register

Offset: 0h + n * 4h

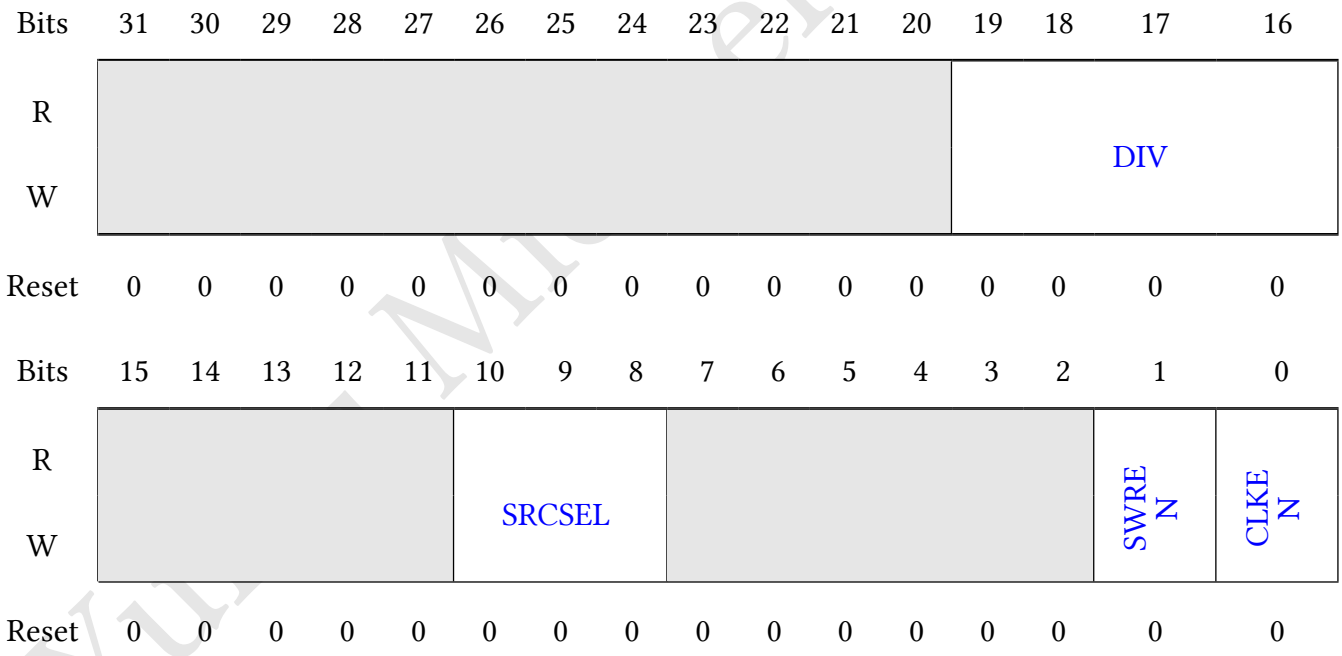


Table 15.3: IPC CTRL Register Description

Field	Function
19 - 16 DIV	**Functional Clock divider** The divider value = register value + 1
10 - 8	**Functional Clock Source Selection**

Field	Function
SRCSEL	000b - no clock 001b - select FIRC 010b - select SIRC_DIV4_CLK 011b - select FXOSC 100b - select SXOSC 101b - select PLL 110b - no clock 111b - FAST_BUS_CLK
1 SWREN	**Software reset** Not all modules support, please see Table 13.1 SWR column for more information. 0b - Software reset is de-asserted 1b - Software reset is asserted
0 CLKEN	**Clock enable** Not all modules support, please see Table 13.1 Bus Clock Enable column for more information. 0b - IP clock is disabled 1b - IP clock is enabled

15.6 Functional Description

The IPC module provides dedicated IPC registers for clock gating and configuration options for on-chip peripherals (modules). Each module's IPC register contains a clock gating control bit (CLKEN) for its bus clock. Before accessing one module, its bus clock must be enabled (CLKEN = 1) in the module's IPC register.

If a module has a functional clock, its IPC register may provide the clock source options, selected by programming the Clock Source Selection (SRCSEL) field, and a module also have a clock divider, selected by writing the Clock Divider (DIV) field.

NOTE:

It is necessary to configure DIV after setting SRCSEL as 0, and then set the specified clock source you want.

Yuntu Microelectronics

Chapter 16

Reset Control Unit (RCU)

16.1 Introduction

Reset Control Unit implements the reset functions for the chip. The Reset Control Unit handles the system reset control when it receives the reset sources.

The supported reset sources including:

- Power on reset or low voltage detect
- Pin reset
- Software reset
- Debug reset
- FMU functional reset
- ARM core lockup reset
- Clock monitor reset
- Watch dog reset
- Low power mode entry acknowledge timeout reset
- High voltage detect reset
- FMU fail react reset

16.1.1 Features

- Record the reset sources of most recent resets.
- Configurable filter for reset pin.
- Reset pin filter can work at both active and low power mode.

16.2 Register Definition

16.2.1 RCU Memory Map

RCU base address 401fc000h

Table 16.1: RCU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Reset Source Status Register (RSSR)	32	RW	00000001h
4h	Reset Pin Filter Register (RPFR)	32	RW	0002FF00h

Yuntu Microelectronics

16.2.1.1 RCU RSSR Register

This register includes read-only status flags to indicate the source of the most recent reset. Multiple flags will be set if multiple reset events occur at the same time.

Offset: 0h

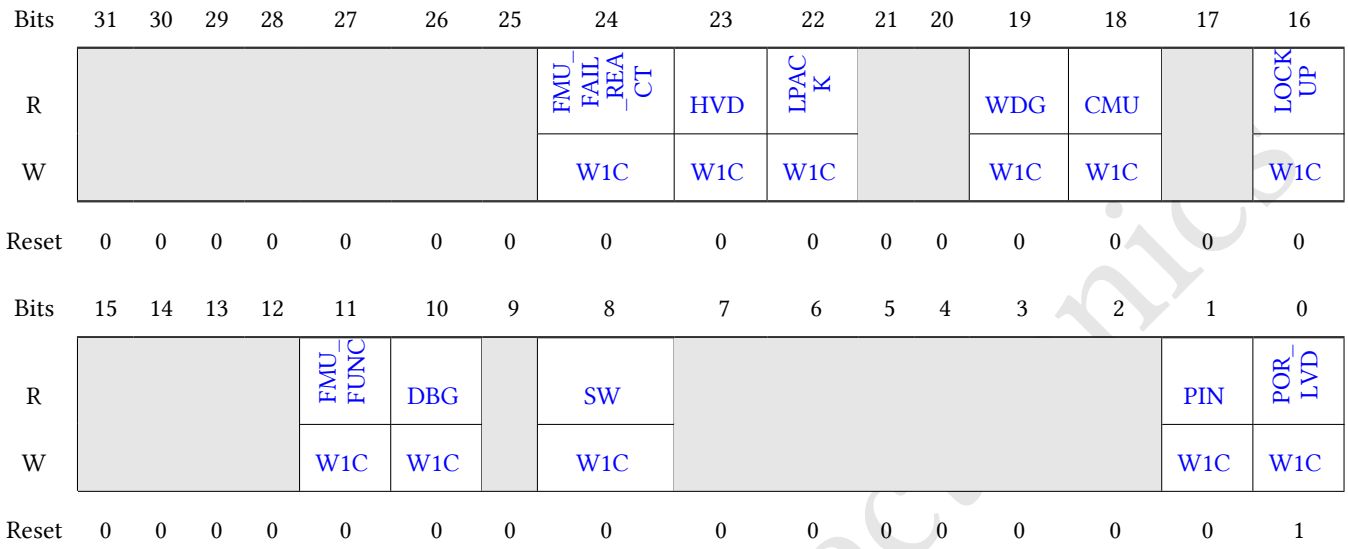


Table 16.2: RCU RSSR Register Description

Field	Function
24 FMU_FAIL_REACT	**FMU Fail React Reset** System reset caused by FMU Fail React. 0b - System reset not caused by FMU Fail React. 1b - System reset caused by FMU Fail React.
23 HVD	**High Voltage Detect Reset** System reset caused by Clock Monitor HVD. 0b - System reset not caused by HVD. 1b - System reset caused by HVD.
22 LPACK	**Low Power Mode Entry Acknowledge Timeout Reset** System reset caused by low power mode entry acknowledge timeout, indicates that a system reset has been caused during an attempt to enter the low power mode (Deepsleep/Standby/Powerdown Mode), while one or more peripherals fail to acknowledge on time. 0b - System reset not caused by Deepsleep/Standby/Powerdown mode entry timeout. 1b - System reset caused by Deepsleep/Standby/Powerdown mode entry timeout.
19 WDG	**Watch Dog Reset** System reset caused by Clock Monitor Watchdog. 0b - System reset not caused by WDG. 1b - System reset caused by WDG.

Yuntu Microelectronics

Field	Function
18 CMU	**Clock Monitor Reset** System reset caused by Clock Monitor if CMURE is set. 0b - System reset not caused by CMU. 1b - System reset caused by CMU.
16 LOCKUP	**ARM Core Lockup Reset** System reset caused by ARM Core Lockup. 0b - System reset not caused by Lockup. 1b - System reset caused by Lockup.
11 FMU_FUNC	**FMU Functional Reset** System reset caused by FMU. 0b - System reset not caused by FMU. 1b - System reset caused by FMU.
10 DBG	**Debug Reset** System reset caused by Debug. 0b - System reset not caused by debug. 1b - System reset caused by debug.
8 SW	**Software Reset** System reset caused by software to set SYSRESETREQ of ARM Core. 0b - System reset not caused by SW. 1b - System reset caused by SW.
1 PIN	**Pin Reset** System reset caused by RESETB pin. 0b - System reset not caused by PIN. 1b - System reset caused by PIN.
0 POR_LVD	**Power On Reset or Low Voltage Detect** System reset caused by POR/LVD. 0b - System reset not caused by POR/LVD. 1b - System reset caused by POR/LVD.

Yuntu Microelectronics

16.2.1.2 RCU RPFR Register

This register controls the reset pin filter functions.

Offset: 4h

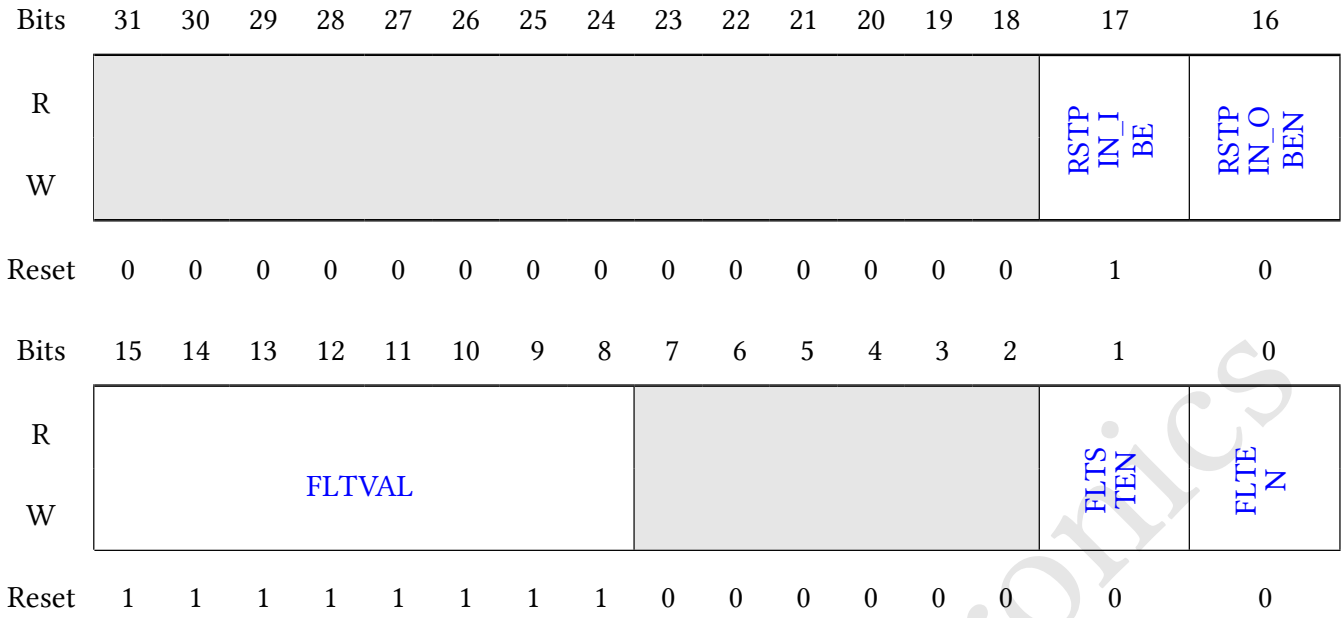


Table 16.3: RCU RPFR Register Description

Field	Function
17 RSTPIN_IBE	**Reset Pin Input Enable** Enable bit of Reset pin input 0b - reset pin ind is disabled 1b - reset pin ind is enabled
16 RSTPIN_OBEN	**Reset Pin Output Enable** Enable bit of Reset pin output 0b - Not output low on reset pin when chip reset occurs 1b - Output low on reset pin when chip reset occurs
15 - 8 FLTVAL	**Reset Pin Filter Counter Value** FLTVAL control the filter counter value of SIRC_DIV4_CLK cycles 00000000b (0x00) - Filter counter value is 0x00 11111111b (0xff) - Filter counter value is 0xFF
1 FLTSTEN	**Reset Pin Filter Enable at Deepsleep and Standby mode** Enable bit of Reset pin filter in Deepsleep and Standby mode. This bit should be set if filter need to be used in Deepsleep and Standby mode. This bit has no effect of reset pin filter in Active mode. 0b - Disable reset pin filter in Deepsleep and Standby mode 1b - Enable reset pin filter in Deepsleep and Standby mode if FLTEN is set
0 FLTEN	**Reset Pin Filter Enable** Enable bit of Reset pin filter, this bit should be set if filter need to be used. 0b - Disable reset pin filter 1b - Enable reset pin filter

Yuntu Microelectronics

16.3 Functional Description

16.3.1 External Pin Reset

A reset is caused by an active-low level on external RESET_b pin. The RESET_b pin supports filter function using SIRC_DIV4_CLK with the control of RPFR registers.

16.3.2 POR and LVD

When MCU is power on, the POR and LVD reset assert. Once the operating voltage attains a usable level, the FIRC and SIRC starts. After FIRC/SIRC start up, the system clocks are available, but the system reset remains asserted until the external pin reset is released. Then the flash controller(EFM) starts its initialization. Once flash initialization completes, ARM core starts to fetch code and operation.

16.3.3 High Voltage Detect Reset

If PCU_CTRL[HVDRE] bit is set and the supply of VDD/VDD25/VDD11 rises above the HVD trip voltage, HVD reset occurs.

16.3.4 Clock Monitor Reset

Clock Monitor Unit(CMU) is a sub module of System Clock Unit(SCU). When it is enabled, it continues to monitor the clock sources. Once the monitored or checked clocks are out of proper frequency range or lost, a reset occurs. If the monitored or checked clock is available but reference clock is lost, a reset occurs.

16.3.5 Watchdog Reset

Watchdog timer monitor the operation of MCU by expecting periodic refresh from software. If the periodic refreshing does not occur, the watchdog can generate interrupt or reset as the configuration of WDG module.

16.3.6 Low Power Mode Entry Acknowledge Timeout Reset

The low power mode entry acknowledge timeout reset is generated if the MCU fails to enter low power mode mode within 2^24 cycles of SIRC clock.

16.3.7 Software Reset

When the application needs to reset the MCU through software, it could set SYSRESETREQ register bit of ARM core to generate a system reset.

16.3.8 Core Lockup Reset

LOCKUP immediately shows a serious error with the kernel software. This is the result of the core being locked due to an unrecoverable exception after the processor's built-in system state protection hardware is activated. LOCKUP causes a system reset if CIM_CTRL1[LOCKUPEN] bit is set.

16.3.9 Debug Reset

Using CDBGRSTREQ of debug module could cause a system reset.

Yuntu Microelectronics

16.3.10 FMU Functional Reset

Refer to FMU Function.

16.3.11 FMU Fail React Reset

Refer to FMU Function.

Yuntu Microelectronics

Chapter 17

Power Control Unit (PCU)

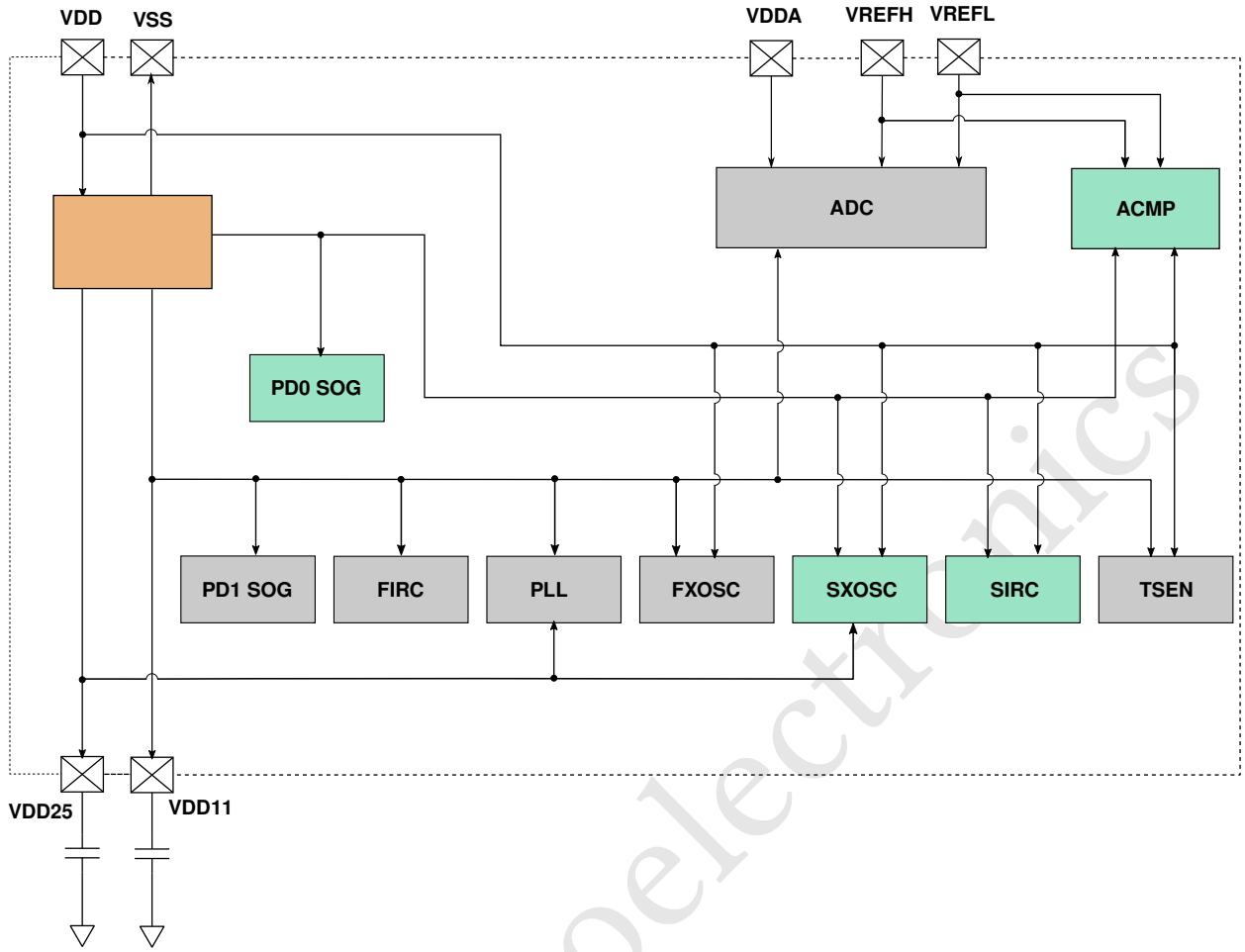
17.1 Introduction

The PCU is the Power Control Unit for the YTM32B1H Family microcontrollers. It provides multiple power options to allow the user to optimize power consumption for the level of functionality needed. It includes the internal voltage regulators, power-on reset (POR) and the integrated low/high voltage detect system with reset (brownout) capability. The voltage regulator requires a 3.3V or 5V input to generate all the required secondary supplies.

Yuntu Microelectronics

17.1.1 Block Diagram

The following figure shows the PCU block diagram.



NOTE

1. PD0: always on power domain

2. PD1: power off in Powerdown mode

Figure 17.1: PCU Block Diagram

Table 17.1: PCU Block Diagram Table

Signal	I/O	Direction
VDD	Supply input	This is the high-voltage supply input to PMC. VDD is used for the PMC internal precision references. After the VDD domain is powered up, it must be kept powered at all times of operation.
VDD25	Supply output	VDD25 power supply domain is driven by a fully integrated LDO voltage regulator.
VDD11	Supply output	VDD11 is the core/logic supply. It is driven by a fully integrated LDO voltage regulator.
VSS	Ground	VSS must be grounded. All VSS Pins need to be externally connected to the same ground node.
VDDA	Supply input	This is the high-voltage supply input to Analog module.
VREFH	Reference supply input	This is the high-voltage reference supply input to Analog module.

Yuntu Microelectronics

Table 17.1 continued from previous page

Signal	I/O	Direction
VREFL	Reference supply input	This is the low-voltage reference supply input to Analog module.

17.1.2 Features

- Combination of internal and external voltage regulator options, offering a variety of power modes
- Active POR providing brown-out detect
- Low voltage reset for all system relevant power domains (LVR)
- High voltage detect (HVD) as indication for software.

17.2 Register Definition

This section includes the PCU module memory map and detailed descriptions of all registers.

17.2.1 PCU Memory Map

PCU base address 401f8000h

Table 17.2: PCU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Sticky Status Register (SSTS)	32	RW	80000000h
4h	Status Register (STS)	32	RW	03000000h
8h	Interrupt Register (INTE)	32	RW	00000000h
ch	Control Register (CTRL)	32	RW	00000000h

17.2.1.1 PCU SSTS Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PORF								GNG1 1F				HVD1 1F			
W	W1C								W1C				W1C			
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GNG2 5F				HVD2 5F								HVD5 0F			
W	W1C				W1C								W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 17.3: PCU SSTS Register Description

Field	Function
31 PORF	**POR Flag** Flags that a power on reset event has occurred. Write 1 to clear it. Other reset sources have no effect. 0b - No power-on reset event has occurred. 1b - Power-on reset event has occurred.
23 GNG11F	**Go/NoGo Detect Flag on VDD11 Domain** Flags that the Go/NoGo sensor has detected a low voltage on the VDD11 domain in FPM. This applies to only to the part of the domain that supplies the 1.1V clocking modules (e.g. FIRC, SIRC). Write 1 to clear it. Other reset sources have no effect. 0b - No event has occurred. 1b - Go event has occurred.
19 HVD11F	**High Voltage Detect Flag on VDD11 in FPM** This ready-only status bit indicates a high voltage detect event on VDD11. Write 1 to clear this bit. 0b - High voltage event on VDD11 not detected. 1b - High voltage event on VDD11 detected.
15 GNG25F	**Go/NoGo Detect Flag on VDD25 Domain** Flags that the Go/NoGo sensor has detected a low voltage on the VDD25 domain in FPM. This applies to only to the part of the domain that supplies the 2.5V clocking modules (e.g. FXOSC, PLL). Write 1 to clear it. Other reset sources have no effect. 0b - No event has occurred. 1b - Go event has occurred.
11 HVD25F	**High Voltage Detect Flag on VDD25 in FPM** This ready-only status bit indicates a high voltage detect event on VDD25. Write 1 to clear this bit. 0b - High voltage event on VDD25 not detected. 1b - High voltage event on VDD25 detected.
3 HVD50F	**High Voltage Detect Flag on VDD in FPM** This ready-only status bit indicates a high voltage detect event on VDD. Write 1 to clear this bit. 0b - High voltage event on VDD not detected. 1b - High voltage event on VDD detected.

Yuntu Microelectronics

17.2.1.2 PCU STS Register

Offset: 4h

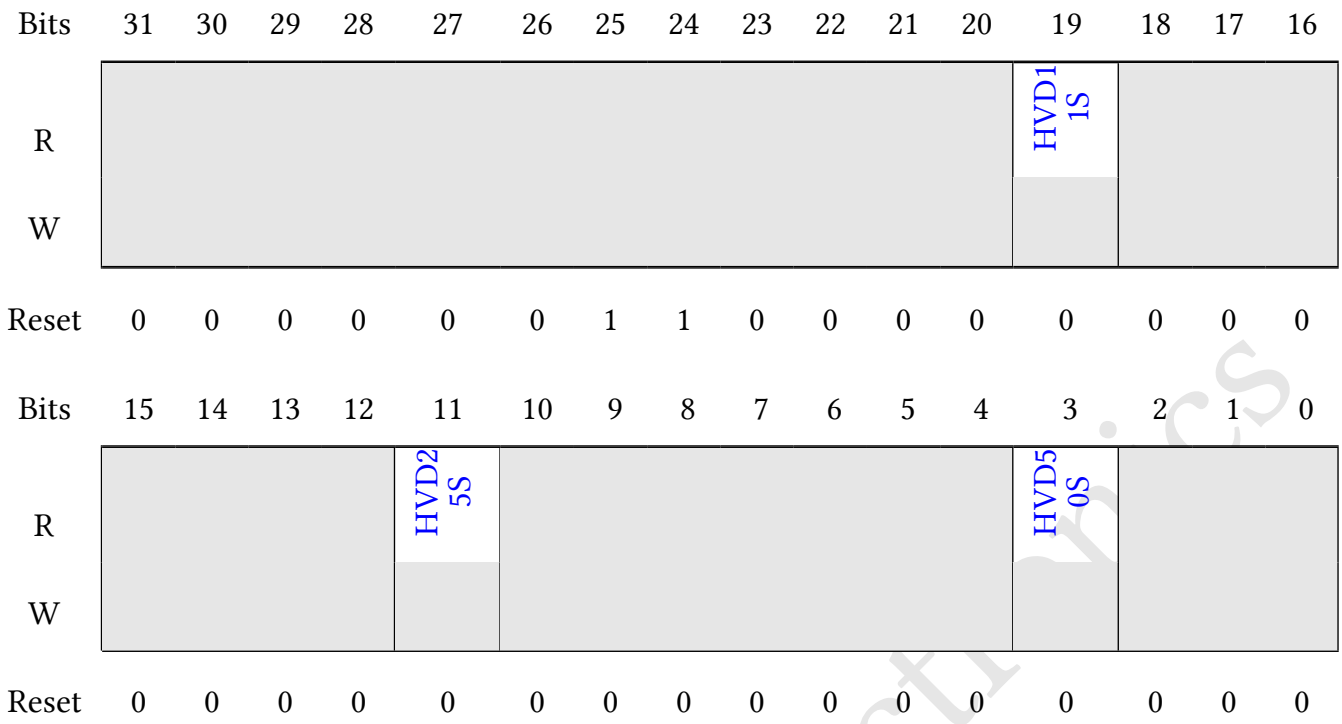


Table 17.4: PCU STS Register Description

Field	Function
19 HVD11S	**High Voltage Detect Status on VDD11 in FPM** Shows the status of the high voltage detect on VDD11 power domain. 0b - Voltage on VDD11 is below high-voltage detect threshold. 1b - Voltage on VDD11 is above high-voltage detect threshold.
11 HVD25S	**High Voltage Detect Status on VDD25 in FPM** Shows the status of the high voltage detect on VDD25 power domain. 0b - Voltage on VDD25 is below high-voltage detect threshold. 1b - Voltage on VDD25 is above high-voltage detect threshold.
3 HVD50S	**High Voltage Detect Status on VDD in FPM** Shows the status of the high voltage detect on VDD power domain. 0b - Voltage on VDD is below high-voltage detect threshold. 1b - Voltage on VDD is above high-voltage detect threshold.

17.2.1.3 PCU INTE Register

Offset: 8h

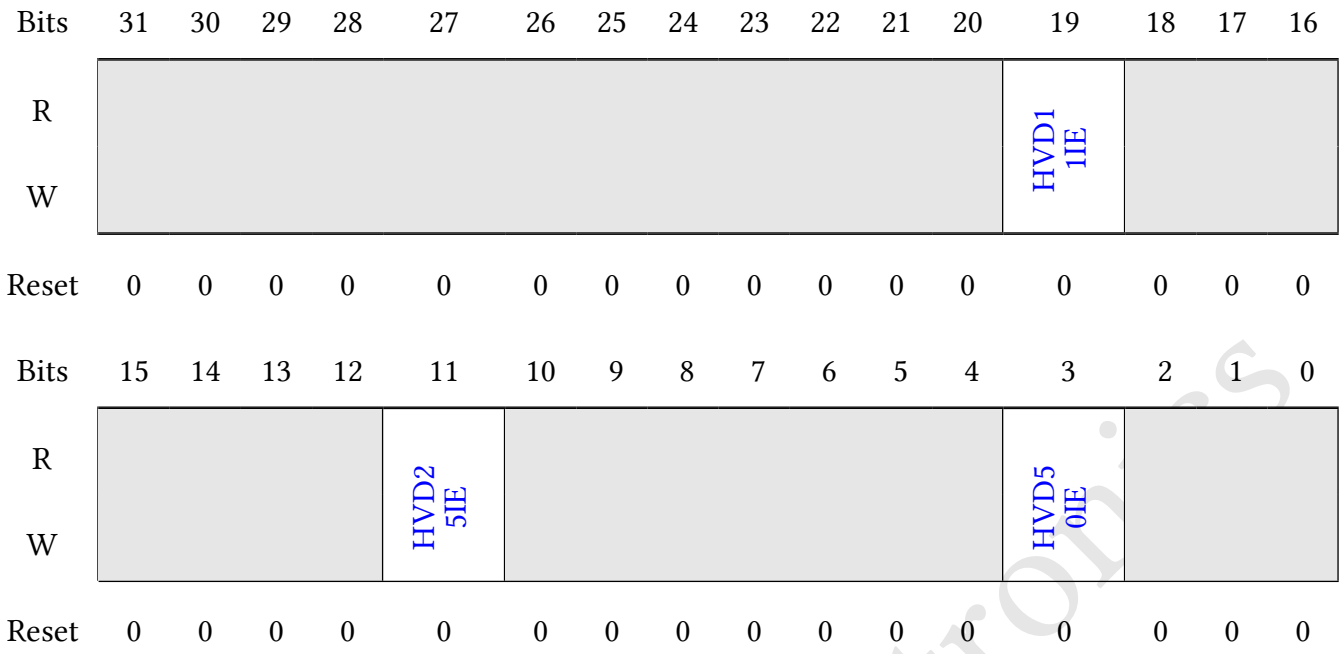


Table 17.5: PCU INTE Register Description

Field	Function
19 HVD11IE	**VDD11 High Voltage Detect Interrupt Enable** This bit enables hardware interrupt requests if HVD11F is set. 0b - HVD hardware interrupt is disabled (use polling) 1b - Request a HVD hardware interrupt when HVD11F is set
11 HVD25IE	**VDD25 High Voltage Detect Interrupt Enable** This bit enables hardware interrupt requests if HVD25F is set. 0b - HVD hardware interrupt is disabled (use polling) 1b - Request a HVD hardware interrupt when HVD25F is set
3 HVD50IE	**VDD High Voltage Detect Interrupt Enable** This bit enables hardware interrupt requests if HVD50F is set. 0b - HVD hardware interrupt is disabled (use polling) 1b - Request a HVD hardware interrupt when HVD50F is set

Yuntu Microelectronics

17.2.1.4 PCU CTRL Register

Offset: ch

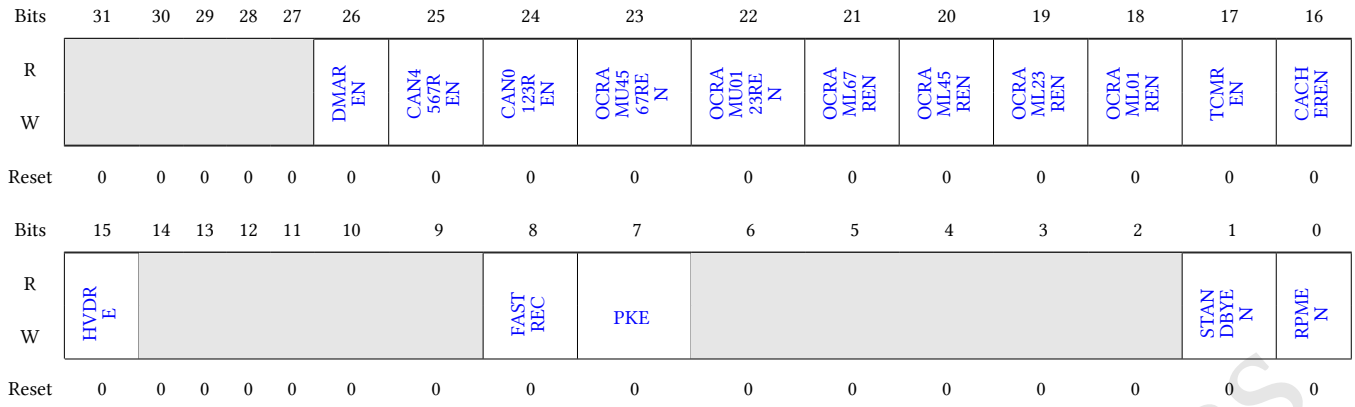


Table 17.6: PCU CTRL Register Description

Field	Function
26 DMAREN	**DMA RAM Retention Enable** This bit controls whether DMA RAM enters retention state when chip enters standby mode. 0b - DMA RAM doesn't enter retention state when chip enters standby mode. 1b - DMA RAM enters retention state when chip enters standby mode.
25 CAN4567REN	**CAN4/5/6/7 RAM Retention Enable** This bit controls whether CAN4/5/6/7 RAM enters retention state when chip enters standby mode. 0b - CAN4/5/6/7 RAM doesn't enter retention state when chip enters standby mode. 1b - CAN4/5/6/7 RAM enters retention state when chip enters standby mode.
24 CAN0123REN	**CAN0/1/2/3 RAM Retention Enable** This bit controls whether CAN0/1/2/3 RAM enters retention state when chip enters standby mode. 0b - CAN0/1/2/3 RAM doesn't enter retention state when chip enters standby mode. 1b - CAN0/1/2/3 RAM enters retention state when chip enters standby mode.
23 OCRAMU4567REN	**OCRAMU (Address 0x20050000-0x2004FFFF) Retention Enable** This bit controls whether OCRAMU (Address 0x20050000-0x2004FFFF) enters retention state when chip enters standby mode. 0b - OCRAMU (Address 0x20050000-0x2004FFFF) doesn't enter retention state when chip enters standby mode. 1b - OCRAMU (Address 0x20050000-0x2004FFFF) enters retention state when chip enters standby mode.
22	**OCRAMU (Address 0x20040000-0x2004FFFF) Retention Enable**

Yuntu Microelectronics

Field	Function
OCRAMU0123REN	This bit controls whether OCRAMU (Address 0x20040000-0x2004FFFF) enters retention state when chip enters standby mode. 0b - OCRAMU (Address 0x20040000-0x2004FFFF) doesn't enter retention state when chip enters standby mode. 1b - OCRAMU (Address 0x20040000-0x2004FFFF) enters retention state when chip enters standby mode.
21 OCRAML67REN	**OCRAML (Address 0x20038000-0x2003FFFF) Retention Enable** This bit controls whether OCRAML (Address 0x20038000-0x2003FFFF) enters retention state when chip enters standby mode. 0b - OCRAML (Address 0x20038000-0x2003FFFF) doesn't enter retention state when chip enters standby mode. 1b - OCRAML (Address 0x20038000-0x2003FFFF) enters retention state when chip enters standby mode.
20 OCRAML45REN	**OCRAML (Address 0x20030000-0x20037FFF) Retention Enable** This bit controls whether OCRAML (Address 0x20030000-0x20037FFF) enters retention state when chip enters standby mode. 0b - OCRAML (Address 0x20030000-0x20037FFF) doesn't enter retention state when chip enters standby mode. 1b - OCRAML (Address 0x20030000-0x20037FFF) enters retention state when chip enters standby mode.
19 OCRAML23REN	**OCRAML (Address 0x20028000-0x2002FFFF) Retention Enable** This bit controls whether OCRAML (Address 0x20028000-0x2002FFFF) enters retention state when chip enters standby mode. 0b - OCRAML (Address 0x20028000-0x2002FFFF) doesn't enter retention state when chip enters standby mode. 1b - OCRAML (Address 0x20028000-0x2002FFFF) enters retention state when chip enters standby mode.
18 OCRAML01REN	**OCRAML (Address 0x20020000-0x20027FFF) Retention Enable** This bit controls whether OCRAML (Address 0x20020000-0x20027FFF) enters retention state when chip enters standby mode. 0b - OCRAML (Address 0x20020000-0x20027FFF) doesn't enter retention state when chip enters standby mode. 1b - OCRAML (Address 0x20020000-0x20027FFF) enters retention state when chip enters standby mode.
17 TCMREN	**TCM Retention Enable** This bit controls whether ITCM/DTCM enters retention state when chip enters standby mode. 0b - ITCM/DTCM doesn't enter retention state when chip enters standby mode. 1b - ITCM/DTCM enters retention state when chip enters standby mode.
16	**Cache RAM Retention Enable**

Yuntu Microelectronics

Field	Function
CACHEREN	This bit controls whether Cache RAM enters retention state when chip enters standby mode. 0b - Cache RAM doesn't enter retention state when chip enters standby mode. 1b - Cache RAM enters retention state when chip enters standby mode.
15 HVDRE	**High Voltage Detect Reset Enable** This bit controls whether high voltage detect event generates a reset. 0b - Don't generate reset when a high voltage detect event occurs. 1b - Generate reset when a high voltage detect event occurs.
8 FASTREC	**Fast Recovery from LPM Enable Bit** This bit controls the recovery time from Low Power Mode (LPM) to Full Performance Mode (FPM). At recovery from Low Power Mode all the tank capacitors from the secondary supplies have to re recharged. This causes a high current demand, that might not be met by the supply driving the VDD primary domain. When selecting the fast recovery time, the current for recharging is approximately 3 times higher than for FASTREC=0. 0b - Normal recovery time from LPM 1b - Fast recovery time from LPM
7 PKE	**PAD Keep Enable Bit** This bit is to control PAD state when wakeup from power mode. 0b - PAD state is reseted when wakeup from Powerdown mode. 1b - PAD state is kept when wakeup from Powerdown mode until PKE bit is cleared to 0. NOTE: If Powerdown mode is waken up by reset pin, PAD state is always reseted.
1 STANDBYEN	**Standby Mode Enable Bit** This bit enables chip to enter Standby mode. 0b - Disable Standby mode entry 1b - Enable Standby mode entry
0 RPMEN	**Reduced Performance Mode Enable Bit** This bit controls the reduce performance mode entry or not. 0b - Disable RPM entry 1b - Enable RPM entry

17.3 Functional Description

The following sections describe functional details of the PCU.

17.3.1 Reset

The Power On Reset (POR) and all Low Voltage Reset monitors (LVRs) are combined into one single MCU POR.

After a MCU POR event, it can be determined which power domain caused it by reading in the STS or

Yuntu Microelectronics

SSTS register.

After initial power ramp up of the MCU, the POR flag and the LVR flag in SSTS register are all set to 1. The Go/Nogo flags will have an arbitrary value.

NOTE. After initial power ramp up, all flags in SSTS register must be cleared (by writing 0xFFFFFFFF to SSTS register).

As the flags are sticky bits, it is required to clear them before usage, so e.g. in case of unexpected MCU POR, the source of the problem can be tracked and debugged by reading the flags in STS and SSTS registers.

17.3.2 Interrupt

High Voltage Detect Interrupt is available in PCU and it can be enabled by setting HVDxxIE bit in INTE register.

17.4 Glossary

POR	Power-on-reset
HVD	High-voltage detection
LVD	Low-voltage detection

Yuntu Microelectronics

Chapter 18

Power Mode

18.1 Introduction

This chapter describes the various power modes supported by the chip.

18.1.1 Active Mode

Allow maximum performance of the chip and use all functions of the chip. It is the default mode when out of reset or power on reset. The on-chip regulators are working at high power mode to supply powers for internal modules. In this mode the chip can work at max frequency 120MHz.

18.1.2 Sleep Mode

Part of the chip is in static state, in Sleep mode, the ARM Cortex-M7 core is stopped, but all the other modules are working same as they are in active mode. All the registers and memory contents are retained. LVD protection is maintained. NVIC remains active to interrupts in Sleep mode.

18.1.3 Deepsleep Mode

The chip is in static state but the high power on chip regulators keep working, all registers and memories retain values, NVIC inactive, peripheries can work with asynchronous clock sources other than bus clocks, asynchronous interrupts from peripheries can wake up chip from Deepsleep mode.

18.1.4 Standby Mode

The chip is in static state, high power regulators are inactive, the chip is powered by low power on chip regulators, all registers and memories retain values, NVIC inactive, peripheries can work with slow asynchronous clocks other than bus clocks, asynchronous interrupts from peripheries can wake up chip from Standby mode.

18.1.5 Powerdown Mode

It is the lowest power mode of this chip, most peripheries are power down, but REGFILE, WKU, PCU, RCU, lpTMR, ACMP, SIRC and SXOSC keep working. Most registers other than those working modules and most SRAM's contents lost. NVIC and asynchronous interrupt Wake-up logic are disabled. WKU is used to wake up chip from Powerdown mode.

For modules that is power off during Powerdown mode, Powerdown Wake-up is same as Power on reset.

Yuntu Microelectronics

18.2 Power Modes Table

Table 18.1: Power Mode

Module	Active	Sleep	Deepsleep	Standby	Powerdown
Core module					
NVIC	FF[1]	FF	Static	Static	OFF
DMA	FF	FF	Static	Static	OFF
Power module					
LDO25	FF	FF	FF	RPM[2]	OFF
LDO11	FF	FF	FF	RPM	RPM
Safety and security module					
LVD	FF	FF	FF	FF	FF
HVD	FF	FF	FF	OFF	OFF
WDG	FF	FF	FF	FF	OFF
CMU	FF	FF	FF	FF	OFF
EMU	FF	FF	Static	Static	OFF
INTM	FF	FF	Static	Static	OFF
FMU	FF	FF	FF	Optional FF (SIRC_STDBYEN)	OFF
HCU	FF	FF	Static	Static	OFF
TRNG	FF	FF	Static	Static	OFF
PCRC	FF	FF	Static	Static	OFF
Clocks					
FIRC	Optional FF (FIRC_EN)	Optional FF (FIRC_EN)	Optional FF (FIRC_DSEN)	OFF	OFF
FXOSC	Optional FF (FXOSC_EN)	Optional FF (FXOSC_EN)	Optional FF (FXOSC_DSEN)	OFF	OFF
PLL	Optional FF (PLL_EN)	Optional FF (PLL_EN)	Optional FF (PLL_DSEN)	OFF	OFF
SIRC	FF	FF	Optional FF (SIRC_DSEN)	Optional FF (SIRC_STDBYEN)	Optional FF (SIRC_PDEN)
Core clock	FF	OFF	OFF	OFF	OFF
Platform clock	FF	FF	OFF	OFF	OFF
Fast bus clock	FF	FF	OFF	OFF	OFF
Slow bus clock	FF	FF	OFF	OFF	OFF
Memory and memory interface					
Flash	FF	FF	Optional FF (DPDEN)[3]	Optional FF (DPDEN)	OFF
OCRAM	FF	FF	FF	Static	2x16KB blocks retained[4]
TCM	FF	FF	FF	FF	OFF
ROM	FF	Static	Static	Static	OFF
REGFILE	FF	FF	FF	FF	FF
Communication interfaces					
FlexCAN	FF	FF	Static	Static	OFF
LINFlexD	FF	FF	Static	Static	OFF
QSPI	FF	FF	Static	Static	OFF
SPI	FF	FF	FF	FF	OFF
I2C	FF	FF	FF	FF	OFF
SENT	FF	FF	FF	FF	OFF
SAI	FF	FF	FF	Static	OFF
Timers					

Yuntu Microelectronics

Table 18.1 continued from previous page

Module	Active	Sleep	Deepsleep	Standby	Powerdown
TMR	FF	FF	Static	Static	OFF
pTMR	FF	FF	Static	Static	OFF
lpTMR	FF	FF	FF	FF	FF
eTMR	FF	FF	Static	Static	OFF
MPWM	FF	FF	Static	Static	OFF
RTC	FF	FF	FF	FF	FF
PTU	FF	FF	Static	Static	OFF
TMU	FF	FF	Static	Static	OFF
Analog					
ADC	FF	FF	FF	Static	OFF
ACMP	FF	FF	FF	FF	FF
Human-Machine interfaces					
GPIO	FF	FF	FF	FF	OFF
WKU	Static	Static	Static	Static	FF

1. FF: Full Function
2. RPM: Reduced Power Mode
3. When DPDEN=1, Flash enters its deep Powerdown mode
4. The content of SRAM with below address is retained.
   0x20020000 - 0x20027FFF retained

18.3 Wake-up Sources

Table 18.2: Wakeup Sources

Wake-up resources[1]	Sleep	Deepsleep (RPMEN=0 STANDBYEN=0)	Standby (RPMEN=1 STANDBYEN=1)	Powerdown (RPMEN=1 STANDBYEN=0)	Comment
Reset	Y	Y	Y	Y	2
NMI	Y	Y	Y		
HardFault					
MemManage					
BusFault					
UsageFault					
SVC					
DebugMon	Y				
PendSV					
SysTick	Y				
DMA0	Y				
DMA1	Y				
DMA2	Y				
DMA3	Y				
DMA4	Y				
DMA5	Y				
DMA6	Y				
DMA7	Y				
DMA8	Y				
DMA9	Y				
DMA10	Y				
DMA11	Y				

Yuntu Microelectronics

Wake-up resources[1]	Sleep	Deepsleep (RPMEN=0)	Standby (RPMEN=1 STB_EN=1)	Powerdown (RPMEN=1 PD_EN=0)	Comment
DMA12	Y				
DMA13	Y				
DMA14	Y				
DMA15	Y				
DMA16	Y				
DMA17	Y				
DMA18	Y				
DMA19	Y				
DMA20	Y				
DMA21	Y				
DMA22	Y				
DMA23	Y				
DMA24	Y				
DMA25	Y				
DMA26	Y				
DMA27	Y				
DMA28	Y				
DMA29	Y				
DMA30	Y				
DMA31	Y				
DMA_Error	Y				
FPU	Y				
EFM	Y				
EFM_Error	Y				
PCU	Y	Y	Y		
EFM_ECC	Y				
WDG	Y	Y	Y		
I2C0_Master_Slave	Y	Y	Y		
SPI0	Y	Y	Y		
SPI1	Y	Y	Y		
SPI2	Y	Y	Y		
I2C1_Master_Slave	Y	Y	Y		
LINFlexD0	Y	Y	Y		
LINFlexD1	Y	Y	Y		
LINFlexD2	Y	Y	Y		
QSPI	Y				
ADC0	Y	Y			
ADC1	Y	Y			
ACMP0	Y	Y	Y	Y	3
ACMP1	Y	Y	Y	Y	3
EMU	Y				
RTC	Y	Y	Y	Y	3
RTC_Seconds	Y	Y	Y	Y	3
pTMR0_Ch0	Y				
pTMR0_Ch1	Y				
pTMR0_Ch2	Y				
pTMR0_Ch3	Y				
PTU0	Y				
pTMR1_Ch0	Y				
pTMR1_Ch1	Y				

Yuntu Microelectronics

Wake-up resources[1]	Sleep	Deepsleep (RPMEN=0)	Standby (RPMEN=1 STB_EN=1)	Powerdown (RPMEN=1 PD_EN=0)	Comment
pTMR1_Ch2	Y				
pTMR1_Ch3	Y				
SCU	Y	Y			
lpTMR0	Y	Y	Y	Y	3
GPIOA	Y	Y	Y		
GPIOB	Y	Y	Y		
GPIOC	Y	Y	Y		
GPIOD	Y	Y	Y		
GPIOE	Y	Y	Y		
PTU1	Y				
CAN0_ORed	Y				
CAN0_Error	Y				
CAN0_Wake_Up	Y	Y	Y		
CAN0_ORed_0_15_MB	Y				
CAN0_ORed_16_31_MB	Y				
CAN0_ORed_32_47_MB	Y				
CAN0_ORed_48_63_MB	Y				
CAN0_ORed_64_79_MB	Y				
CAN0_ORed_80_95_MB	Y				
CAN0_ORed_96_111_MB	Y				
CAN0_ORed_112_127_MB	Y				
CAN1_ORed	Y				
CAN1_Error	Y				
CAN1_Wake_Up	Y	Y	Y		
CAN1_ORed_0_15_MB	Y				
CAN1_ORed_16_31_MB	Y				
CAN1_ORed_32_47_MB	Y				
CAN1_ORed_48_63_MB	Y				
CAN1_ORed_64_79_MB	Y				
CAN1_ORed_80_95_MB	Y				
CAN1_ORed_96_111_MB	Y				
CAN1_ORed_112_127_MB	Y				
CAN2_ORed	Y				
CAN2_Error	Y				
CAN2_Wake_Up	Y	Y	Y		
CAN2_ORed_0_15_MB	Y				
CAN2_ORed_16_31_MB	Y				
CAN2_ORed_32_47_MB	Y				
CAN2_ORed_48_63_MB	Y				
eTMR0_Ch0_Ch1	Y				
eTMR0_Ch2_Ch3	Y				
eTMR0_Ch4_Ch5	Y				
eTMR0_Ch6_Ch7	Y				
eTMR0_Fault	Y				
eTMR0_Ovf	Y				
eTMR1_Ch0_Ch1	Y				
eTMR1_Ch2_Ch3	Y				
eTMR1_Ch4_Ch5	Y				
eTMR1_Ch6_Ch7	Y				
eTMR1_Fault	Y				

Yuntu Microelectronics

Wake-up resources[1]	Sleep	Deepsleep (RPMEN=0)	Standby (RPMEN=1 STB_EN=1)	Powerdown (RPMEN=1 PD_EN=0)	Comment
eTMR1_Ovf	Y				
eTMR2_Ch0_Ch1	Y				
eTMR2_Ch2_Ch3	Y				
eTMR2_Ch4_Ch5	Y				
eTMR2_Ch6_Ch7	Y				
eTMR2_Fault	Y				
eTMR2_Ovf	Y				
eTMR3_Ch0_Ch1	Y				
eTMR3_Ch2_Ch3	Y				
eTMR3_Ch4_Ch5	Y				
eTMR3_Ch6_Ch7	Y				
eTMR3_Fault	Y				
eTMR3_Ovf	Y				
eTMR4_Ch0_Ch1	Y				
eTMR4_Ch2_Ch3	Y				
eTMR4_Ch4_Ch5	Y				
eTMR4_Ch6_Ch7	Y				
eTMR4_Fault	Y				
eTMR4_Ovf	Y				
eTMR5_Ch0_Ch1	Y				
eTMR5_Ch2_Ch3	Y				
eTMR5_Ch4_Ch5	Y				
eTMR5_Ch6_Ch7	Y				
eTMR5_Fault	Y				
eTMR5_Ovf	Y				
SENT0	Y	Y	Y		
SENT1	Y	Y	Y		
FMU_Alarm	Y	Y	Y		
FMU_MISC	Y	Y	Y		
TRNG	Y				
HCU	Y				
INTM	Y				
TMR0	Y				
LINFlexD3	Y				
LINFlexD4	Y				
LINFlexD5	Y				
I2C2_Master_Slave	Y	Y	Y		
SPI3	Y	Y	Y		
SPI4	Y	Y	Y		
SPI5	Y	Y	Y		
CAN3_ORed	Y				
CAN3_Error	Y				
CAN3_Wake_Up	Y	Y	Y		
CAN3_ORed_0_15_MB	Y				
CAN3_ORed_16_31_MB	Y				
CAN3_ORed_32_47_MB	Y				
CAN3_ORed_48_63_MB	Y				
CAN4_ORed	Y				
CAN4_Error	Y				
CAN4_Wake_Up	Y	Y	Y		

Yuntu Microelectronics

Wake-up resources[1]	Sleep	Deepsleep (RPMEN=0)	Standby (RPMEN=1 STB_EN=1)	Powerdown (RPMEN=1 PD_EN=0)	Comment
CAN4_ORed_0_15_MB	Y				
CAN4_ORed_16_31_MB	Y				
CAN4_ORed_32_47_MB	Y				
CAN4_ORed_48_63_MB	Y				
CAN5_ORed	Y				
CAN5_Error	Y				
CAN5_Wake_Up	Y	Y	Y		
CAN5_ORed_0_15_MB	Y				
CAN5_ORed_16_31_MB	Y				
WKU				Y	4
LINFlexD6	Y				
LINFlexD7	Y				
LINFlexD8	Y				
LINFlexD9	Y				
SPI6	Y	Y	Y		
SPI7	Y	Y	Y		
I2C3_Master_Slave	Y	Y	Y		
I2C4_Master_Slave	Y	Y	Y		
CAN6_ORed	Y				
CAN6_Error	Y				
CAN6_Wake_Up	Y	Y	Y		
CAN6_ORed_0_15_MB	Y				
CAN6_ORed_16_31_MB	Y				
TMR1	Y				
CORE_CTI					
SAI0	Y	Y			
SAI1	Y	Y			
CAN7_ORed	Y				
CAN7_Error	Y				
CAN7_Wake_Up	Y	Y	Y		
CAN7_ORed_0_15_MB	Y				
CAN7_ORed_16_31_MB	Y				
MPWM0_Ch0_Ch7	Y				
MPWM0_Ch8_Ch15	Y				
MPWM1_Ch0_Ch7	Y				
MPWM1_Ch8_Ch15	Y				
MPWM2_Ch0_Ch7	Y				
MPWM2_Ch8_Ch15	Y				

1. The item of the first column is consistent with the interrupt of "Table 10.1".
2. "Y" means specified IRQ can wake up MCU in corresponding power mode.
3. RTC, ACMP and lpTMR interrupt Wake-up is required to enable in WKU module.
4. As for WKU module wake up in Powerdown mode, please see detail information in WKU chapter.

Yuntu Microelectronics

Chapter 19

Wake-up Unit (WKU)

19.1 Introduction

This chapter describe the function of wake-up unit, which works in powerdown mode to handle the wake-up events from wake-up pin or reset pin. WKU is disabled in other power modes.

19.1.1 Features

- Support for up to 64 external input pins and up to 4 internal modules with individual enable bits for MCU interrupt from Powerdown mode.
- Input sources may be external pins or from internal modules capable of running in Powerdown mode.
- External input pins programmable for falling-edge, rising-edge, or any-edge detection.
- Support to enable external input pin and filter detection in Powerdown mode.
- Optional digital filters provided to qualify an external pin detect. Note that when the SIRC and SXOSC clock is disabled, filters are disabled and bypassed.

19.1.2 Wake-up pins introduction

Table 19.1: Wake-up Pins

Wake-up pins	Pin name
WKU[0]	PTE_11
WKU[1]	PTE_10
WKU[2]	PTE_5
WKU[3]	PTE_4
WKU[4]	PTE_3
WKU[5]	PTE_1
WKU[6]	PTE_0
WKU[7]	PTE_9
WKU[8]	PTE_8
WKU[9]	PTC_3
WKU[10]	PTB_26
WKU[11]	PTD_5
WKU[12]	PTC_1
WKU[13]	PTC_0
WKU[14]	PTC_15

Yuntu Microelectronics

Wake-up pins	Pin name
WKU[15]	PTC_14
WKU[16]	PTB_1
WKU[17]	PTB_0
WKU[18]	PTA_7
WKU[19]	PTA_6
WKU[20]	PTB_13
WKU[21]	PTB_12
WKU[22]	PTD_3
WKU[23]	PTD_2
WKU[24]	PTA_3
WKU[25]	PTA_2
WKU[26]	PTC_7
WKU[27]	PTC_6
WKU[28]	PTE_6
WKU[29]	PTE_2
WKU[30]	PTA_13
WKU[31]	PTA_12
WKU[32]	PTE_16
WKU[33]	PTE_15
WKU[34]	PTA_22
WKU[35]	PTE_14
WKU[36]	PTD_15
WKU[37]	PTD_13
WKU[38]	PTB_21
WKU[39]	PTB_23
WKU[40]	PTB_2
WKU[41]	PTC_25
WKU[42]	PTC_11
WKU[43]	PTC_9
WKU[44]	PTB_17
WKU[45]	PTD_4
WKU[46]	PTB_9
WKU[47]	PTB_8
WKU[48]	PTA_15
WKU[49]	PTE_23
WKU[50]	PTE_25
WKU[51]	PTA_9
WKU[52]	PTA_8
WKU[53]	PTB_16
WKU[54]	PTB_15
WKU[55]	PTA_1
WKU[56]	PTA_16
WKU[57]	PTA_25
WKU[58]	PTA_30
WKU[59]	PTB_28
WKU[60]	PTC_18
WKU[61]	PTA_26
WKU[62]	PTC_23
WKU[63]	PTB_11

Yuntu Microelectronics

19.2 Register Definition

This subsection provides a description of the WKU module memory map and registers.

19.2.1 WKU Memory Map

WKU base address 40188000h

Table 19.2: WKU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Pin Control Register (PCR0)	8	RW	00h
1h	Pin Control Register (PCR1)	8	RW	00h
2h	Pin Control Register (PCR2)	8	RW	00h
3h	Pin Control Register (PCR3)	8	RW	00h
4h	Pin Control Register (PCR4)	8	RW	00h
5h	Pin Control Register (PCR5)	8	RW	00h
6h	Pin Control Register (PCR6)	8	RW	00h
7h	Pin Control Register (PCR7)	8	RW	00h
8h	Pin Control Register (PCR8)	8	RW	00h
9h	Pin Control Register (PCR9)	8	RW	00h
ah	Pin Control Register (PCR10)	8	RW	00h
bh	Pin Control Register (PCR11)	8	RW	00h
ch	Pin Control Register (PCR12)	8	RW	00h
dh	Pin Control Register (PCR13)	8	RW	00h
eh	Pin Control Register (PCR14)	8	RW	00h
fh	Pin Control Register (PCR15)	8	RW	00h
10h	Pin Control Register (PCR16)	8	RW	00h
11h	Pin Control Register (PCR17)	8	RW	00h
12h	Pin Control Register (PCR18)	8	RW	00h
13h	Pin Control Register (PCR19)	8	RW	00h
14h	Pin Control Register (PCR20)	8	RW	00h
15h	Pin Control Register (PCR21)	8	RW	00h
16h	Pin Control Register (PCR22)	8	RW	00h
17h	Pin Control Register (PCR23)	8	RW	00h
18h	Pin Control Register (PCR24)	8	RW	00h
19h	Pin Control Register (PCR25)	8	RW	00h
1ah	Pin Control Register (PCR26)	8	RW	00h
1bh	Pin Control Register (PCR27)	8	RW	00h
1ch	Pin Control Register (PCR28)	8	RW	00h
1dh	Pin Control Register (PCR29)	8	RW	00h
1eh	Pin Control Register (PCR30)	8	RW	00h
1fh	Pin Control Register (PCR31)	8	RW	00h
20h	Pin Control Register (PCR32)	8	RW	00h
21h	Pin Control Register (PCR33)	8	RW	00h
22h	Pin Control Register (PCR34)	8	RW	00h
23h	Pin Control Register (PCR35)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
24h	Pin Control Register (PCR36)	8	RW	00h
25h	Pin Control Register (PCR37)	8	RW	00h
26h	Pin Control Register (PCR38)	8	RW	00h
27h	Pin Control Register (PCR39)	8	RW	00h
28h	Pin Control Register (PCR40)	8	RW	00h
29h	Pin Control Register (PCR41)	8	RW	00h
2ah	Pin Control Register (PCR42)	8	RW	00h
2bh	Pin Control Register (PCR43)	8	RW	00h
2ch	Pin Control Register (PCR44)	8	RW	00h
2dh	Pin Control Register (PCR45)	8	RW	00h
2eh	Pin Control Register (PCR46)	8	RW	00h
2fh	Pin Control Register (PCR47)	8	RW	00h
30h	Pin Control Register (PCR48)	8	RW	00h
31h	Pin Control Register (PCR49)	8	RW	00h
32h	Pin Control Register (PCR50)	8	RW	00h
33h	Pin Control Register (PCR51)	8	RW	00h
34h	Pin Control Register (PCR52)	8	RW	00h
35h	Pin Control Register (PCR53)	8	RW	00h
36h	Pin Control Register (PCR54)	8	RW	00h
37h	Pin Control Register (PCR55)	8	RW	00h
38h	Pin Control Register (PCR56)	8	RW	00h
39h	Pin Control Register (PCR57)	8	RW	00h
3ah	Pin Control Register (PCR58)	8	RW	00h
3bh	Pin Control Register (PCR59)	8	RW	00h
3ch	Pin Control Register (PCR60)	8	RW	00h
3dh	Pin Control Register (PCR61)	8	RW	00h
3eh	Pin Control Register (PCR62)	8	RW	00h
3fh	Pin Control Register (PCR63)	8	RW	00h
40h	Pin Flag Register (PFR0)	32	RW	00000000h
44h	Pin Flag Register (PFR1)	32	RW	00000000h
48h	Reset Pin Control Register (RPCR)	32	RW	00000000h
4ch	Module Enable Register (MER)	32	RW	00000000h

Yuntu Microelectronics

19.2.1.1 WKU PCR Register

This register includes the field to enable and select the edge detect type for the external wake-up input pins.

Offset: 0h + n * 1h

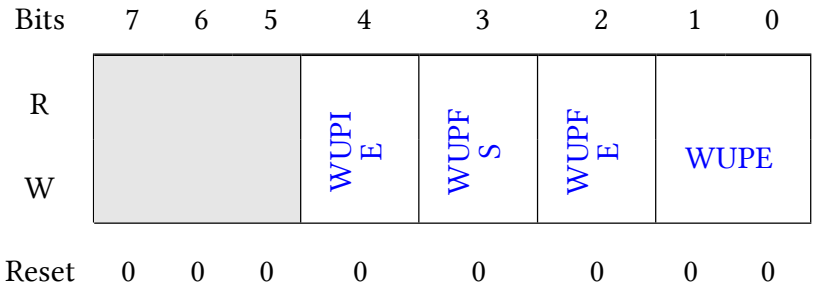


Table 19.3: WKU PCR Register Description

Field	Function
4 WUPIE	**Wake-up Pin Interrupt Enable** Interrupt should always be enabled when pin selected as wake-up source 0b - wake-up pin interrupt is disabled. 1b - wake-up pin interrupt is enabled.
3 WUPFS	**Wake-up Pin Filter Clock Select** Select wake-up pin filter clock source. 0b - SIRC clock is selected as wake-up in filter clock source. 1b - SXOSC clock is selected as wake-up in filter clock source.
2 WUPFE	**Wake-up Pin Filter Enable** Enables the digital filter for the external pin detect. 0b - Filter disabled. 1b - Filter enabled.
1 - 0 WUPE	**Wake-up Pin Enable For WKU_Pn** Enables and configures the edge detection for the wake-up pin. 00b - External input pin disabled as wake-up input 01b - External input pin enabled with rising edge detection 10b - External input pin enabled with falling edge detection 11b - External input pin enabled with rising or falling edge detection

Yuntu Microelectronics

19.2.1.2 WKU PFR Register

The PFR register contains the wake-up flags indicating which wake-up source caused the MCU to exit Powerdown mode. The external wake-up flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFn bit. The wake-up flag (WUFn), if set, will remain set if the associated WUPEn bit is cleared.

Offset: 40h + n * 4h

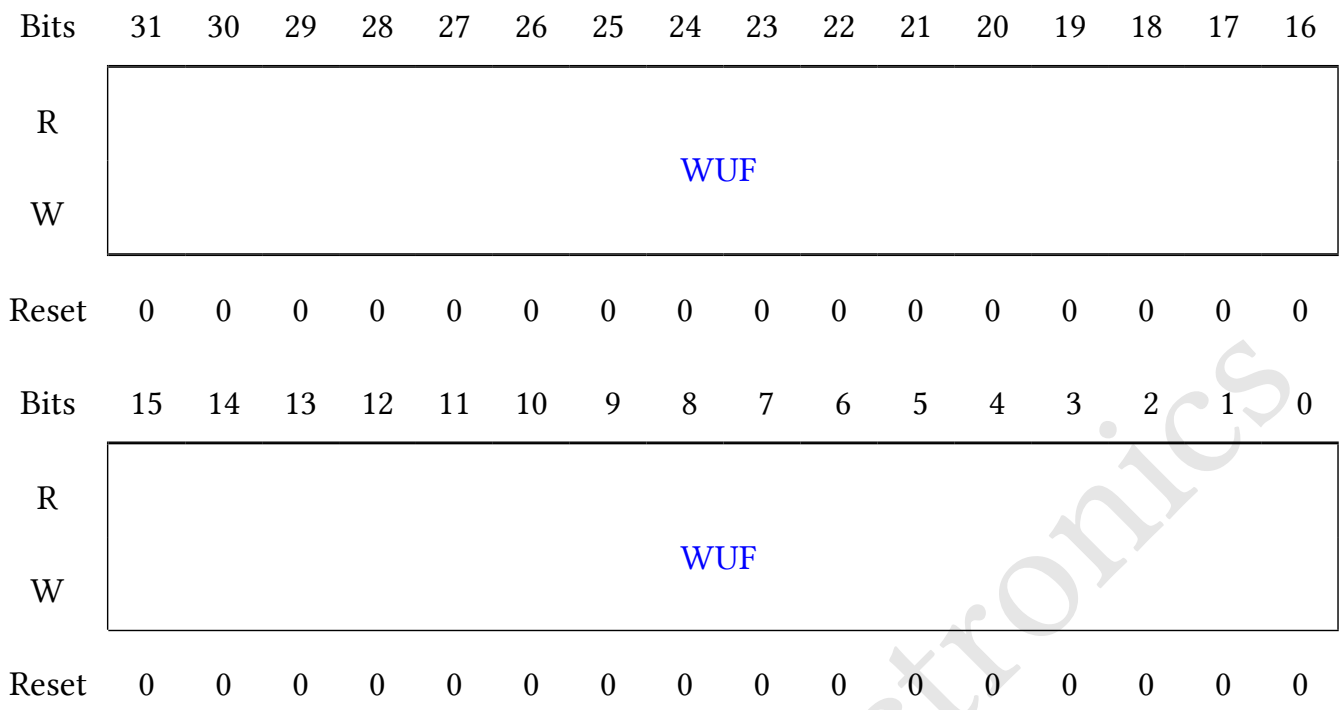


Table 19.4: WKU PFR Register Description

Field	Function
31 - 0 WUF	**Wake-up Flag for WKU_Pin** Indicates that an enabled external wake-up pin was a source of exiting a Powerdown mode. To clear the flag write a one to WUFn. 0b - WKU_Pn input was not a wake-up source 1b - WKU_Pn input was a wake-up source

Yuntu Microelectronics

19.2.1.3 WKU RPCR Register

This register controls the reset pin filter functions.

Offset: 48h

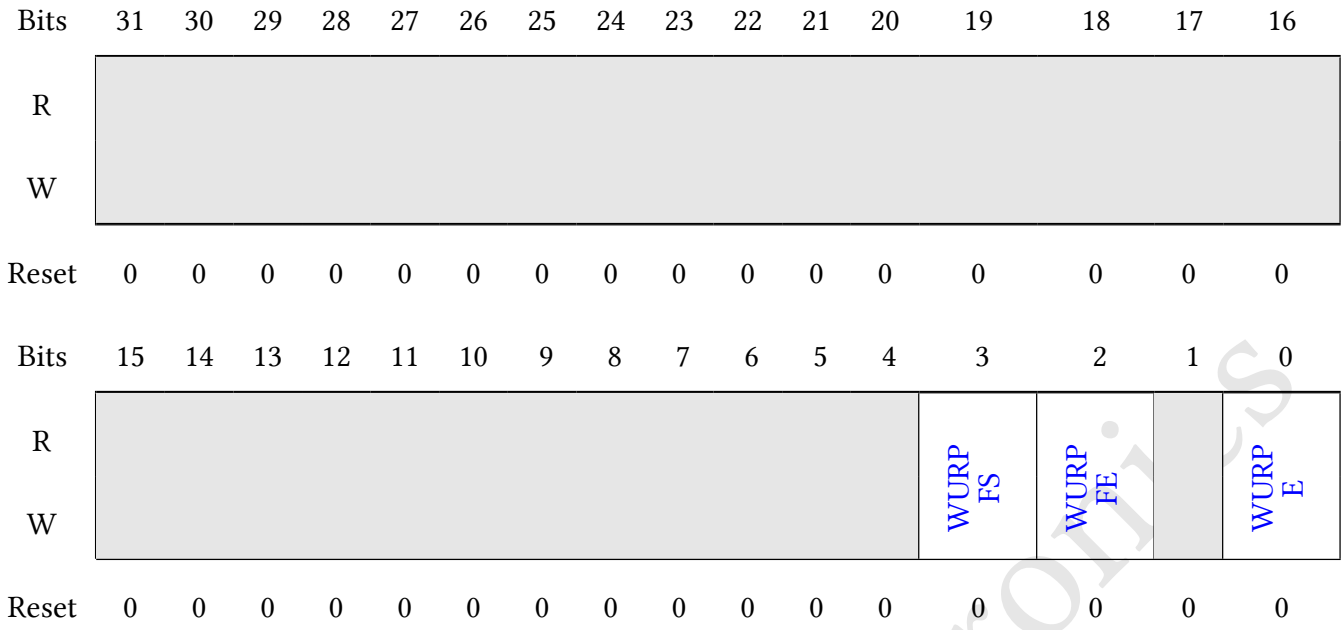


Table 19.5: WKU RPCR Register Description

Field	Function
3 WURPFS	**Reset Pin Filter Clock Select** Select Reset pin filter clock source. 0b - SIRC clock is selected as Reset pin filter clock source. 1b - SXOSC clock is selected as Reset pin filter clock source.
2 WURPFE	**Reset Pin Filter Enable** Enable bit of Reset pin filter, this bit should be set if filter need to be used. 0b - Disable reset pin filter 1b - Enable reset pin filter
0 WURPE	**Reset Pin Enable** Enable Reset pin as a wake-up source input. 0b - Reset pin not used as wake-up source 1b - Reset pin used as wake-up source

Yuntu Microelectronics

19.2.1.4 WKU MER Register

This register contains the bits to enable an internal module interrupt as a wake-up source.

Offset: 4ch

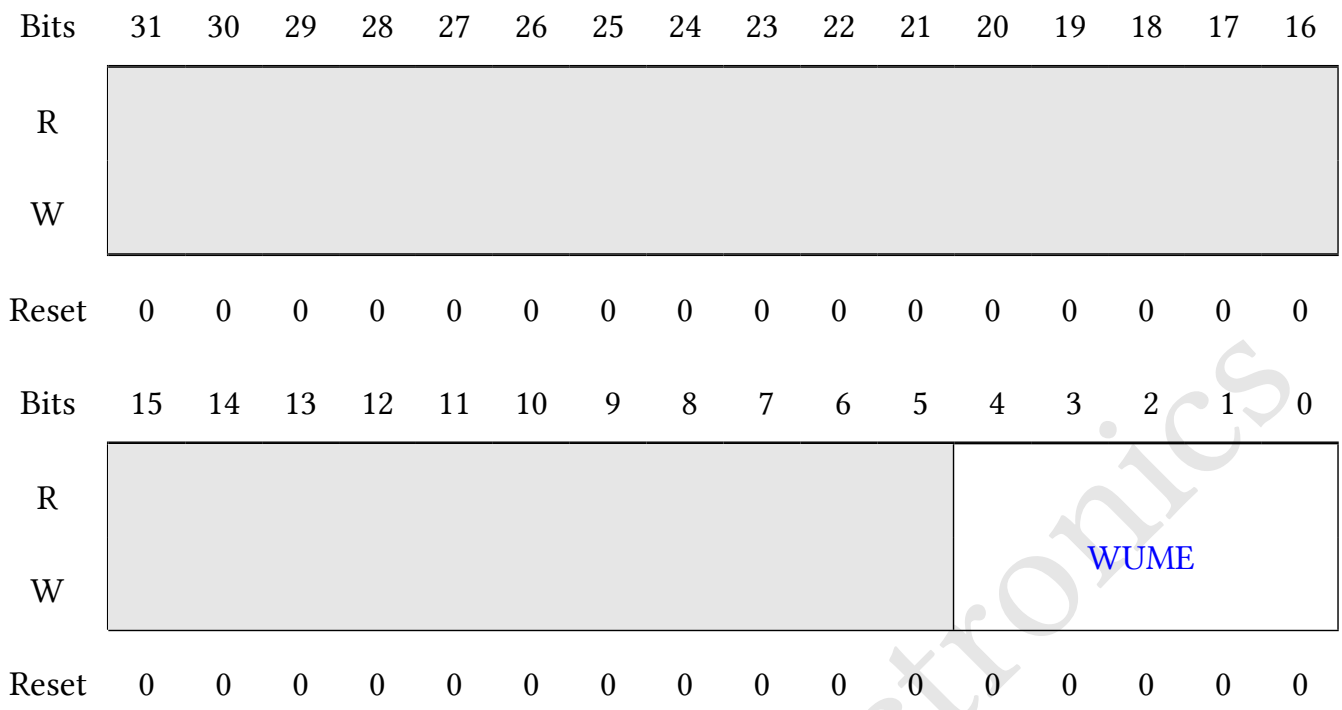


Table 19.6: WKU MER Register Description

Field	Function
4 - 0 WUME	**Wake-up Module Enable bits** 0b00001 - RTC_IRQ 0b00010 - RTC_Seconds_IRQ 0b00100 - lpTMR0_IRQ 0b01000 - ACMP0_IRQ 0b10000 - ACMP1_IRQ

Yuntu Microelectronics

Part VI

Communication Interfaces

Yuntu Microelectronics

Chapter 20

Flexible Controller Area Network (FlexCAN)

20.1 Chip-specific FlexCAN Information

20.1.1 Instantiation Information

This chip contains up to 8 instances of FlexCAN, main configuration as below table:

Table 20.1: CAN Instance Information

Instance	CAN0	CAN1	CAN2	CAN3	CAN4	CAN5	CAN6	CAN7
Number of MBs	128	128	64	64	64	32	32	32
PNET	NO	NO	YES	NO	NO	NO	NO	NO
HR TIMESTAMP	YES	YES	YES	YES	YES	NO	NO	NO
ERF (Enhanced RX FIFO)	YES	YES	YES	YES	YES	NO	NO	NO
ERF Size	32	32	32	20	20	-	-	-
ERF filter ELEM	128	128	128	128	128	-	-	-

NOTE: All instances support:

- IRMQ
- ECC
- DMA
- FD_EN
- CIA bit timing
- External Tick Time
- RX FIFO Filtering
- not support Enhanced MB MEM

NOTE: See attached interrupt map file for the vector numbers for the CAN instances and their corresponding interrupt sources.

20.2 Introduction

The FlexCAN module is a communication controller implementing the CAN protocol according to the ISO 11898-1 standard and CAN Specification 2.0 Part B. Figure: “Block Diagram of CAN” shows a

Yuntu Microelectronics

general block diagram of the FlexCAN, which includes one associated memory for storing message buffers (MBs), Receive Global Mask registers, Receive Individual Mask registers, Receive (Rx) FIFO filters, and Rx FIFO ID filters.

NOTE: The Legacy Rx FIFOs cannot be used in CAN FD mode. The Enhanced Rx FIFO, if implemented, can be used in CAN FD mode.

20.2.1 Block Diagram

The following figure shows the FlexCAN block diagram.

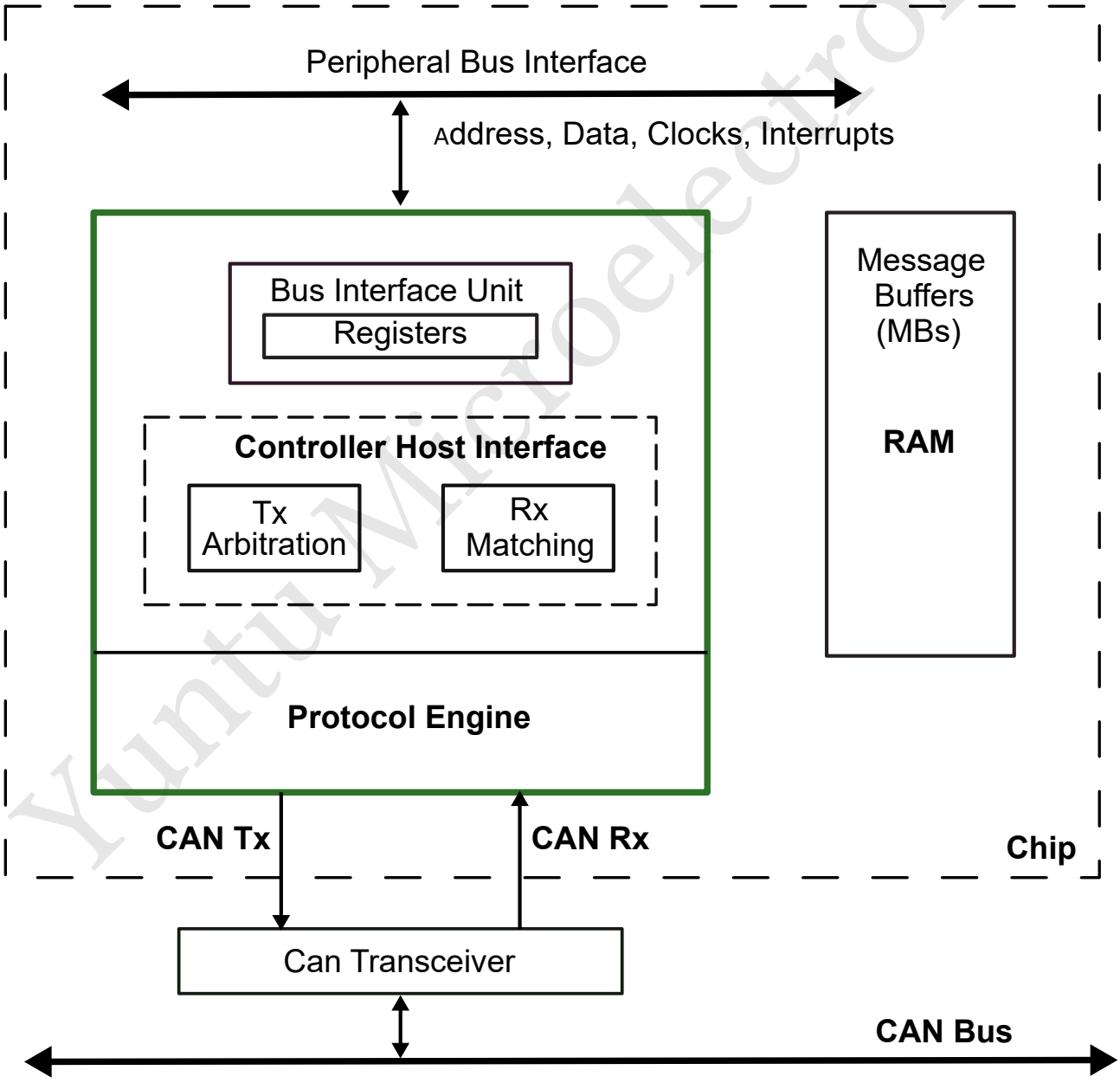


Figure 20.1: Block Diagram of CAN

Yuntu Microelectronics

20.2.2 Overview

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific requirements of this field, including:

- Real-time processing
- Reliable operation in the EMI environment of a vehicle
- Cost-effectiveness
- Required bandwidth

The FlexCAN module is a full implementation of the CAN protocol specification, the CAN with Flexible Data rate (CAN FD) protocol, and the CAN 2.0 Part B protocol, supporting both standard and extended message frames and long payloads up to 64 bytes, transferred at rates up to 8 Mbps. The message buffers are stored in an embedded RAM dedicated to the FlexCAN module.

The Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- Requesting RAM access for receiving and transmitting message frames
- Validating received messages
- Performing error handling
- Detecting CAN FD messages

The Controller Host Interface (CHI) submodule handles message buffer selection for reception and transmission, including arbitration and ID matching algorithms for both CAN FD and non-CAN FD message formats.

The Bus Interface Unit (BIU) submodule controls the access to and from the internal interface bus to establish connection to the CPU and to other blocks. Clocks, address and data buses, interrupt outputs, DMA, and test signals are accessed through the BIU.

20.2.3 Features

Features of the FlexCAN module include:

- Full implementation of the CAN FD protocol and CAN Specification 2.0, Part B
  - Standard data frames
  - Extended data frames
  - Zero to sixty-four bytes data length
  - Programmable bit rate
  - Content-related addressing
- Compliant with the ISO 11898-1 standard
- Silicon-proven implementation passing ISO 16845-1:2016 CAN conformance tests
- Flexible mailboxes configurable to store 0 to 8, 16, 32, or 64 bytes data length
- Each mailbox configurable as receive or transmit, all supporting standard and extended messages
- Individual Rx Mask registers per mailbox
- Full-featured Legacy Rx FIFO with storage capacity for up to 6 CAN frames and automatic internal pointer handling with DMA support
- Full-featured Enhanced Rx FIFO with storage capacity for up to 32 CAN FD frames and automatic internal pointer handling with DMA support

Yuntu Microelectronics

- Transmission abort capability
- Flexible message buffers, totaling 128 message buffers of 8 bytes data length each, configurable as Rx or Tx
- Programmable clock source to the CAN Protocol Engine, either peripheral clock or oscillator clock
- RAM not used by reception or transmission structures can be used as general purpose RAM space
- Listen-Only mode capability
- Programmable Loop-Back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Time stamp based on 16-bit free-running timer, with an optional external time tick or high-resolution 32-bit on-chip timer
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low-power mode, with programmable Wake Up on bus activity or matching with received frames (Pretended Networking)
- Transceiver Delay Compensation feature when transmitting CAN FD messages at faster data rates
- Remote request frames may be managed automatically or by software
- CAN bit time settings and configuration bits can only be written in Freeze mode
- Tx mailbox status (lowest priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit available in Error in Status 1 register to indicate that the FlexCAN is synchronous with CAN bus
- CRC status for transmitted message
- Legacy Rx FIFO Global Mask register
- Selectable priority between mailboxes and Rx FIFO during matching process
- Powerful Legacy Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard, or 512 partial (8 bit) IDs, with up to 32 ID Filter Table elements
- Powerful Enhanced Rx FIFO ID filtering, capable of matching incoming IDs against either 64 extended or 128 standard ID filter elements with three filtering schemes: mask + filter, range, and two filters without mask
- 100% backward compatibility with previous FlexCAN version
- Supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to 5 parity bits. The error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors) and errors in 2 bits can be detected but not corrected (non-correctable errors).
- Supports Pretended Networking functionality in low-power mode: Doze mode, Deepsleep mode

Yuntu Microelectronics

20.2.4 Modes of Operation

The FlexCAN module has the following functional modes:

- Normal Mode (User or Supervisor):

  In Normal mode, the FlexCAN operates receiving and/or transmitting message frames, errors are managed normally, and all CAN protocol functions are enabled.

  User and Supervisor modes differ in the access to some restricted control registers.

- Freeze Mode:

  Freeze mode is enabled when the CAN_MCR.FRZ bit is set. If enabled, Freeze mode is entered when CAN_MCR.HALT is set or when Debug mode is requested at chip level and CAN_MCR.FRZACK is set by the FlexCAN module. In Freeze mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost.

  See "Freeze Mode" for more information.

- Loop-Back Mode:

  The FlexCAN enters Loop-Back mode when CAN_CTRL1.LPB is asserted. In LoopBack mode, the FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). The FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In Loop-Back mode, the FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

- Listen-Only Mode:

  The FlexCAN enters Listen-Only mode when CAN_CTRL1.LOM is asserted. In Listen-Only mode, transmission is disabled, all error counters are frozen, and the FlexCAN operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station are received. If the FlexCAN detects a message that has not been acknowledged, it flags a BIT0 error (without changing the receive error counter), as if it were trying to acknowledge the message.

- CAN FD Active Mode:

  In CAN FD Active mode, the FlexCAN is capable of transmitting and receiving all messages formatted according to the CAN FD Protocol and CAN 2.0 Protocol 2.0 in an interleaved fashion. The CPU can set the FlexCAN into CAN FD Active mode by setting the MCR.FDEN when the FlexCAN is in Freeze Mode.

  Below table summarizes the differences in FlexCAN feature availability in CAN FD Active mode (CAN_MCR.FDEN = 1) and Classical CAN mode (CAN_MCR.FDEN = 0).

Table 20.2: CAN Feature Availability in Classic CAN and CAN FD Modes

Feature	Classical CAN	CAN FD
Legacy RX FIFO	YES	NO
Legacy RX FIFO DMA	YES	NO
Enhanced RX FIFO	YES	YES
Enhanced RX FIFO DMA	YES	YES
Pretended Networking	YES	NO

Yuntu Microelectronics

For low-power operation, the FlexCAN module has:

- Module Disable Mode:

  Module Disable mode is entered when CAN_MCR.MDIS is asserted by the CPU and CAN_MCR.LPMACK is asserted by the FlexCAN. When disabled, the FlexCAN issues a request to disable the clocks to the PE and CHI submodules. Exit from Module Disable mode is done by negating CAN_MCR.MDIS.

  See "Module Disable Mode" for more information.

- Doze Mode:

  Doze mode is entered when CAN_MCR.DOZE is asserted, Doze mode is requested at the chip level, and CAN_MCR.LPMACK is asserted by the FlexCAN. When in Doze mode, the FlexCAN issues a request to disable the clocks to the PE and the CHI submodules. Exit from Doze mode occurs when CAN_MCR.DOZE is negated, when the chip is removed from Doze mode, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled.

  See "Doze Mode" for more information.

- Deepsleep Mode:

  Deepsleep mode is entered when Deepsleep mode is requested at chip level and CAN_MCR.LPMACK is asserted by the FlexCAN. When in Deepsleep Mode, the FlexCAN puts itself in an inactive state, then informs the CPU that the clocks can be shut down globally. Exit from Deepsleep mode occurs when the Deepsleep mode request is removed, or when activity is detected on the CAN bus and the Self Wake Up mechanism is enabled.

  See "Deepsleep Mode" for more information.

- Pretended Networking Mode:

  Pretended Networking can be selected to operate together with Doze mode or Deepsleep mode. Before entering Doze mode or Deepsleep mode, CAN_MCR.PNET_EN must be asserted. Once in low-power mode, the CHI submodule clocks are shut down and the PE submodule is kept clocked, so that the receive process is still active to filter incoming messages (see "Receive Process under Pretended Networking Mode") as defined by the configuration registers (see "Pretended Networking Control 1 Register (CAN_CTRL1_PN)". Upon detecting a wakeup event, a Wake Up interrupt is issued to the system. When CAN_MCR.PNET_EN is asserted, the CPU must disable the Self Wake Up feature by negating CAN_MCR.SLFWAK.

20.3 Register Definition

The address space occupied by the FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address offset 0x0080.

Each register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV field in the CAN_MCR register. These registers are identified as S/U in the Access column of Table: "CAN register access and reset information" .

NOTE: An invalid register access results in a bus error. This includes reading a write-only register, writing a read-only register, or accessing an invalid address.

Yuntu Microelectronics

20.3.1 Register Access and Reset Information

Yuntu Microelectronics

Table 20.3: CAN Register Access and Reset Information

Register	Access Type	Hard Reset	Soft Reset
Module Configuration Register (CAN_MCR)	S	Yes	Yes
Control 1 Register (CAN_CTRL1)	S/U	Yes	No
Free Running Timer Register (CAN_TIMER)	S/U	Yes	Yes
Rx Mailboxes Global Mask Register (CAN_RXMGMASK)	S/U	No	No
Rx 14 Mask Register (CAN_RX14MASK)	S/U	No	No
Rx 15 Mask Register (CAN_RX15MASK)	S/U	No	No
Error Counter Register (CAN_ECR)	S/U	Yes	Yes
Error and Status 1 Register (CAN_ESR1)	S/U	Yes	Yes
Interrupt Masks 2 Register (CAN_IMASK2)	S/U	Yes	Yes
Interrupt Masks 1 Register (CAN_IMASK1)	S/U	Yes	Yes
Interrupt Flags 2 Register (CAN_IFLAG2)	S/U	Yes	Yes
Interrupt Flags 1 Register (CAN_IFLAG1)	S/U	Yes	Yes
Control 2 Register (CAN_CTRL2)	S/U	Yes	No
Error and Status 2 Register (CAN_ESR2)	S/U	Yes	Yes
CRC Register (CAN_CRCR)	S/U	Yes	Yes
Legacy Rx FIFO Global Mask Register (CAN_RXFGMASK)	S/U	No	No
Legacy Rx FIFO Information Register (CAN_RXFIR)	S/U	No	No
CAN Bit Timing Register (CAN_CBT)	S/U	Yes	No
Interrupt Masks 4 Register (CAN_IMASK4)	S/U	Yes	Yes
Interrupt Masks 3 Register (CAN_IMASK3)	S/U	Yes	Yes
Interrupt Flags 4 Register (CAN_IFLAG4)	S/U	Yes	Yes
Interrupt Flags 3 Register (CAN_IFLAG3)	S/U	Yes	Yes
Message buffers	S/U	No	No
Rx Individual Mask Registers (CAN_RXIMRn)	S/U	No	No
Memory Error Control Register (CAN_MECR)	S/U	Yes	Yes
Error Injection Address Register (CAN_ERRIAR)	S/U	Yes	Yes
Error Injection Data Pattern Register (CAN_ERRIDPR)	S/U	Yes	Yes
Error Injection Parity Pattern Register (CAN_ERRIPPR)	S/U	Yes	Yes
Error Report Address Register (CAN_RERRAR)	S/U	Yes	Yes
Error Report Data Register (CAN_RERRDR)	S/U	Yes	Yes
Error Report Syndrome Register (CAN_RERRSYNR)	S/U	Yes	Yes
Error Status Register (CAN_ERRSR)	S/U	Yes	Yes
Pretended Networking Control 1 Register (CAN_CTRL1_PN)	S/U	Yes	Yes
Pretended Networking Control 2 Register (CAN_CTRL2_PN)	S/U	Yes	Yes

Yuntu Microelectronics

Table 20.3 continued from previous page

Register	Access Type	Hard Reset	Soft Reset
Pretended Networking Wake Up Match Register (CAN_WU_MTC)	S/U	Yes	Yes
Pretended Networking ID Filter 1 Register (CAN_FLT_ID1)	S/U	Yes	Yes
Pretended Networking DLC Filter Register (CAN_FLT_DLC)	S/U	Yes	Yes
Pretended Networking Payload Low Filter 1 Register (CAN_PL1_LO)	S/U	Yes	Yes
Pretended Networking Payload High Filter 1 Register (CAN_PL1_HI)	S/U	Yes	Yes
Pretended Networking ID Filter 2 Register / ID Mask Register(CAN_FLT_ID2_IDMASK)	S/U	Yes	Yes
Pretended Networking Payload Low Filter 2 Register / Payload Low Mask Register (CAN_PL2_PLMASK_LO)	S/U	Yes	Yes
Pretended Networking Payload High Filter 2 High Order Bits / Payload High Mask Register (CAN_PL2_PLMASK_HI)	S/U	Yes	Yes
Pretended Networking Wake Up Message Buffer 0 registers (CAN_WMB0_*)	S/U	Yes	No
Pretended Networking Wake Up Message Buffer 1 registers (CAN_WMB1_*)	S/U	Yes	No
Pretended Networking Wake Up Message Buffer 2 registers (CAN_WMB2_*)	S/U	Yes	No
Pretended Networking Wake Up Message Buffer 3 registers (CAN_WMB3_*)	S/U	Yes	No
Enhanced CAN Bit Timing Prescalers Register (CAN_EPRS)	S/U	Yes	No
Enhanced Nominal CAN Bit Timing Register (CAN_ENCBT)	S/U	Yes	No
Enhanced Data Phase CAN Bit Timing Register (CAN_EDCBT)	S/U	Yes	No
Enhanced Transceiver Delay Compensation Register (CAN_ETDC)	S/U	Yes	No
CAN FD Control Register (CAN_FDCTRL)	S/U	Yes	No
CAN FD Bit Timing Register (CAN_FDCBT)	S/U	Yes	No
CAN FD CRC Register (CAN_FDCRC)	S/U	Yes	Yes
Enhanced Rx FIFO Control Register (CAN_ERFCR)	S/U	Yes	Yes
Enhanced Rx FIFO Interrupt Enable Register (CAN_ERFIER)	S/U	Yes	Yes
Enhanced Rx FIFO Status Register (CAN_ERFSR)	S/U	Yes	Yes
High Resolution Time Stamp Registers (CAN_HR_TIME_STAMPn)	S/U	No	No
Enhanced Rx FIFO Filter Element Registers (CAN_ERFFELn)	S/U	No	No

Yuntu Microelectronics

20.3.2 Memory Mapping

The FlexCAN module can store CAN messages for transmission and reception using message buffers and Rx FIFO structures.

The address range from offset 0x80 to 0x87F allocates the one-hundred-twenty-eight 128-bit message buffers (MBs).

The address range from offset 0x1000 to 0x17FF allocates an extra message buffer space that can be used for CAN FD operation.

The message buffer memory map is described in "FlexCAN Message Buffer Memory Map".

The address range from offset 0x2000 to 0x204C allocates the Enhanced Rx FIFO output. The address range from offset 0x2050 to 0x29FC allocates the rest of Enhanced RX FIFO 31 elements. The memory map for the Enhanced Rx FIFO is described in "Enhanced Rx FIFO Structure".

20.3.3 CAN Memory Map

CAN0 base address 400d0000h
CAN1 base address 400d4000h
CAN2 base address 400d8000h
CAN3 base address 400dc000h
CAN4 base address 400e0000h
CAN5 base address 400e4000h
CAN6 base address 400e8000h
CAN7 base address 400ec000h

Table 20.4: CAN Memory Map

Offset	Register	Width	Access	Reset Value
0h	Module Configuration Register (MCR)	32	RW	D890000Fh
4h	Control 1 Register (CTRL1)	32	RW	00000000h
8h	Free Running Timer Register (TIMER)	32	RW	00000000h
10h	Rx Mailboxes Global Mask Register (RXMGMASK)	32	RW	00000000h
14h	Rx 14 Mask Register (RX14MASK)	32	RW	00000000h
18h	Rx 15 Mask Register (RX15MASK)	32	RW	00000000h
1ch	Error Counter Register (ECR)	32	RW	00000000h
20h	Error and Status Register (ESR1)	32	RW	00000000h
24h	Interrupt Mask 2 Register (IMASK2)	32	RW	00000000h
28h	Interrupt Mask 1 Register (IMASK1)	32	RW	00000000h
2ch	Interrupt Flag 2 Register (IFLAG2)	32	RW	00000000h
30h	Interrupt Flag 1 Register (IFLAG1)	32	RW	00000000h
34h	Control 2 Register (CTRL2)	32	RW	00000000h
38h	Error and Status 2 Register (ESR2)	32	RW	00000000h
44h	CRC Register (CRCR)	32	RW	00000000h
48h	Legacy Rx FIFO Global Mask Register (RXFGMASK)	32	RW	00000000h
4ch	Legacy Rx FIFO Information Register (RXFIR)	32	RW	00000000h
50h	Bit Timing Register (CBT)	32	RW	00000000h
68h	Interrupt Mask 4 Register (IMASK4)	32	RW	00000000h
6ch	Interrupt Mask 3 Register (IMASK3)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
70h	Interrupt Flags 4 Register (IFLAG4)	32	RW	00000000h
74h	Interrupt Flags 3 Register (IFLAG3)	32	RW	00000000h
80h	Embedded RAM (RAM0)	32	RW	00000000h
84h	Embedded RAM (RAM1)	32	RW	00000000h
88h	Embedded RAM (RAM2)	32	RW	00000000h
8ch	Embedded RAM (RAM3)	32	RW	00000000h
90h	Embedded RAM (RAM4)	32	RW	00000000h
94h	Embedded RAM (RAM5)	32	RW	00000000h
98h	Embedded RAM (RAM6)	32	RW	00000000h
9ch	Embedded RAM (RAM7)	32	RW	00000000h
a0h	Embedded RAM (RAM8)	32	RW	00000000h
a4h	Embedded RAM (RAM9)	32	RW	00000000h
a8h	Embedded RAM (RAM10)	32	RW	00000000h
ach	Embedded RAM (RAM11)	32	RW	00000000h
b0h	Embedded RAM (RAM12)	32	RW	00000000h
b4h	Embedded RAM (RAM13)	32	RW	00000000h
b8h	Embedded RAM (RAM14)	32	RW	00000000h
bch	Embedded RAM (RAM15)	32	RW	00000000h
c0h	Embedded RAM (RAM16)	32	RW	00000000h
c4h	Embedded RAM (RAM17)	32	RW	00000000h
c8h	Embedded RAM (RAM18)	32	RW	00000000h
cch	Embedded RAM (RAM19)	32	RW	00000000h
d0h	Embedded RAM (RAM20)	32	RW	00000000h
d4h	Embedded RAM (RAM21)	32	RW	00000000h
d8h	Embedded RAM (RAM22)	32	RW	00000000h
dch	Embedded RAM (RAM23)	32	RW	00000000h
e0h	Embedded RAM (RAM24)	32	RW	00000000h
e4h	Embedded RAM (RAM25)	32	RW	00000000h
e8h	Embedded RAM (RAM26)	32	RW	00000000h
ech	Embedded RAM (RAM27)	32	RW	00000000h
f0h	Embedded RAM (RAM28)	32	RW	00000000h
f4h	Embedded RAM (RAM29)	32	RW	00000000h
f8h	Embedded RAM (RAM30)	32	RW	00000000h
fch	Embedded RAM (RAM31)	32	RW	00000000h
100h	Embedded RAM (RAM32)	32	RW	00000000h
104h	Embedded RAM (RAM33)	32	RW	00000000h
108h	Embedded RAM (RAM34)	32	RW	00000000h
10ch	Embedded RAM (RAM35)	32	RW	00000000h
110h	Embedded RAM (RAM36)	32	RW	00000000h
114h	Embedded RAM (RAM37)	32	RW	00000000h
118h	Embedded RAM (RAM38)	32	RW	00000000h
11ch	Embedded RAM (RAM39)	32	RW	00000000h
120h	Embedded RAM (RAM40)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
124h	Embedded RAM (RAM41)	32	RW	00000000h
128h	Embedded RAM (RAM42)	32	RW	00000000h
12ch	Embedded RAM (RAM43)	32	RW	00000000h
130h	Embedded RAM (RAM44)	32	RW	00000000h
134h	Embedded RAM (RAM45)	32	RW	00000000h
138h	Embedded RAM (RAM46)	32	RW	00000000h
13ch	Embedded RAM (RAM47)	32	RW	00000000h
140h	Embedded RAM (RAM48)	32	RW	00000000h
144h	Embedded RAM (RAM49)	32	RW	00000000h
148h	Embedded RAM (RAM50)	32	RW	00000000h
14ch	Embedded RAM (RAM51)	32	RW	00000000h
150h	Embedded RAM (RAM52)	32	RW	00000000h
154h	Embedded RAM (RAM53)	32	RW	00000000h
158h	Embedded RAM (RAM54)	32	RW	00000000h
15ch	Embedded RAM (RAM55)	32	RW	00000000h
160h	Embedded RAM (RAM56)	32	RW	00000000h
164h	Embedded RAM (RAM57)	32	RW	00000000h
168h	Embedded RAM (RAM58)	32	RW	00000000h
16ch	Embedded RAM (RAM59)	32	RW	00000000h
170h	Embedded RAM (RAM60)	32	RW	00000000h
174h	Embedded RAM (RAM61)	32	RW	00000000h
178h	Embedded RAM (RAM62)	32	RW	00000000h
17ch	Embedded RAM (RAM63)	32	RW	00000000h
180h	Embedded RAM (RAM64)	32	RW	00000000h
184h	Embedded RAM (RAM65)	32	RW	00000000h
188h	Embedded RAM (RAM66)	32	RW	00000000h
18ch	Embedded RAM (RAM67)	32	RW	00000000h
190h	Embedded RAM (RAM68)	32	RW	00000000h
194h	Embedded RAM (RAM69)	32	RW	00000000h
198h	Embedded RAM (RAM70)	32	RW	00000000h
19ch	Embedded RAM (RAM71)	32	RW	00000000h
1a0h	Embedded RAM (RAM72)	32	RW	00000000h
1a4h	Embedded RAM (RAM73)	32	RW	00000000h
1a8h	Embedded RAM (RAM74)	32	RW	00000000h
1ach	Embedded RAM (RAM75)	32	RW	00000000h
1b0h	Embedded RAM (RAM76)	32	RW	00000000h
1b4h	Embedded RAM (RAM77)	32	RW	00000000h
1b8h	Embedded RAM (RAM78)	32	RW	00000000h
1bch	Embedded RAM (RAM79)	32	RW	00000000h
1c0h	Embedded RAM (RAM80)	32	RW	00000000h
1c4h	Embedded RAM (RAM81)	32	RW	00000000h
1c8h	Embedded RAM (RAM82)	32	RW	00000000h
1cch	Embedded RAM (RAM83)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
1d0h	Embedded RAM (RAM84)	32	RW	00000000h
1d4h	Embedded RAM (RAM85)	32	RW	00000000h
1d8h	Embedded RAM (RAM86)	32	RW	00000000h
1dch	Embedded RAM (RAM87)	32	RW	00000000h
1e0h	Embedded RAM (RAM88)	32	RW	00000000h
1e4h	Embedded RAM (RAM89)	32	RW	00000000h
1e8h	Embedded RAM (RAM90)	32	RW	00000000h
1ech	Embedded RAM (RAM91)	32	RW	00000000h
1f0h	Embedded RAM (RAM92)	32	RW	00000000h
1f4h	Embedded RAM (RAM93)	32	RW	00000000h
1f8h	Embedded RAM (RAM94)	32	RW	00000000h
1fch	Embedded RAM (RAM95)	32	RW	00000000h
200h	Embedded RAM (RAM96)	32	RW	00000000h
204h	Embedded RAM (RAM97)	32	RW	00000000h
208h	Embedded RAM (RAM98)	32	RW	00000000h
20ch	Embedded RAM (RAM99)	32	RW	00000000h
210h	Embedded RAM (RAM100)	32	RW	00000000h
214h	Embedded RAM (RAM101)	32	RW	00000000h
218h	Embedded RAM (RAM102)	32	RW	00000000h
21ch	Embedded RAM (RAM103)	32	RW	00000000h
220h	Embedded RAM (RAM104)	32	RW	00000000h
224h	Embedded RAM (RAM105)	32	RW	00000000h
228h	Embedded RAM (RAM106)	32	RW	00000000h
22ch	Embedded RAM (RAM107)	32	RW	00000000h
230h	Embedded RAM (RAM108)	32	RW	00000000h
234h	Embedded RAM (RAM109)	32	RW	00000000h
238h	Embedded RAM (RAM110)	32	RW	00000000h
23ch	Embedded RAM (RAM111)	32	RW	00000000h
240h	Embedded RAM (RAM112)	32	RW	00000000h
244h	Embedded RAM (RAM113)	32	RW	00000000h
248h	Embedded RAM (RAM114)	32	RW	00000000h
24ch	Embedded RAM (RAM115)	32	RW	00000000h
250h	Embedded RAM (RAM116)	32	RW	00000000h
254h	Embedded RAM (RAM117)	32	RW	00000000h
258h	Embedded RAM (RAM118)	32	RW	00000000h
25ch	Embedded RAM (RAM119)	32	RW	00000000h
260h	Embedded RAM (RAM120)	32	RW	00000000h
264h	Embedded RAM (RAM121)	32	RW	00000000h
268h	Embedded RAM (RAM122)	32	RW	00000000h
26ch	Embedded RAM (RAM123)	32	RW	00000000h
270h	Embedded RAM (RAM124)	32	RW	00000000h
274h	Embedded RAM (RAM125)	32	RW	00000000h
278h	Embedded RAM (RAM126)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
27ch	Embedded RAM (RAM127)	32	RW	00000000h
280h	Embedded RAM (RAM128)	32	RW	00000000h
284h	Embedded RAM (RAM129)	32	RW	00000000h
288h	Embedded RAM (RAM130)	32	RW	00000000h
28ch	Embedded RAM (RAM131)	32	RW	00000000h
290h	Embedded RAM (RAM132)	32	RW	00000000h
294h	Embedded RAM (RAM133)	32	RW	00000000h
298h	Embedded RAM (RAM134)	32	RW	00000000h
29ch	Embedded RAM (RAM135)	32	RW	00000000h
2a0h	Embedded RAM (RAM136)	32	RW	00000000h
2a4h	Embedded RAM (RAM137)	32	RW	00000000h
2a8h	Embedded RAM (RAM138)	32	RW	00000000h
2ach	Embedded RAM (RAM139)	32	RW	00000000h
2b0h	Embedded RAM (RAM140)	32	RW	00000000h
2b4h	Embedded RAM (RAM141)	32	RW	00000000h
2b8h	Embedded RAM (RAM142)	32	RW	00000000h
2bch	Embedded RAM (RAM143)	32	RW	00000000h
2c0h	Embedded RAM (RAM144)	32	RW	00000000h
2c4h	Embedded RAM (RAM145)	32	RW	00000000h
2c8h	Embedded RAM (RAM146)	32	RW	00000000h
2cch	Embedded RAM (RAM147)	32	RW	00000000h
2d0h	Embedded RAM (RAM148)	32	RW	00000000h
2d4h	Embedded RAM (RAM149)	32	RW	00000000h
2d8h	Embedded RAM (RAM150)	32	RW	00000000h
2dch	Embedded RAM (RAM151)	32	RW	00000000h
2e0h	Embedded RAM (RAM152)	32	RW	00000000h
2e4h	Embedded RAM (RAM153)	32	RW	00000000h
2e8h	Embedded RAM (RAM154)	32	RW	00000000h
2ech	Embedded RAM (RAM155)	32	RW	00000000h
2f0h	Embedded RAM (RAM156)	32	RW	00000000h
2f4h	Embedded RAM (RAM157)	32	RW	00000000h
2f8h	Embedded RAM (RAM158)	32	RW	00000000h
2fch	Embedded RAM (RAM159)	32	RW	00000000h
300h	Embedded RAM (RAM160)	32	RW	00000000h
304h	Embedded RAM (RAM161)	32	RW	00000000h
308h	Embedded RAM (RAM162)	32	RW	00000000h
30ch	Embedded RAM (RAM163)	32	RW	00000000h
310h	Embedded RAM (RAM164)	32	RW	00000000h
314h	Embedded RAM (RAM165)	32	RW	00000000h
318h	Embedded RAM (RAM166)	32	RW	00000000h
31ch	Embedded RAM (RAM167)	32	RW	00000000h
320h	Embedded RAM (RAM168)	32	RW	00000000h
324h	Embedded RAM (RAM169)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
328h	Embedded RAM (RAM170)	32	RW	00000000h
32ch	Embedded RAM (RAM171)	32	RW	00000000h
330h	Embedded RAM (RAM172)	32	RW	00000000h
334h	Embedded RAM (RAM173)	32	RW	00000000h
338h	Embedded RAM (RAM174)	32	RW	00000000h
33ch	Embedded RAM (RAM175)	32	RW	00000000h
340h	Embedded RAM (RAM176)	32	RW	00000000h
344h	Embedded RAM (RAM177)	32	RW	00000000h
348h	Embedded RAM (RAM178)	32	RW	00000000h
34ch	Embedded RAM (RAM179)	32	RW	00000000h
350h	Embedded RAM (RAM180)	32	RW	00000000h
354h	Embedded RAM (RAM181)	32	RW	00000000h
358h	Embedded RAM (RAM182)	32	RW	00000000h
35ch	Embedded RAM (RAM183)	32	RW	00000000h
360h	Embedded RAM (RAM184)	32	RW	00000000h
364h	Embedded RAM (RAM185)	32	RW	00000000h
368h	Embedded RAM (RAM186)	32	RW	00000000h
36ch	Embedded RAM (RAM187)	32	RW	00000000h
370h	Embedded RAM (RAM188)	32	RW	00000000h
374h	Embedded RAM (RAM189)	32	RW	00000000h
378h	Embedded RAM (RAM190)	32	RW	00000000h
37ch	Embedded RAM (RAM191)	32	RW	00000000h
380h	Embedded RAM (RAM192)	32	RW	00000000h
384h	Embedded RAM (RAM193)	32	RW	00000000h
388h	Embedded RAM (RAM194)	32	RW	00000000h
38ch	Embedded RAM (RAM195)	32	RW	00000000h
390h	Embedded RAM (RAM196)	32	RW	00000000h
394h	Embedded RAM (RAM197)	32	RW	00000000h
398h	Embedded RAM (RAM198)	32	RW	00000000h
39ch	Embedded RAM (RAM199)	32	RW	00000000h
3a0h	Embedded RAM (RAM200)	32	RW	00000000h
3a4h	Embedded RAM (RAM201)	32	RW	00000000h
3a8h	Embedded RAM (RAM202)	32	RW	00000000h
3ach	Embedded RAM (RAM203)	32	RW	00000000h
3b0h	Embedded RAM (RAM204)	32	RW	00000000h
3b4h	Embedded RAM (RAM205)	32	RW	00000000h
3b8h	Embedded RAM (RAM206)	32	RW	00000000h
3bch	Embedded RAM (RAM207)	32	RW	00000000h
3c0h	Embedded RAM (RAM208)	32	RW	00000000h
3c4h	Embedded RAM (RAM209)	32	RW	00000000h
3c8h	Embedded RAM (RAM210)	32	RW	00000000h
3cch	Embedded RAM (RAM211)	32	RW	00000000h
3d0h	Embedded RAM (RAM212)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3d4h	Embedded RAM (RAM213)	32	RW	00000000h
3d8h	Embedded RAM (RAM214)	32	RW	00000000h
3dch	Embedded RAM (RAM215)	32	RW	00000000h
3e0h	Embedded RAM (RAM216)	32	RW	00000000h
3e4h	Embedded RAM (RAM217)	32	RW	00000000h
3e8h	Embedded RAM (RAM218)	32	RW	00000000h
3ech	Embedded RAM (RAM219)	32	RW	00000000h
3f0h	Embedded RAM (RAM220)	32	RW	00000000h
3f4h	Embedded RAM (RAM221)	32	RW	00000000h
3f8h	Embedded RAM (RAM222)	32	RW	00000000h
3fch	Embedded RAM (RAM223)	32	RW	00000000h
400h	Embedded RAM (RAM224)	32	RW	00000000h
404h	Embedded RAM (RAM225)	32	RW	00000000h
408h	Embedded RAM (RAM226)	32	RW	00000000h
40ch	Embedded RAM (RAM227)	32	RW	00000000h
410h	Embedded RAM (RAM228)	32	RW	00000000h
414h	Embedded RAM (RAM229)	32	RW	00000000h
418h	Embedded RAM (RAM230)	32	RW	00000000h
41ch	Embedded RAM (RAM231)	32	RW	00000000h
420h	Embedded RAM (RAM232)	32	RW	00000000h
424h	Embedded RAM (RAM233)	32	RW	00000000h
428h	Embedded RAM (RAM234)	32	RW	00000000h
42ch	Embedded RAM (RAM235)	32	RW	00000000h
430h	Embedded RAM (RAM236)	32	RW	00000000h
434h	Embedded RAM (RAM237)	32	RW	00000000h
438h	Embedded RAM (RAM238)	32	RW	00000000h
43ch	Embedded RAM (RAM239)	32	RW	00000000h
440h	Embedded RAM (RAM240)	32	RW	00000000h
444h	Embedded RAM (RAM241)	32	RW	00000000h
448h	Embedded RAM (RAM242)	32	RW	00000000h
44ch	Embedded RAM (RAM243)	32	RW	00000000h
450h	Embedded RAM (RAM244)	32	RW	00000000h
454h	Embedded RAM (RAM245)	32	RW	00000000h
458h	Embedded RAM (RAM246)	32	RW	00000000h
45ch	Embedded RAM (RAM247)	32	RW	00000000h
460h	Embedded RAM (RAM248)	32	RW	00000000h
464h	Embedded RAM (RAM249)	32	RW	00000000h
468h	Embedded RAM (RAM250)	32	RW	00000000h
46ch	Embedded RAM (RAM251)	32	RW	00000000h
470h	Embedded RAM (RAM252)	32	RW	00000000h
474h	Embedded RAM (RAM253)	32	RW	00000000h
478h	Embedded RAM (RAM254)	32	RW	00000000h
47ch	Embedded RAM (RAM255)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
480h	Embedded RAM (RAM256)	32	RW	00000000h
484h	Embedded RAM (RAM257)	32	RW	00000000h
488h	Embedded RAM (RAM258)	32	RW	00000000h
48ch	Embedded RAM (RAM259)	32	RW	00000000h
490h	Embedded RAM (RAM260)	32	RW	00000000h
494h	Embedded RAM (RAM261)	32	RW	00000000h
498h	Embedded RAM (RAM262)	32	RW	00000000h
49ch	Embedded RAM (RAM263)	32	RW	00000000h
4a0h	Embedded RAM (RAM264)	32	RW	00000000h
4a4h	Embedded RAM (RAM265)	32	RW	00000000h
4a8h	Embedded RAM (RAM266)	32	RW	00000000h
4ach	Embedded RAM (RAM267)	32	RW	00000000h
4b0h	Embedded RAM (RAM268)	32	RW	00000000h
4b4h	Embedded RAM (RAM269)	32	RW	00000000h
4b8h	Embedded RAM (RAM270)	32	RW	00000000h
4bch	Embedded RAM (RAM271)	32	RW	00000000h
4c0h	Embedded RAM (RAM272)	32	RW	00000000h
4c4h	Embedded RAM (RAM273)	32	RW	00000000h
4c8h	Embedded RAM (RAM274)	32	RW	00000000h
4cch	Embedded RAM (RAM275)	32	RW	00000000h
4d0h	Embedded RAM (RAM276)	32	RW	00000000h
4d4h	Embedded RAM (RAM277)	32	RW	00000000h
4d8h	Embedded RAM (RAM278)	32	RW	00000000h
4dch	Embedded RAM (RAM279)	32	RW	00000000h
4e0h	Embedded RAM (RAM280)	32	RW	00000000h
4e4h	Embedded RAM (RAM281)	32	RW	00000000h
4e8h	Embedded RAM (RAM282)	32	RW	00000000h
4ech	Embedded RAM (RAM283)	32	RW	00000000h
4f0h	Embedded RAM (RAM284)	32	RW	00000000h
4f4h	Embedded RAM (RAM285)	32	RW	00000000h
4f8h	Embedded RAM (RAM286)	32	RW	00000000h
4fch	Embedded RAM (RAM287)	32	RW	00000000h
500h	Embedded RAM (RAM288)	32	RW	00000000h
504h	Embedded RAM (RAM289)	32	RW	00000000h
508h	Embedded RAM (RAM290)	32	RW	00000000h
50ch	Embedded RAM (RAM291)	32	RW	00000000h
510h	Embedded RAM (RAM292)	32	RW	00000000h
514h	Embedded RAM (RAM293)	32	RW	00000000h
518h	Embedded RAM (RAM294)	32	RW	00000000h
51ch	Embedded RAM (RAM295)	32	RW	00000000h
520h	Embedded RAM (RAM296)	32	RW	00000000h
524h	Embedded RAM (RAM297)	32	RW	00000000h
528h	Embedded RAM (RAM298)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
52ch	Embedded RAM (RAM299)	32	RW	00000000h
530h	Embedded RAM (RAM300)	32	RW	00000000h
534h	Embedded RAM (RAM301)	32	RW	00000000h
538h	Embedded RAM (RAM302)	32	RW	00000000h
53ch	Embedded RAM (RAM303)	32	RW	00000000h
540h	Embedded RAM (RAM304)	32	RW	00000000h
544h	Embedded RAM (RAM305)	32	RW	00000000h
548h	Embedded RAM (RAM306)	32	RW	00000000h
54ch	Embedded RAM (RAM307)	32	RW	00000000h
550h	Embedded RAM (RAM308)	32	RW	00000000h
554h	Embedded RAM (RAM309)	32	RW	00000000h
558h	Embedded RAM (RAM310)	32	RW	00000000h
55ch	Embedded RAM (RAM311)	32	RW	00000000h
560h	Embedded RAM (RAM312)	32	RW	00000000h
564h	Embedded RAM (RAM313)	32	RW	00000000h
568h	Embedded RAM (RAM314)	32	RW	00000000h
56ch	Embedded RAM (RAM315)	32	RW	00000000h
570h	Embedded RAM (RAM316)	32	RW	00000000h
574h	Embedded RAM (RAM317)	32	RW	00000000h
578h	Embedded RAM (RAM318)	32	RW	00000000h
57ch	Embedded RAM (RAM319)	32	RW	00000000h
580h	Embedded RAM (RAM320)	32	RW	00000000h
584h	Embedded RAM (RAM321)	32	RW	00000000h
588h	Embedded RAM (RAM322)	32	RW	00000000h
58ch	Embedded RAM (RAM323)	32	RW	00000000h
590h	Embedded RAM (RAM324)	32	RW	00000000h
594h	Embedded RAM (RAM325)	32	RW	00000000h
598h	Embedded RAM (RAM326)	32	RW	00000000h
59ch	Embedded RAM (RAM327)	32	RW	00000000h
5a0h	Embedded RAM (RAM328)	32	RW	00000000h
5a4h	Embedded RAM (RAM329)	32	RW	00000000h
5a8h	Embedded RAM (RAM330)	32	RW	00000000h
5ach	Embedded RAM (RAM331)	32	RW	00000000h
5b0h	Embedded RAM (RAM332)	32	RW	00000000h
5b4h	Embedded RAM (RAM333)	32	RW	00000000h
5b8h	Embedded RAM (RAM334)	32	RW	00000000h
5bch	Embedded RAM (RAM335)	32	RW	00000000h
5c0h	Embedded RAM (RAM336)	32	RW	00000000h
5c4h	Embedded RAM (RAM337)	32	RW	00000000h
5c8h	Embedded RAM (RAM338)	32	RW	00000000h
5cch	Embedded RAM (RAM339)	32	RW	00000000h
5d0h	Embedded RAM (RAM340)	32	RW	00000000h
5d4h	Embedded RAM (RAM341)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
5d8h	Embedded RAM (RAM342)	32	RW	00000000h
5dch	Embedded RAM (RAM343)	32	RW	00000000h
5e0h	Embedded RAM (RAM344)	32	RW	00000000h
5e4h	Embedded RAM (RAM345)	32	RW	00000000h
5e8h	Embedded RAM (RAM346)	32	RW	00000000h
5ech	Embedded RAM (RAM347)	32	RW	00000000h
5f0h	Embedded RAM (RAM348)	32	RW	00000000h
5f4h	Embedded RAM (RAM349)	32	RW	00000000h
5f8h	Embedded RAM (RAM350)	32	RW	00000000h
5fch	Embedded RAM (RAM351)	32	RW	00000000h
600h	Embedded RAM (RAM352)	32	RW	00000000h
604h	Embedded RAM (RAM353)	32	RW	00000000h
608h	Embedded RAM (RAM354)	32	RW	00000000h
60ch	Embedded RAM (RAM355)	32	RW	00000000h
610h	Embedded RAM (RAM356)	32	RW	00000000h
614h	Embedded RAM (RAM357)	32	RW	00000000h
618h	Embedded RAM (RAM358)	32	RW	00000000h
61ch	Embedded RAM (RAM359)	32	RW	00000000h
620h	Embedded RAM (RAM360)	32	RW	00000000h
624h	Embedded RAM (RAM361)	32	RW	00000000h
628h	Embedded RAM (RAM362)	32	RW	00000000h
62ch	Embedded RAM (RAM363)	32	RW	00000000h
630h	Embedded RAM (RAM364)	32	RW	00000000h
634h	Embedded RAM (RAM365)	32	RW	00000000h
638h	Embedded RAM (RAM366)	32	RW	00000000h
63ch	Embedded RAM (RAM367)	32	RW	00000000h
640h	Embedded RAM (RAM368)	32	RW	00000000h
644h	Embedded RAM (RAM369)	32	RW	00000000h
648h	Embedded RAM (RAM370)	32	RW	00000000h
64ch	Embedded RAM (RAM371)	32	RW	00000000h
650h	Embedded RAM (RAM372)	32	RW	00000000h
654h	Embedded RAM (RAM373)	32	RW	00000000h
658h	Embedded RAM (RAM374)	32	RW	00000000h
65ch	Embedded RAM (RAM375)	32	RW	00000000h
660h	Embedded RAM (RAM376)	32	RW	00000000h
664h	Embedded RAM (RAM377)	32	RW	00000000h
668h	Embedded RAM (RAM378)	32	RW	00000000h
66ch	Embedded RAM (RAM379)	32	RW	00000000h
670h	Embedded RAM (RAM380)	32	RW	00000000h
674h	Embedded RAM (RAM381)	32	RW	00000000h
678h	Embedded RAM (RAM382)	32	RW	00000000h
67ch	Embedded RAM (RAM383)	32	RW	00000000h
680h	Embedded RAM (RAM384)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
684h	Embedded RAM (RAM385)	32	RW	00000000h
688h	Embedded RAM (RAM386)	32	RW	00000000h
68ch	Embedded RAM (RAM387)	32	RW	00000000h
690h	Embedded RAM (RAM388)	32	RW	00000000h
694h	Embedded RAM (RAM389)	32	RW	00000000h
698h	Embedded RAM (RAM390)	32	RW	00000000h
69ch	Embedded RAM (RAM391)	32	RW	00000000h
6a0h	Embedded RAM (RAM392)	32	RW	00000000h
6a4h	Embedded RAM (RAM393)	32	RW	00000000h
6a8h	Embedded RAM (RAM394)	32	RW	00000000h
6ach	Embedded RAM (RAM395)	32	RW	00000000h
6b0h	Embedded RAM (RAM396)	32	RW	00000000h
6b4h	Embedded RAM (RAM397)	32	RW	00000000h
6b8h	Embedded RAM (RAM398)	32	RW	00000000h
6bch	Embedded RAM (RAM399)	32	RW	00000000h
6c0h	Embedded RAM (RAM400)	32	RW	00000000h
6c4h	Embedded RAM (RAM401)	32	RW	00000000h
6c8h	Embedded RAM (RAM402)	32	RW	00000000h
6cch	Embedded RAM (RAM403)	32	RW	00000000h
6d0h	Embedded RAM (RAM404)	32	RW	00000000h
6d4h	Embedded RAM (RAM405)	32	RW	00000000h
6d8h	Embedded RAM (RAM406)	32	RW	00000000h
6dch	Embedded RAM (RAM407)	32	RW	00000000h
6e0h	Embedded RAM (RAM408)	32	RW	00000000h
6e4h	Embedded RAM (RAM409)	32	RW	00000000h
6e8h	Embedded RAM (RAM410)	32	RW	00000000h
6ech	Embedded RAM (RAM411)	32	RW	00000000h
6f0h	Embedded RAM (RAM412)	32	RW	00000000h
6f4h	Embedded RAM (RAM413)	32	RW	00000000h
6f8h	Embedded RAM (RAM414)	32	RW	00000000h
6fch	Embedded RAM (RAM415)	32	RW	00000000h
700h	Embedded RAM (RAM416)	32	RW	00000000h
704h	Embedded RAM (RAM417)	32	RW	00000000h
708h	Embedded RAM (RAM418)	32	RW	00000000h
70ch	Embedded RAM (RAM419)	32	RW	00000000h
710h	Embedded RAM (RAM420)	32	RW	00000000h
714h	Embedded RAM (RAM421)	32	RW	00000000h
718h	Embedded RAM (RAM422)	32	RW	00000000h
71ch	Embedded RAM (RAM423)	32	RW	00000000h
720h	Embedded RAM (RAM424)	32	RW	00000000h
724h	Embedded RAM (RAM425)	32	RW	00000000h
728h	Embedded RAM (RAM426)	32	RW	00000000h
72ch	Embedded RAM (RAM427)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
730h	Embedded RAM (RAM428)	32	RW	00000000h
734h	Embedded RAM (RAM429)	32	RW	00000000h
738h	Embedded RAM (RAM430)	32	RW	00000000h
73ch	Embedded RAM (RAM431)	32	RW	00000000h
740h	Embedded RAM (RAM432)	32	RW	00000000h
744h	Embedded RAM (RAM433)	32	RW	00000000h
748h	Embedded RAM (RAM434)	32	RW	00000000h
74ch	Embedded RAM (RAM435)	32	RW	00000000h
750h	Embedded RAM (RAM436)	32	RW	00000000h
754h	Embedded RAM (RAM437)	32	RW	00000000h
758h	Embedded RAM (RAM438)	32	RW	00000000h
75ch	Embedded RAM (RAM439)	32	RW	00000000h
760h	Embedded RAM (RAM440)	32	RW	00000000h
764h	Embedded RAM (RAM441)	32	RW	00000000h
768h	Embedded RAM (RAM442)	32	RW	00000000h
76ch	Embedded RAM (RAM443)	32	RW	00000000h
770h	Embedded RAM (RAM444)	32	RW	00000000h
774h	Embedded RAM (RAM445)	32	RW	00000000h
778h	Embedded RAM (RAM446)	32	RW	00000000h
77ch	Embedded RAM (RAM447)	32	RW	00000000h
780h	Embedded RAM (RAM448)	32	RW	00000000h
784h	Embedded RAM (RAM449)	32	RW	00000000h
788h	Embedded RAM (RAM450)	32	RW	00000000h
78ch	Embedded RAM (RAM451)	32	RW	00000000h
790h	Embedded RAM (RAM452)	32	RW	00000000h
794h	Embedded RAM (RAM453)	32	RW	00000000h
798h	Embedded RAM (RAM454)	32	RW	00000000h
79ch	Embedded RAM (RAM455)	32	RW	00000000h
7a0h	Embedded RAM (RAM456)	32	RW	00000000h
7a4h	Embedded RAM (RAM457)	32	RW	00000000h
7a8h	Embedded RAM (RAM458)	32	RW	00000000h
7ach	Embedded RAM (RAM459)	32	RW	00000000h
7b0h	Embedded RAM (RAM460)	32	RW	00000000h
7b4h	Embedded RAM (RAM461)	32	RW	00000000h
7b8h	Embedded RAM (RAM462)	32	RW	00000000h
7bch	Embedded RAM (RAM463)	32	RW	00000000h
7c0h	Embedded RAM (RAM464)	32	RW	00000000h
7c4h	Embedded RAM (RAM465)	32	RW	00000000h
7c8h	Embedded RAM (RAM466)	32	RW	00000000h
7cch	Embedded RAM (RAM467)	32	RW	00000000h
7d0h	Embedded RAM (RAM468)	32	RW	00000000h
7d4h	Embedded RAM (RAM469)	32	RW	00000000h
7d8h	Embedded RAM (RAM470)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
7dch	Embedded RAM (RAM471)	32	RW	00000000h
7e0h	Embedded RAM (RAM472)	32	RW	00000000h
7e4h	Embedded RAM (RAM473)	32	RW	00000000h
7e8h	Embedded RAM (RAM474)	32	RW	00000000h
7ech	Embedded RAM (RAM475)	32	RW	00000000h
7f0h	Embedded RAM (RAM476)	32	RW	00000000h
7f4h	Embedded RAM (RAM477)	32	RW	00000000h
7f8h	Embedded RAM (RAM478)	32	RW	00000000h
7fch	Embedded RAM (RAM479)	32	RW	00000000h
800h	Embedded RAM (RAM480)	32	RW	00000000h
804h	Embedded RAM (RAM481)	32	RW	00000000h
808h	Embedded RAM (RAM482)	32	RW	00000000h
80ch	Embedded RAM (RAM483)	32	RW	00000000h
810h	Embedded RAM (RAM484)	32	RW	00000000h
814h	Embedded RAM (RAM485)	32	RW	00000000h
818h	Embedded RAM (RAM486)	32	RW	00000000h
81ch	Embedded RAM (RAM487)	32	RW	00000000h
820h	Embedded RAM (RAM488)	32	RW	00000000h
824h	Embedded RAM (RAM489)	32	RW	00000000h
828h	Embedded RAM (RAM490)	32	RW	00000000h
82ch	Embedded RAM (RAM491)	32	RW	00000000h
830h	Embedded RAM (RAM492)	32	RW	00000000h
834h	Embedded RAM (RAM493)	32	RW	00000000h
838h	Embedded RAM (RAM494)	32	RW	00000000h
83ch	Embedded RAM (RAM495)	32	RW	00000000h
840h	Embedded RAM (RAM496)	32	RW	00000000h
844h	Embedded RAM (RAM497)	32	RW	00000000h
848h	Embedded RAM (RAM498)	32	RW	00000000h
84ch	Embedded RAM (RAM499)	32	RW	00000000h
850h	Embedded RAM (RAM500)	32	RW	00000000h
854h	Embedded RAM (RAM501)	32	RW	00000000h
858h	Embedded RAM (RAM502)	32	RW	00000000h
85ch	Embedded RAM (RAM503)	32	RW	00000000h
860h	Embedded RAM (RAM504)	32	RW	00000000h
864h	Embedded RAM (RAM505)	32	RW	00000000h
868h	Embedded RAM (RAM506)	32	RW	00000000h
86ch	Embedded RAM (RAM507)	32	RW	00000000h
870h	Embedded RAM (RAM508)	32	RW	00000000h
874h	Embedded RAM (RAM509)	32	RW	00000000h
878h	Embedded RAM (RAM510)	32	RW	00000000h
87ch	Embedded RAM (RAM511)	32	RW	00000000h
880h	RX Individual Mask Register (RXIMR0)	32	RW	00000000h
884h	RX Individual Mask Register (RXIMR1)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
888h	RX Individual Mask Register (RXIMR2)	32	RW	00000000h
88ch	RX Individual Mask Register (RXIMR3)	32	RW	00000000h
890h	RX Individual Mask Register (RXIMR4)	32	RW	00000000h
894h	RX Individual Mask Register (RXIMR5)	32	RW	00000000h
898h	RX Individual Mask Register (RXIMR6)	32	RW	00000000h
89ch	RX Individual Mask Register (RXIMR7)	32	RW	00000000h
8a0h	RX Individual Mask Register (RXIMR8)	32	RW	00000000h
8a4h	RX Individual Mask Register (RXIMR9)	32	RW	00000000h
8a8h	RX Individual Mask Register (RXIMR10)	32	RW	00000000h
8ach	RX Individual Mask Register (RXIMR11)	32	RW	00000000h
8b0h	RX Individual Mask Register (RXIMR12)	32	RW	00000000h
8b4h	RX Individual Mask Register (RXIMR13)	32	RW	00000000h
8b8h	RX Individual Mask Register (RXIMR14)	32	RW	00000000h
8bch	RX Individual Mask Register (RXIMR15)	32	RW	00000000h
8c0h	RX Individual Mask Register (RXIMR16)	32	RW	00000000h
8c4h	RX Individual Mask Register (RXIMR17)	32	RW	00000000h
8c8h	RX Individual Mask Register (RXIMR18)	32	RW	00000000h
8cch	RX Individual Mask Register (RXIMR19)	32	RW	00000000h
8d0h	RX Individual Mask Register (RXIMR20)	32	RW	00000000h
8d4h	RX Individual Mask Register (RXIMR21)	32	RW	00000000h
8d8h	RX Individual Mask Register (RXIMR22)	32	RW	00000000h
8dch	RX Individual Mask Register (RXIMR23)	32	RW	00000000h
8e0h	RX Individual Mask Register (RXIMR24)	32	RW	00000000h
8e4h	RX Individual Mask Register (RXIMR25)	32	RW	00000000h
8e8h	RX Individual Mask Register (RXIMR26)	32	RW	00000000h
8ech	RX Individual Mask Register (RXIMR27)	32	RW	00000000h
8f0h	RX Individual Mask Register (RXIMR28)	32	RW	00000000h
8f4h	RX Individual Mask Register (RXIMR29)	32	RW	00000000h
8f8h	RX Individual Mask Register (RXIMR30)	32	RW	00000000h
8fch	RX Individual Mask Register (RXIMR31)	32	RW	00000000h
900h	RX Individual Mask Register (RXIMR32)	32	RW	00000000h
904h	RX Individual Mask Register (RXIMR33)	32	RW	00000000h
908h	RX Individual Mask Register (RXIMR34)	32	RW	00000000h
90ch	RX Individual Mask Register (RXIMR35)	32	RW	00000000h
910h	RX Individual Mask Register (RXIMR36)	32	RW	00000000h
914h	RX Individual Mask Register (RXIMR37)	32	RW	00000000h
918h	RX Individual Mask Register (RXIMR38)	32	RW	00000000h
91ch	RX Individual Mask Register (RXIMR39)	32	RW	00000000h
920h	RX Individual Mask Register (RXIMR40)	32	RW	00000000h
924h	RX Individual Mask Register (RXIMR41)	32	RW	00000000h
928h	RX Individual Mask Register (RXIMR42)	32	RW	00000000h
92ch	RX Individual Mask Register (RXIMR43)	32	RW	00000000h
930h	RX Individual Mask Register (RXIMR44)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
934h	RX Individual Mask Register (RXIMR45)	32	RW	00000000h
938h	RX Individual Mask Register (RXIMR46)	32	RW	00000000h
93ch	RX Individual Mask Register (RXIMR47)	32	RW	00000000h
940h	RX Individual Mask Register (RXIMR48)	32	RW	00000000h
944h	RX Individual Mask Register (RXIMR49)	32	RW	00000000h
948h	RX Individual Mask Register (RXIMR50)	32	RW	00000000h
94ch	RX Individual Mask Register (RXIMR51)	32	RW	00000000h
950h	RX Individual Mask Register (RXIMR52)	32	RW	00000000h
954h	RX Individual Mask Register (RXIMR53)	32	RW	00000000h
958h	RX Individual Mask Register (RXIMR54)	32	RW	00000000h
95ch	RX Individual Mask Register (RXIMR55)	32	RW	00000000h
960h	RX Individual Mask Register (RXIMR56)	32	RW	00000000h
964h	RX Individual Mask Register (RXIMR57)	32	RW	00000000h
968h	RX Individual Mask Register (RXIMR58)	32	RW	00000000h
96ch	RX Individual Mask Register (RXIMR59)	32	RW	00000000h
970h	RX Individual Mask Register (RXIMR60)	32	RW	00000000h
974h	RX Individual Mask Register (RXIMR61)	32	RW	00000000h
978h	RX Individual Mask Register (RXIMR62)	32	RW	00000000h
97ch	RX Individual Mask Register (RXIMR63)	32	RW	00000000h
980h	RX Individual Mask Register (RXIMR64)	32	RW	00000000h
984h	RX Individual Mask Register (RXIMR65)	32	RW	00000000h
988h	RX Individual Mask Register (RXIMR66)	32	RW	00000000h
98ch	RX Individual Mask Register (RXIMR67)	32	RW	00000000h
990h	RX Individual Mask Register (RXIMR68)	32	RW	00000000h
994h	RX Individual Mask Register (RXIMR69)	32	RW	00000000h
998h	RX Individual Mask Register (RXIMR70)	32	RW	00000000h
99ch	RX Individual Mask Register (RXIMR71)	32	RW	00000000h
9a0h	RX Individual Mask Register (RXIMR72)	32	RW	00000000h
9a4h	RX Individual Mask Register (RXIMR73)	32	RW	00000000h
9a8h	RX Individual Mask Register (RXIMR74)	32	RW	00000000h
9ach	RX Individual Mask Register (RXIMR75)	32	RW	00000000h
9b0h	RX Individual Mask Register (RXIMR76)	32	RW	00000000h
9b4h	RX Individual Mask Register (RXIMR77)	32	RW	00000000h
9b8h	RX Individual Mask Register (RXIMR78)	32	RW	00000000h
9bch	RX Individual Mask Register (RXIMR79)	32	RW	00000000h
9c0h	RX Individual Mask Register (RXIMR80)	32	RW	00000000h
9c4h	RX Individual Mask Register (RXIMR81)	32	RW	00000000h
9c8h	RX Individual Mask Register (RXIMR82)	32	RW	00000000h
9cch	RX Individual Mask Register (RXIMR83)	32	RW	00000000h
9d0h	RX Individual Mask Register (RXIMR84)	32	RW	00000000h
9d4h	RX Individual Mask Register (RXIMR85)	32	RW	00000000h
9d8h	RX Individual Mask Register (RXIMR86)	32	RW	00000000h
9dch	RX Individual Mask Register (RXIMR87)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
9e0h	RX Individual Mask Register (RXIMR88)	32	RW	00000000h
9e4h	RX Individual Mask Register (RXIMR89)	32	RW	00000000h
9e8h	RX Individual Mask Register (RXIMR90)	32	RW	00000000h
9ech	RX Individual Mask Register (RXIMR91)	32	RW	00000000h
9f0h	RX Individual Mask Register (RXIMR92)	32	RW	00000000h
9f4h	RX Individual Mask Register (RXIMR93)	32	RW	00000000h
9f8h	RX Individual Mask Register (RXIMR94)	32	RW	00000000h
9fch	RX Individual Mask Register (RXIMR95)	32	RW	00000000h
a00h	RX Individual Mask Register (RXIMR96)	32	RW	00000000h
a04h	RX Individual Mask Register (RXIMR97)	32	RW	00000000h
a08h	RX Individual Mask Register (RXIMR98)	32	RW	00000000h
a0ch	RX Individual Mask Register (RXIMR99)	32	RW	00000000h
a10h	RX Individual Mask Register (RXIMR100)	32	RW	00000000h
a14h	RX Individual Mask Register (RXIMR101)	32	RW	00000000h
a18h	RX Individual Mask Register (RXIMR102)	32	RW	00000000h
a1ch	RX Individual Mask Register (RXIMR103)	32	RW	00000000h
a20h	RX Individual Mask Register (RXIMR104)	32	RW	00000000h
a24h	RX Individual Mask Register (RXIMR105)	32	RW	00000000h
a28h	RX Individual Mask Register (RXIMR106)	32	RW	00000000h
a2ch	RX Individual Mask Register (RXIMR107)	32	RW	00000000h
a30h	RX Individual Mask Register (RXIMR108)	32	RW	00000000h
a34h	RX Individual Mask Register (RXIMR109)	32	RW	00000000h
a38h	RX Individual Mask Register (RXIMR110)	32	RW	00000000h
a3ch	RX Individual Mask Register (RXIMR111)	32	RW	00000000h
a40h	RX Individual Mask Register (RXIMR112)	32	RW	00000000h
a44h	RX Individual Mask Register (RXIMR113)	32	RW	00000000h
a48h	RX Individual Mask Register (RXIMR114)	32	RW	00000000h
a4ch	RX Individual Mask Register (RXIMR115)	32	RW	00000000h
a50h	RX Individual Mask Register (RXIMR116)	32	RW	00000000h
a54h	RX Individual Mask Register (RXIMR117)	32	RW	00000000h
a58h	RX Individual Mask Register (RXIMR118)	32	RW	00000000h
a5ch	RX Individual Mask Register (RXIMR119)	32	RW	00000000h
a60h	RX Individual Mask Register (RXIMR120)	32	RW	00000000h
a64h	RX Individual Mask Register (RXIMR121)	32	RW	00000000h
a68h	RX Individual Mask Register (RXIMR122)	32	RW	00000000h
a6ch	RX Individual Mask Register (RXIMR123)	32	RW	00000000h
a70h	RX Individual Mask Register (RXIMR124)	32	RW	00000000h
a74h	RX Individual Mask Register (RXIMR125)	32	RW	00000000h
a78h	RX Individual Mask Register (RXIMR126)	32	RW	00000000h
a7ch	RX Individual Mask Register (RXIMR127)	32	RW	00000000h
ae0h	Memory Error Control Register (MECR)	32	RW	00000000h
ae4h	Error Injection Address Register (ERRIAR)	32	RW	00000000h
ae8h	Error Injection Data Pattern Register (ERRIDPR)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
aech	Error Injection Parity Pattern Register (ERRIPPR)	32	RW	00000000h
af0h	Error Report Address Register (RERRAR)	32	RW	00000000h
af4h	Error Report Data Register (RERRDR)	32	RW	00000000h
af8h	Error Report Syndrome Register (RERRSYNR)	32	RW	00000000h
afch	Error Status Register (ERRSR)	32	RW	00000000h
b00h	Pretended Networking Control 1 Register (CTRL1_PN)	32	RW	00000000h
b04h	Pretended Networking Control 2 Register (CTRL2_PN)	32	RW	00000000h
b08h	Pretended Networking Wake Up Match Register (WU_MTC)	32	RW	00000000h
b0ch	Pretended Networking ID Filter 1 Register (FLT_ID1)	32	RW	00000000h
b10h	Pretended Networking DLC Filter Register (FLT_DLC)	32	RW	00000000h
b14h	Pretended Networking Payload Low Filter 1 Register (PL1_LO)	32	RW	00000000h
b18h	Pretended Networking Payload High Filter 1 Register (PL1_HI)	32	RW	00000000h
b1ch	Pretended Networking ID Filter 2 Register (FLT_ID2_IDMASK)	32	RW	00000000h
b20h	Pretended Networking Payload Low Filter 2 Register (PL2_PLMASK_LO)	32	RW	00000000h
b24h	Pretended Networking Payload High Filter 2 Register (PL2_PLMASK_HI)	32	RW	00000000h
b40h	Wakeup Message Buffer Control Status Register (WMB0_WMBn_CS)	32	RW	00000000h
b44h	Wakeup Message Buffer ID (WMB0_WMBn_ID)	32	RW	00000000h
b48h	Wakeup Message Buffer Data Byte 0~3 (WMB0_WMBn_D03)	32	RW	00000000h
b4ch	Wakeup Message Buffer Data Byte 4~7 (WMB0_WMBn_D47)	32	RW	00000000h
b50h	Wakeup Message Buffer Control Status Register (WMB1_WMBn_CS)	32	RW	00000000h
b54h	Wakeup Message Buffer ID (WMB1_WMBn_ID)	32	RW	00000000h
b58h	Wakeup Message Buffer Data Byte 0~3 (WMB1_WMBn_D03)	32	RW	00000000h
b5ch	Wakeup Message Buffer Data Byte 4~7 (WMB1_WMBn_D47)	32	RW	00000000h
b60h	Wakeup Message Buffer Control Status Register (WMB2_WMBn_CS)	32	RW	00000000h
b64h	Wakeup Message Buffer ID (WMB2_WMBn_ID)	32	RW	00000000h
b68h	Wakeup Message Buffer Data Byte 0~3 (WMB2_WMBn_D03)	32	RW	00000000h
b6ch	Wakeup Message Buffer Data Byte 4~7 (WMB2_WMBn_D47)	32	RW	00000000h
b70h	Wakeup Message Buffer Control Status Register (WMB3_WMBn_CS)	32	RW	00000000h
b74h	Wakeup Message Buffer ID (WMB3_WMBn_ID)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
b78h	Wakeup Message Buffer Data Byte 0~3 (WMB3_WMBn_D03)	32	RW	00000000h
b7ch	Wakeup Message Buffer Data Byte 4~7 (WMB3_WMBn_D47)	32	RW	00000000h
bf0h	Enhanced CAN Bit Timing Prescalers Register (EPRS)	32	RW	00000000h
bf4h	Enhanced Nominal CAN Bit Timing Register (ENCBT)	32	RW	00000000h
bf8h	Enhanced Data Phase CAN Bit Timing Register (EDCBT)	32	RW	00000000h
bfch	Enhanced Transceiver Delay Compensation Register (ETDC)	32	RW	00000000h
c00h	FD Control Register (FDCTRL)	32	RW	00000000h
c04h	FD Bit Timing Register (FDCBT)	32	RW	00000000h
c08h	FD CRC Register (FDCRC)	32	RW	00000000h
c0ch	Enhanced Rx FIFO Control Register (ERFCR)	32	RW	00000000h
c10h	Enhanced Rx FIFO Interrupt Enable Register (ERFIER)	32	RW	00000000h
c14h	Enhanced Rx FIFO Status Register (ERFSR)	32	RW	00000000h
c30h	High Resolution Time Stamp Registers (HR_TIME_STAMPn0)	32	RW	00000000h
c34h	High Resolution Time Stamp Registers (HR_TIME_STAMPn1)	32	RW	00000000h
c38h	High Resolution Time Stamp Registers (HR_TIME_STAMPn2)	32	RW	00000000h
c3ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn3)	32	RW	00000000h
c40h	High Resolution Time Stamp Registers (HR_TIME_STAMPn4)	32	RW	00000000h
c44h	High Resolution Time Stamp Registers (HR_TIME_STAMPn5)	32	RW	00000000h
c48h	High Resolution Time Stamp Registers (HR_TIME_STAMPn6)	32	RW	00000000h
c4ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn7)	32	RW	00000000h
c50h	High Resolution Time Stamp Registers (HR_TIME_STAMPn8)	32	RW	00000000h
c54h	High Resolution Time Stamp Registers (HR_TIME_STAMPn9)	32	RW	00000000h
c58h	High Resolution Time Stamp Registers (HR_TIME_STAMPn10)	32	RW	00000000h
c5ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn11)	32	RW	00000000h
c60h	High Resolution Time Stamp Registers (HR_TIME_STAMPn12)	32	RW	00000000h
c64h	High Resolution Time Stamp Registers (HR_TIME_STAMPn13)	32	RW	00000000h
c68h	High Resolution Time Stamp Registers (HR_TIME_STAMPn14)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
c6ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn15)	32	RW	00000000h
c70h	High Resolution Time Stamp Registers (HR_TIME_STAMPn16)	32	RW	00000000h
c74h	High Resolution Time Stamp Registers (HR_TIME_STAMPn17)	32	RW	00000000h
c78h	High Resolution Time Stamp Registers (HR_TIME_STAMPn18)	32	RW	00000000h
c7ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn19)	32	RW	00000000h
c80h	High Resolution Time Stamp Registers (HR_TIME_STAMPn20)	32	RW	00000000h
c84h	High Resolution Time Stamp Registers (HR_TIME_STAMPn21)	32	RW	00000000h
c88h	High Resolution Time Stamp Registers (HR_TIME_STAMPn22)	32	RW	00000000h
c8ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn23)	32	RW	00000000h
c90h	High Resolution Time Stamp Registers (HR_TIME_STAMPn24)	32	RW	00000000h
c94h	High Resolution Time Stamp Registers (HR_TIME_STAMPn25)	32	RW	00000000h
c98h	High Resolution Time Stamp Registers (HR_TIME_STAMPn26)	32	RW	00000000h
c9ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn27)	32	RW	00000000h
ca0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn28)	32	RW	00000000h
ca4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn29)	32	RW	00000000h
ca8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn30)	32	RW	00000000h
cach	High Resolution Time Stamp Registers (HR_TIME_STAMPn31)	32	RW	00000000h
cb0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn32)	32	RW	00000000h
cb4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn33)	32	RW	00000000h
cb8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn34)	32	RW	00000000h
cbch	High Resolution Time Stamp Registers (HR_TIME_STAMPn35)	32	RW	00000000h
cc0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn36)	32	RW	00000000h
cc4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn37)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
cc8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn38)	32	RW	00000000h
ccch	High Resolution Time Stamp Registers (HR_TIME_STAMPn39)	32	RW	00000000h
cd0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn40)	32	RW	00000000h
cd4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn41)	32	RW	00000000h
cd8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn42)	32	RW	00000000h
cdch	High Resolution Time Stamp Registers (HR_TIME_STAMPn43)	32	RW	00000000h
ce0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn44)	32	RW	00000000h
ce4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn45)	32	RW	00000000h
ce8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn46)	32	RW	00000000h
cech	High Resolution Time Stamp Registers (HR_TIME_STAMPn47)	32	RW	00000000h
cf0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn48)	32	RW	00000000h
cf4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn49)	32	RW	00000000h
cf8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn50)	32	RW	00000000h
cfch	High Resolution Time Stamp Registers (HR_TIME_STAMPn51)	32	RW	00000000h
d00h	High Resolution Time Stamp Registers (HR_TIME_STAMPn52)	32	RW	00000000h
d04h	High Resolution Time Stamp Registers (HR_TIME_STAMPn53)	32	RW	00000000h
d08h	High Resolution Time Stamp Registers (HR_TIME_STAMPn54)	32	RW	00000000h
d0ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn55)	32	RW	00000000h
d10h	High Resolution Time Stamp Registers (HR_TIME_STAMPn56)	32	RW	00000000h
d14h	High Resolution Time Stamp Registers (HR_TIME_STAMPn57)	32	RW	00000000h
d18h	High Resolution Time Stamp Registers (HR_TIME_STAMPn58)	32	RW	00000000h
d1ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn59)	32	RW	00000000h
d20h	High Resolution Time Stamp Registers (HR_TIME_STAMPn60)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
d24h	High Resolution Time Stamp Registers (HR_TIME_STAMPn61)	32	RW	00000000h
d28h	High Resolution Time Stamp Registers (HR_TIME_STAMPn62)	32	RW	00000000h
d2ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn63)	32	RW	00000000h
d30h	High Resolution Time Stamp Registers (HR_TIME_STAMPn64)	32	RW	00000000h
d34h	High Resolution Time Stamp Registers (HR_TIME_STAMPn65)	32	RW	00000000h
d38h	High Resolution Time Stamp Registers (HR_TIME_STAMPn66)	32	RW	00000000h
d3ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn67)	32	RW	00000000h
d40h	High Resolution Time Stamp Registers (HR_TIME_STAMPn68)	32	RW	00000000h
d44h	High Resolution Time Stamp Registers (HR_TIME_STAMPn69)	32	RW	00000000h
d48h	High Resolution Time Stamp Registers (HR_TIME_STAMPn70)	32	RW	00000000h
d4ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn71)	32	RW	00000000h
d50h	High Resolution Time Stamp Registers (HR_TIME_STAMPn72)	32	RW	00000000h
d54h	High Resolution Time Stamp Registers (HR_TIME_STAMPn73)	32	RW	00000000h
d58h	High Resolution Time Stamp Registers (HR_TIME_STAMPn74)	32	RW	00000000h
d5ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn75)	32	RW	00000000h
d60h	High Resolution Time Stamp Registers (HR_TIME_STAMPn76)	32	RW	00000000h
d64h	High Resolution Time Stamp Registers (HR_TIME_STAMPn77)	32	RW	00000000h
d68h	High Resolution Time Stamp Registers (HR_TIME_STAMPn78)	32	RW	00000000h
d6ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn79)	32	RW	00000000h
d70h	High Resolution Time Stamp Registers (HR_TIME_STAMPn80)	32	RW	00000000h
d74h	High Resolution Time Stamp Registers (HR_TIME_STAMPn81)	32	RW	00000000h
d78h	High Resolution Time Stamp Registers (HR_TIME_STAMPn82)	32	RW	00000000h
d7ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn83)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
d80h	High Resolution Time Stamp Registers (HR_TIME_STAMPn84)	32	RW	00000000h
d84h	High Resolution Time Stamp Registers (HR_TIME_STAMPn85)	32	RW	00000000h
d88h	High Resolution Time Stamp Registers (HR_TIME_STAMPn86)	32	RW	00000000h
d8ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn87)	32	RW	00000000h
d90h	High Resolution Time Stamp Registers (HR_TIME_STAMPn88)	32	RW	00000000h
d94h	High Resolution Time Stamp Registers (HR_TIME_STAMPn89)	32	RW	00000000h
d98h	High Resolution Time Stamp Registers (HR_TIME_STAMPn90)	32	RW	00000000h
d9ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn91)	32	RW	00000000h
da0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn92)	32	RW	00000000h
da4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn93)	32	RW	00000000h
da8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn94)	32	RW	00000000h
dach	High Resolution Time Stamp Registers (HR_TIME_STAMPn95)	32	RW	00000000h
db0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn96)	32	RW	00000000h
db4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn97)	32	RW	00000000h
db8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn98)	32	RW	00000000h
dbch	High Resolution Time Stamp Registers (HR_TIME_STAMPn99)	32	RW	00000000h
dc0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn100)	32	RW	00000000h
dc4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn101)	32	RW	00000000h
dc8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn102)	32	RW	00000000h
dcch	High Resolution Time Stamp Registers (HR_TIME_STAMPn103)	32	RW	00000000h
dd0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn104)	32	RW	00000000h
dd4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn105)	32	RW	00000000h
dd8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn106)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
ddch	High Resolution Time Stamp Registers (HR_TIME_STAMPn107)	32	RW	00000000h
de0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn108)	32	RW	00000000h
de4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn109)	32	RW	00000000h
de8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn110)	32	RW	00000000h
dech	High Resolution Time Stamp Registers (HR_TIME_STAMPn111)	32	RW	00000000h
df0h	High Resolution Time Stamp Registers (HR_TIME_STAMPn112)	32	RW	00000000h
df4h	High Resolution Time Stamp Registers (HR_TIME_STAMPn113)	32	RW	00000000h
df8h	High Resolution Time Stamp Registers (HR_TIME_STAMPn114)	32	RW	00000000h
dfch	High Resolution Time Stamp Registers (HR_TIME_STAMPn115)	32	RW	00000000h
e00h	High Resolution Time Stamp Registers (HR_TIME_STAMPn116)	32	RW	00000000h
e04h	High Resolution Time Stamp Registers (HR_TIME_STAMPn117)	32	RW	00000000h
e08h	High Resolution Time Stamp Registers (HR_TIME_STAMPn118)	32	RW	00000000h
e0ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn119)	32	RW	00000000h
e10h	High Resolution Time Stamp Registers (HR_TIME_STAMPn120)	32	RW	00000000h
e14h	High Resolution Time Stamp Registers (HR_TIME_STAMPn121)	32	RW	00000000h
e18h	High Resolution Time Stamp Registers (HR_TIME_STAMPn122)	32	RW	00000000h
e1ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn123)	32	RW	00000000h
e20h	High Resolution Time Stamp Registers (HR_TIME_STAMPn124)	32	RW	00000000h
e24h	High Resolution Time Stamp Registers (HR_TIME_STAMPn125)	32	RW	00000000h
e28h	High Resolution Time Stamp Registers (HR_TIME_STAMPn126)	32	RW	00000000h
e2ch	High Resolution Time Stamp Registers (HR_TIME_STAMPn127)	32	RW	00000000h
3000h	Enhanced Rx FIFO Filter Element Registers (ERFFELn0)	32	RW	00000000h
3004h	Enhanced Rx FIFO Filter Element Registers (ERFFELn1)	32	RW	00000000h
3008h	Enhanced Rx FIFO Filter Element Registers (ERFFELn2)	32	RW	00000000h
300ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn3)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3010h	Enhanced Rx FIFO Filter Element Registers (ERFFELn4)	32	RW	00000000h
3014h	Enhanced Rx FIFO Filter Element Registers (ERFFELn5)	32	RW	00000000h
3018h	Enhanced Rx FIFO Filter Element Registers (ERFFELn6)	32	RW	00000000h
301ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn7)	32	RW	00000000h
3020h	Enhanced Rx FIFO Filter Element Registers (ERFFELn8)	32	RW	00000000h
3024h	Enhanced Rx FIFO Filter Element Registers (ERFFELn9)	32	RW	00000000h
3028h	Enhanced Rx FIFO Filter Element Registers (ERFFELn10)	32	RW	00000000h
302ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn11)	32	RW	00000000h
3030h	Enhanced Rx FIFO Filter Element Registers (ERFFELn12)	32	RW	00000000h
3034h	Enhanced Rx FIFO Filter Element Registers (ERFFELn13)	32	RW	00000000h
3038h	Enhanced Rx FIFO Filter Element Registers (ERFFELn14)	32	RW	00000000h
303ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn15)	32	RW	00000000h
3040h	Enhanced Rx FIFO Filter Element Registers (ERFFELn16)	32	RW	00000000h
3044h	Enhanced Rx FIFO Filter Element Registers (ERFFELn17)	32	RW	00000000h
3048h	Enhanced Rx FIFO Filter Element Registers (ERFFELn18)	32	RW	00000000h
304ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn19)	32	RW	00000000h
3050h	Enhanced Rx FIFO Filter Element Registers (ERFFELn20)	32	RW	00000000h
3054h	Enhanced Rx FIFO Filter Element Registers (ERFFELn21)	32	RW	00000000h
3058h	Enhanced Rx FIFO Filter Element Registers (ERFFELn22)	32	RW	00000000h
305ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn23)	32	RW	00000000h
3060h	Enhanced Rx FIFO Filter Element Registers (ERFFELn24)	32	RW	00000000h
3064h	Enhanced Rx FIFO Filter Element Registers (ERFFELn25)	32	RW	00000000h
3068h	Enhanced Rx FIFO Filter Element Registers (ERFFELn26)	32	RW	00000000h
306ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn27)	32	RW	00000000h
3070h	Enhanced Rx FIFO Filter Element Registers (ERFFELn28)	32	RW	00000000h
3074h	Enhanced Rx FIFO Filter Element Registers (ERFFELn29)	32	RW	00000000h
3078h	Enhanced Rx FIFO Filter Element Registers (ERFFELn30)	32	RW	00000000h
307ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn31)	32	RW	00000000h
3080h	Enhanced Rx FIFO Filter Element Registers (ERFFELn32)	32	RW	00000000h
3084h	Enhanced Rx FIFO Filter Element Registers (ERFFELn33)	32	RW	00000000h
3088h	Enhanced Rx FIFO Filter Element Registers (ERFFELn34)	32	RW	00000000h
308ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn35)	32	RW	00000000h
3090h	Enhanced Rx FIFO Filter Element Registers (ERFFELn36)	32	RW	00000000h
3094h	Enhanced Rx FIFO Filter Element Registers (ERFFELn37)	32	RW	00000000h
3098h	Enhanced Rx FIFO Filter Element Registers (ERFFELn38)	32	RW	00000000h
309ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn39)	32	RW	00000000h
30a0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn40)	32	RW	00000000h
30a4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn41)	32	RW	00000000h
30a8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn42)	32	RW	00000000h
30ach	Enhanced Rx FIFO Filter Element Registers (ERFFELn43)	32	RW	00000000h
30b0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn44)	32	RW	00000000h
30b4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn45)	32	RW	00000000h
30b8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn46)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
30bch	Enhanced Rx FIFO Filter Element Registers (ERFFELn47)	32	RW	00000000h
30c0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn48)	32	RW	00000000h
30c4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn49)	32	RW	00000000h
30c8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn50)	32	RW	00000000h
30cch	Enhanced Rx FIFO Filter Element Registers (ERFFELn51)	32	RW	00000000h
30d0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn52)	32	RW	00000000h
30d4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn53)	32	RW	00000000h
30d8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn54)	32	RW	00000000h
30dch	Enhanced Rx FIFO Filter Element Registers (ERFFELn55)	32	RW	00000000h
30e0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn56)	32	RW	00000000h
30e4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn57)	32	RW	00000000h
30e8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn58)	32	RW	00000000h
30ech	Enhanced Rx FIFO Filter Element Registers (ERFFELn59)	32	RW	00000000h
30f0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn60)	32	RW	00000000h
30f4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn61)	32	RW	00000000h
30f8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn62)	32	RW	00000000h
30fch	Enhanced Rx FIFO Filter Element Registers (ERFFELn63)	32	RW	00000000h
3100h	Enhanced Rx FIFO Filter Element Registers (ERFFELn64)	32	RW	00000000h
3104h	Enhanced Rx FIFO Filter Element Registers (ERFFELn65)	32	RW	00000000h
3108h	Enhanced Rx FIFO Filter Element Registers (ERFFELn66)	32	RW	00000000h
310ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn67)	32	RW	00000000h
3110h	Enhanced Rx FIFO Filter Element Registers (ERFFELn68)	32	RW	00000000h
3114h	Enhanced Rx FIFO Filter Element Registers (ERFFELn69)	32	RW	00000000h
3118h	Enhanced Rx FIFO Filter Element Registers (ERFFELn70)	32	RW	00000000h
311ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn71)	32	RW	00000000h
3120h	Enhanced Rx FIFO Filter Element Registers (ERFFELn72)	32	RW	00000000h
3124h	Enhanced Rx FIFO Filter Element Registers (ERFFELn73)	32	RW	00000000h
3128h	Enhanced Rx FIFO Filter Element Registers (ERFFELn74)	32	RW	00000000h
312ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn75)	32	RW	00000000h
3130h	Enhanced Rx FIFO Filter Element Registers (ERFFELn76)	32	RW	00000000h
3134h	Enhanced Rx FIFO Filter Element Registers (ERFFELn77)	32	RW	00000000h
3138h	Enhanced Rx FIFO Filter Element Registers (ERFFELn78)	32	RW	00000000h
313ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn79)	32	RW	00000000h
3140h	Enhanced Rx FIFO Filter Element Registers (ERFFELn80)	32	RW	00000000h
3144h	Enhanced Rx FIFO Filter Element Registers (ERFFELn81)	32	RW	00000000h
3148h	Enhanced Rx FIFO Filter Element Registers (ERFFELn82)	32	RW	00000000h
314ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn83)	32	RW	00000000h
3150h	Enhanced Rx FIFO Filter Element Registers (ERFFELn84)	32	RW	00000000h
3154h	Enhanced Rx FIFO Filter Element Registers (ERFFELn85)	32	RW	00000000h
3158h	Enhanced Rx FIFO Filter Element Registers (ERFFELn86)	32	RW	00000000h
315ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn87)	32	RW	00000000h
3160h	Enhanced Rx FIFO Filter Element Registers (ERFFELn88)	32	RW	00000000h
3164h	Enhanced Rx FIFO Filter Element Registers (ERFFELn89)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3168h	Enhanced Rx FIFO Filter Element Registers (ERFFELn90)	32	RW	00000000h
316ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn91)	32	RW	00000000h
3170h	Enhanced Rx FIFO Filter Element Registers (ERFFELn92)	32	RW	00000000h
3174h	Enhanced Rx FIFO Filter Element Registers (ERFFELn93)	32	RW	00000000h
3178h	Enhanced Rx FIFO Filter Element Registers (ERFFELn94)	32	RW	00000000h
317ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn95)	32	RW	00000000h
3180h	Enhanced Rx FIFO Filter Element Registers (ERFFELn96)	32	RW	00000000h
3184h	Enhanced Rx FIFO Filter Element Registers (ERFFELn97)	32	RW	00000000h
3188h	Enhanced Rx FIFO Filter Element Registers (ERFFELn98)	32	RW	00000000h
318ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn99)	32	RW	00000000h
3190h	Enhanced Rx FIFO Filter Element Registers (ERFFELn100)	32	RW	00000000h
3194h	Enhanced Rx FIFO Filter Element Registers (ERFFELn101)	32	RW	00000000h
3198h	Enhanced Rx FIFO Filter Element Registers (ERFFELn102)	32	RW	00000000h
319ch	Enhanced Rx FIFO Filter Element Registers (ERFFELn103)	32	RW	00000000h
31a0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn104)	32	RW	00000000h
31a4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn105)	32	RW	00000000h
31a8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn106)	32	RW	00000000h
31ach	Enhanced Rx FIFO Filter Element Registers (ERFFELn107)	32	RW	00000000h
31b0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn108)	32	RW	00000000h
31b4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn109)	32	RW	00000000h
31b8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn110)	32	RW	00000000h
31bch	Enhanced Rx FIFO Filter Element Registers (ERFFELn111)	32	RW	00000000h
31c0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn112)	32	RW	00000000h
31c4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn113)	32	RW	00000000h
31c8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn114)	32	RW	00000000h
31cch	Enhanced Rx FIFO Filter Element Registers (ERFFELn115)	32	RW	00000000h
31d0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn116)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
31d4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn117)	32	RW	00000000h
31d8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn118)	32	RW	00000000h
31dch	Enhanced Rx FIFO Filter Element Registers (ERFFELn119)	32	RW	00000000h
31e0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn120)	32	RW	00000000h
31e4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn121)	32	RW	00000000h
31e8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn122)	32	RW	00000000h
31ech	Enhanced Rx FIFO Filter Element Registers (ERFFELn123)	32	RW	00000000h
31f0h	Enhanced Rx FIFO Filter Element Registers (ERFFELn124)	32	RW	00000000h
31f4h	Enhanced Rx FIFO Filter Element Registers (ERFFELn125)	32	RW	00000000h
31f8h	Enhanced Rx FIFO Filter Element Registers (ERFFELn126)	32	RW	00000000h
31fch	Enhanced Rx FIFO Filter Element Registers (ERFFELn127)	32	RW	00000000h

20.3.3.1 CAN MCR Register

CAN_MCR defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MDIS	FRZ	RFEN	HALT	NOTRDY	WAKMSK	SOFTRST	FRZACK	SUPV	SLFWAK	WRNEN	LPMACK	WAKSRC	DOZE	SRXDIS	IRMQ
W																
Reset	1	1	0	1	1	0	0	0	1	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DMA	PNET_EN	LPRIOEN	AEN	FDEN		IDAM			MAXMB						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Yuntu Microelectronics

Table 20.5: CAN MCR Register Description

Field	Function
31 MDIS	**Module Disable - Disables or enables the FlexCAN module.** When disabled, the FlexCAN disables the clocks to the PE and CHI submodules. MDIS is not affected by soft reset. 0b - Enable the FlexCAN module 1b - Disable the FlexCAN module
30 FRZ	**Freeze Enable** The FRZ bit specifies the FlexCAN behavior when CAN_MCR.HALT is set or when Debug mode is requested at chip level. When FRZ is asserted, the FlexCAN is enabled to enter Freeze mode. Negation of FRZ causes the FlexCAN to exit from Freeze mode. FRZ is set by hardware when a non-correctable error is detected and CAN_MECR.NCEFAFRZ is asserted. 0b - Not enabled to enter freeze mode 1b - Enabled to enter freeze mode
29 RFEN	**Legacy Rx FIFO Enable - Enables or disables the legacy Rx FIFO feature.** When RFEN is set, MBs 0-5 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xDC) is used by the FIFO engine and additional MBs (up to 32, depending on CAN_CTRL2.RFFN setting), which are used as Legacy Rx FIFO ID Filter Table elements. RFEN also impacts the minimum number of peripheral clocks per CAN bit. RFEN can be written in Freeze mode only because it is blocked by hardware in other modes. NOTE: RFEN cannot be set when CAN FD operation is enabled (see FDEN bit). 0b - Disable Rx FIFO 1b - Enable Rx FIFO.
28 HALT	**Halt FlexCAN** Assertion of HALT puts the FlexCAN module into Freeze mode. The CPU should clear HALT after initializing the message buffers and the control registers CAN_CTRL1 and CAN_CTRL2. No reception or transmission is performed by the FlexCAN before HALT is cleared. Freeze mode cannot be entered when the FlexCAN is in a low-power mode. The HALT bit is set by hardware when a non-correctable error is detected and CAN_MECR.NCEFAFRZ is asserted. 0b - No Freeze mode request 1b - Enter Freeze mode if the FRZ bit is asserted
27 NOTRDY	**FlexCAN Not Ready** NOTRDY is a read-only bit that indicates that the FlexCAN is either in Module Disable, Doze, Deepsleep, or Freeze mode. NOTRDY is negated when the FlexCAN has exited these modes. NOTRDY bit is not affected by soft reset 0b - FlexCAN is either in Normal, Listen-Only or Loop-Back mode 1b - FlexCAN is either in Module Disable, Doze, Deepsleep or Freeze mode

Yuntu Microelectronics

Field	Function
26 WAKMSK	**FlexCAN Wake Up Interrupt Mask** Enables Wake Up interrupt generation under the Self Wake Up mechanism: 0b - Disable Wake Up interrupt 1b - Enable Wake Up interrupt
25 SOFTRST	**FlexCAN Software Reset** When SOFTRST is asserted, the FlexCAN resets its internal state machines and some of the memory-mapped registers. SOFTRST can be asserted directly by the CPU by writing to CAN_MCR. SOFTRST is also asserted when global soft reset is requested at chip level. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFTRST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll SOFTRST to know when the soft reset has completed. Soft reset cannot be applied when clocks are shut down in a low-power mode. The FlexCAN should be first removed from low-power mode, then soft reset can be applied. SOFTRST is not affected by soft reset 0b - No reset request 1b - Reset the registers affected by software reset
24 FRZACK	**Freeze Mode Acknowledge** FRZACK is a read-only bit that indicates that the FlexCAN is in Freeze mode and its prescaler is stopped. A Freeze mode request cannot be granted until current transmission or reception processes have finished. Therefore, software can poll FRZACK to know when the FlexCAN has actually entered Freeze mode. If the Freeze mode request is negated, FRZACK is negated after the FlexCAN prescaler is running again. If Freeze mode is requested when the FlexCAN is in a low-power mode, FRZACK will be set only when the low-power mode is exited. See “Freeze Mode”. FRZACK is not affected by soft reset. NOTE: FRZACK is asserted within 178 CAN bits from the Freeze mode request by the CPU, and negated within 2 CAN bits after the Freeze mode request removal (see “Protocol Timing”). 0b - FlexCAN not in Freeze Mode, prescaler running 1b - FlexCAN in Freeze Mode, prescaler stopped
23	**Supervisor Mode - Configures the FlexCAN to be either in Supervisor or User mode.**

Yuntu Microelectronics

Field	Function
SUPV	The registers affected by SUPV are marked as S/U in the Access Type column of Table: “CAN register access and reset information”. The SUPV reset value is 1, so the affected registers start with Supervisor access only. SUPV can be written only in Freeze mode because it is blocked by hardware in other modes. 1b - FlexCAN is in Supervisor Mode, Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location. 0b - FlexCAN is in User mode, Affected registers allow both Supervisor and Unrestricted accesses
22 SLFWAK	**Self Wake Up - Enables the Self Wake Up feature when the FlexCAN is in a low-power mode other than Module Disable.** When the Self Wake Up feature is enabled, the FlexCAN module monitors the bus for wakeup event (a recessive-to-dominant transition). If a wakeup event is detected during Deepsleep mode, the FlexCAN, if enabled to do so, generates a Wake Up interrupt to the CPU so that it can exit Deepsleep mode globally and the FlexCAN can request to resume the clocks. When FlexCAN is in a low-power mode other than Module Disable, SLFWAK cannot be written as it is blocked by hardware. When Pretended Networking mode is set, Self Wake Up must be disabled. 1b - Enable FlexCAN self Wake Up feature. 0b - Disable FlexCAN self Wake Up feature.
21 WRNEN	**Warning Interrupt Enable** Assertion of WRNEN enables the generation of the TWRNINT and RWRNINT flags in the Error and Status Register 1 (CAN_ESR1). If WRNEN is negated, the TWRNINT and RWRNINT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. WRNEN can be written only in Freeze mode because it is blocked by hardware in other modes. 1b - TWRNINT and RWRNINT bits are set when the respective error counter transitions from less than 96 to greater than or equal to 96 0b - TWRNINT and RWRNINT bits are zero, independent of the values in the error counters
20	**Low-Power Mode Acknowledge**

Yuntu Microelectronics

Field	Function
LPMACK	LPMACK is a read-only bit that indicates that the FlexCAN is in a low-power mode (Module Disable, Doze or Deepsleep). A low-power mode cannot be entered until all current transmission or reception processes have finished. Therefore, the CPU can poll LPMACK to know when the FlexCAN has actually entered low-power mode. LPMACK is not affected by soft reset. NOTE: LPMACK is asserted within 180 CAN bits from the low-power mode request by the CPU, and negated within 2 CAN bits after the low-power mode request removal (see "Protocol Timing"). When the FlexCAN is in Pretended Networking mode, LPMACK is negated within 180 CAN bits after the low-power mode request removal. 1b - FlexCAN is in the low-power mode 0b - FlexCAN is not in the low-power mode
19 WAKSRC	**Wake Up Source - Controls whether the integrated low-pass filter is applied to protect the Rx CAN input from spurious Wake Up.** WAKSRC can be written only in Freeze mode because it is blocked by hardware in other modes. 1b - FlexCAN uses the filtered RX input to detect recessive to dominant edges on the can bus 0b - FlexCAN uses the unfiltered RX input to detect recessive to dominant edges on the can bus
18 DOZE	**Doze Mode Enable - Controls whether the FlexCAN is allowed to enter low-power mode when Doze mode is requested at chip level.** DOZE is automatically reset when the FlexCAN wakes up from Doze mode upon detecting activity on the CAN bus (Self Wake Up enabled). 1b - FlexCAN is enabled to enter low-power mode when Doze mode is requested. 0b - FlexCAN is not enabled to enter low-power mode when Doze mode is requested.
17 SRXDIS	**Self Reception Disable** SRXDIS controls whether the FlexCAN is allowed to receive frames transmitted by itself. If SRXDIS is asserted, frames transmitted by the FlexCAN are not stored in any MB, regardless of whether the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal is generated due to the frame reception. SRXDIS can be written only in Freeze mode because it is blocked by hardware in other modes 1b - Disable self-reception. 0b - Enable self-reception.
16	**Individual Rx Masking And Queue Enable**

Yuntu Microelectronics

Field	Function
IRMQ	IRMQ controls whether Rx matching process is based on individual masking and queue or on masking scheme with CAN_RXMGMASK, CAN_RX14MASK, CAN_RX15MASK, and CAN_RXFGMASK. IRMQ can be written in Freeze mode only because it is blocked by hardware in other modes. 1b - Enable individual Rx masking and queue feature. 0b - Disable individual Rx masking and queue feature. For backward compatibility with legacy applications, the reading of C/S word locks the MB even if it is EMPTY
15 DMA	**DMA Enable** The DMA bit enables or disables the DMA feature. The DMA feature can only be used with the Legacy Rx FIFO or the Enhanced Rx FIFO; consequently, either CAN_MCR.RFEN or CAN_ERFCR.ERFEN must be asserted. When DMA and RFEN are both set, CAN_IFLAG1.BUF5I generates the DMA request and no Legacy Rx FIFO interrupt is generated. When DMA and CAN_ERFCR.ERFEN are both set, CAN_ERFSR.ERFWMI generates the DMA request. The DMA bit can be written in Freeze mode only as it is blocked by hardware in other modes. The DMA bit is not available(reserved, write 0) if parameter DMA_EN is 0 1b - Enable DMA feature for Rx FIFO. 0b - Disable DMA feature for Rx FIFO.
14 PNET_EN	**Pretended Networking Enable** When Pretended Networking is enabled, the PE submodule is kept operational in Doze mode and Deepsleep mode, so that it is able to process Rx message filtering as defined by the Pretended Networking configuration registers. See "Receive Process under Pretended Networking Mode". PNET_EN can be written in Freeze mode only. The PNET_EN bit is not available (reserved, write 0) if parameter PNET_EN is 0. 1b - Enable Pretended Networking mode. 0b - Disable Pretended Networking mode.
13 LPRIOEN	**Local Priority Enable** The LPRIOEN bit is provided for backward compatibility with legacy applications. LPRIOEN controls whether the Local Priority feature is enabled or not. Local Priority is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11 bits for standard frames and 29 bits for extended frames. LPRIOEN can be written only in Freeze mode because it is blocked by hardware in other modes.
12	**Abort Enable - Enables the Tx abort mechanism.**

Yuntu Microelectronics

Field	Function
AEN	The Tx abort mechanism ensures a safe procedure for aborting a pending transmission, so that no frame is sent on the CAN bus without notification. See "Transmission Abort Mechanism". AEN can be written only in Freeze mode because it is blocked by hardware in other modes. NOTE: When AEN is asserted, the abort mechanism must only be used for updating mailboxes configured for transmission. Caution: Writing the Abort code into Rx mailboxes can cause unpredictable results when AEN is asserted. 1b - Enable abort 0b - Disable abort.
11 FDEN	**CAN FD Operation Enable –Enables CAN with Flexible Data rate (CAN FD) operation.** FDEN can be written in Freeze mode only. The FEDN bit is not available (reserved, write 0) if parameter FD_EN is 0. NOTE: The Rx FIFO Enable (RFEN) bit cannot be set if FDEN is asserted. 1b - CAN FD is enabled. The FlexCAN is able to receive and transmit messages in both CAN FD and CAN 2.0 formats. 0b - CAN FD is disabled. The FlexCAN is able to receive and transmit messages in CAN 2.0 format.
9 - 8 IDAM	**ID Acceptance Mode –Identifies the format of the Rx FIFO ID Filter Table elements.** All elements of the table are configured at the same time by IDAM (they are all the same format). See "Legacy Rx FIFO Structure". IDAM can be written only in Freeze mode because it is blocked by hardware in other modes. 00b - Format A –One full ID (standard and extended) per ID Filter Table element 01b - Format B –Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID Filter Table element 10b - Format C –Four partial 8-bit standard IDs per ID Filter Table element 11b - Format D –All frames rejected.
6 - 0	**Number of the Last Message Buffer –Defines the number of the last message buffer that will take part in the matching and arbitration processes.**

Yuntu Microelectronics

Field	Function
MAXMB	The reset value (0x0F) is equivalent to a 16 message buffer configuration. MAXMB can be written only in Freeze mode because it is blocked by hardware in other modes. Number of the last MB = MAXMB NOTE: MAXMB must be programmed with a value smaller than or equal to the number of available message buffers, as described in "FlexCAN Memory Partition for CAN FD". Additionally, the definition of MAXMB value must take into account the region of MBs occupied by Rx FIFO and its ID filters table space defined by CAN_CTRL2.RFFN. MAXMB also impacts the minimum number of peripheral clocks per CAN bit as described in the FlexCAN Integration Guide.

20.3.3.2 CAN CTRL1 Register

The CAN_CTRL1 register is defined for specific FlexCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop-Back mode, Listen-Only mode, Bus Off recovery behavior, and interrupt enabling (Bus Off, Error, Warning). It also determines the division factor for the clock prescaler.

The CAN bit timing variables (PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW) can also be configured in CAN_CBT register, which extends the range of all these variables. If CAN_CBT.BTF is set, the PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields of CAN_CTRL1 become read-only.

When CAN_CTRL2.BTE = 1, the PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields in CAN_CTRL1 are not used. They are read as zero and a write operation to them has no effect.

NOTE: When the CAN FD feature is enabled, do not use the PRESDIV, RJW, PSEG1, PSEG2, and PROPSEG fields of the CAN_CTRL1 register for CAN bit timing. Instead, use the CAN_CBT register's EPRESDIV, ERJW, EPSEG1, EPSEG2, and EPROPSEG fields.

The contents of the CAN_CTRL1 register are not affected by soft reset.

NOTE: The CAN bit variables in CAN_CTRL1 and in CAN_CBT are stored in the same register.

Offset: 4h

Yuntu Microelectronics

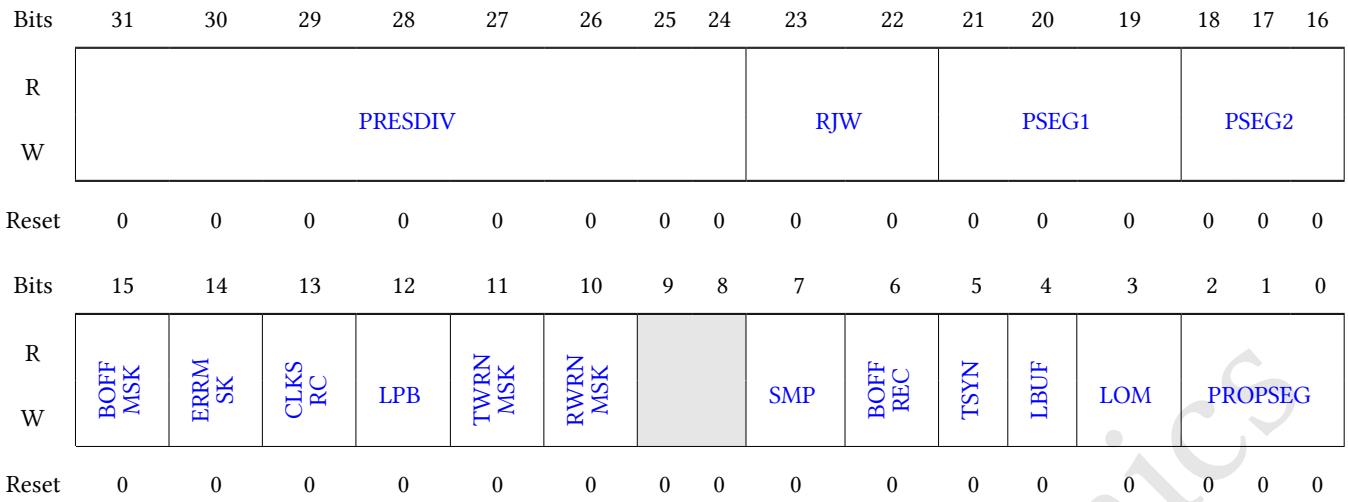


Table 20.6: CAN CTRL1 Register Description

Field	Function
31 - 24 PRESDIV	**Prescaler Division Factor –Defines the ratio between the PE clock frequency and the serial clock (Sclock) frequency.** The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency. The maximum value of PRESDIV is 0xFF, which gives a minimum Sclock frequency equal to the PE clock frequency divided by 256. See section "Protocol Timing". PRESDIV can be written only in Freeze mode because it is blocked by hardware in other modes. Sclock frequency = PE clock frequency / (PRESDIV + 1)
23 - 22 RJW	**Resync Jump Width –Defines the maximum number of time quanta that a bit time can be changed by one re-synchronization.** One time quantum is equal to the Sclock period. The valid programmable values for RJW are 0–3. RJW can be written only in Freeze mode because it is blocked by hardware in other modes. Resync Jump Width = RJW + 1
21 - 19 PSEG1	**Phase Segment 1 –Defines the length of phase segment 1 in the bit time.** The valid programmable values are 0–7. PSEG1 can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 1 = (PSEG1 + 1) × Time-Quanta
18 - 16 PSEG2	**Phase Segment 2 –Defines the length of phase segment 2 in the bit time.** The valid programmable values are 1–7. PSEG2 can be written only in Freeze mode because it is blocked by hardware in other modes. Phase Buffer Segment 2 = (PSEG2 + 1) × Time-Quanta
15	**Bus Off Interrupt Mask –Provides a mask for the Bus Off interrupt (CAN_ESR1.BOFFINT).**

Yuntu Microelectronics

Field	Function
BOFFMSK	0b - Bus Off interrupt disabled. 1b - Bus Off interrupt enabled.
14 ERRMSK	**Error Interrupt Mask –Provides a mask for the Error interrupt (CAN_ESR1.ERRINT).** 0b - Error interrupt disabled. 1b - Error interrupt enabled.
13 CLKSRC	**CAN Engine Clock Source –Selects the clock source to the PE submodule to be either the peripheral clock or the oscillator clock.** The selected clock is fed to the prescaler to generate the serial clock (Sclock). To ensure reliable operation, CLKSRC can be written only in Disable mode. In other modes, writing to CLKSRC is blocked by hardware. See "Protocol Timing". 0b - The PE clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the peripheral clock frequency. 1b - The PE clock source is the peripheral clock.
12 LPB	**Loop-Back Mode –Configures the FlexCAN to operate in Loop-Back mode.** In Loop-Back mode, the FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is fed back internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). The FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In Loop-Back mode, the FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. LPB can be written only in Freeze mode because it is blocked by hardware in other modes. NOTE: In Loop-Back mode, CAN_MCR.SRXDIS cannot be asserted because it would impede the self-reception of a transmitted message. NOTE: CAN_FDCTRL.TDCEN must be 0 (Transceiver Delay Compensation feature disabled) when LPB is asserted. 0b - Loop-Back disabled. 1b - Loop-Back enabled.
11 TWRNMSK	**Tx Warning Interrupt Mask –Provides a mask for the Tx Warning interrupt (CAN_ESR1.TWRNINT).** TWRNMSK is read as zero when CAN_MCR.WRNEN is negated. TWRNMSK can be written only if CAN_MCR.WRNEN is asserted 0b - Tx Warning interrupt disabled. 1b - Tx Warning interrupt enabled.
10	**Rx Warning Interrupt Mask –Provides a mask for the Rx Warning interrupt (CAN_ESR1.RWRNINT).**

Yuntu Microelectronics

Field	Function
RWRNMSK	RWRNMSK is read as zero when CAN_MCR.WRNEN is negated. RWRNMSK can be written only if CAN_MCR.WRNEN is asserted. 0b - Rx Warning interrupt disabled. 1b - Rx Warning interrupt enabled.
7 SMP	**CAN Bit Sampling –Defines the sampling mode of CAN bits at the Rx input.** SMP can be written only in Freeze mode because it is blocked by hardware in other modes. NOTE: For proper operation, to assert SMP it is necessary to guarantee a minimum value of two time quanta in CAN_CTRL1.PSEG1 (or CAN_CBT.EPSEG1). SMP cannot be asserted when CAN FD is enabled (CAN_MCR.FDEN = 1). 0b - One sample is used to determine the bit value. 1b - Three samples are used to determine the value of the received bit, the regular sample (sample point) and two preceding samples; a majority rule is used.
6 BOFFREC	**Bus Off Recovery –Defines how the FlexCAN recovers from Bus Off state.** If BOFFREC is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If BOFFREC is asserted, automatic recovering from Bus Off is disabled and the FlexCAN module remains in Bus Off state until BOFFREC is negated by software. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFFREC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then the FlexCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, BOFFREC can be re-asserted again during Bus Off, but it will be effective only the next time the FlexCAN module enters Bus Off. If BOFFREC was negated when the FlexCAN module entered Bus Off, asserting BOFFREC during Bus Off will not be effective for the current Bus Off recovery. 0b - Automatic recovering from Bus Off state enabled. 1b - Automatic recovering from Bus Off state disabled.
5 TSYN	**Timer Sync –Enables a mechanism that resets the free-running timer each time a message is received in message buffer 0.** The Timer Sync feature provides a means to synchronize multiple FlexCAN stations with a special ”SYNC” message (global network time). If CAN_MCR.RFEN is set (Legacy Rx FIFO enabled), the first available mailbox, according to the CAN_CTRL2.RFFN setting, is used for timer synchronization instead of MB0. TSYN can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Timer Sync feature disabled. 1b - Timer Sync feature enabled.
4	**Lowest Buffer Transmitted First –Defines the ordering mechanism for message buffer transmission.**

Yuntu Microelectronics

Field	Function
LBUF	When LBUF is asserted, the CAN_MCR.LPRIOEN bit does not affect the priority arbitration. LBUF can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Buffer with highest priority is transmitted first. 1b - Lowest number buffer is transmitted first.
3 LOM	**Listen-Only Mode –Configures the FlexCAN to operate in Listen-Only mode.** In Listen-Only mode, transmission is disabled, all error counters described in CAN_ECR are frozen and the FlexCAN module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If the FlexCAN detects a message that has not been acknowledged, it flags a BIT0 error without changing the receive error counter (RXERRCNT) in CAN_ECR, as if it were trying to acknowledge the message. Listen-Only mode is acknowledged by the state of the CAN_ESR1.FLTCONF field indicating Passive Error. There can be some delay between the Listen-Only mode request and acknowledge. LOM can be written in Freeze mode only because it is blocked by hardware in other modes. 0b - Listen-Only mode is deactivated. 1b - FlexCAN module operates in Listen-Only mode.
2 - 0 PROPSEG	**Propagation Segment –Defines the length of the propagation segment in the bit time.** The valid programmable values are 0–7. PROPSEG can be written only in Freeze mode because it is blocked by hardware in other modes. Propagation segment time = (PROPSEG + 1) × time-quanta. Time-quantum = one Sclock period.

20.3.3.3 CAN TIMER Register

The CAN_TIMER register represents a 16-bit free-running counter that can be read and written by the CPU. The timer starts from 0x0 after reset, counts linearly to 0xFFFF, and wraps around.
When CAN_CTRL2.TIMER_SRC is asserted, the timer is continuously incremented by an external time tick. The time tick must be synchronous to the peripheral clock, with a minimum pulse width of one ipg_clk cycle.
When CAN_CTRL2.TIMER_SRC is negated, the timer is incremented by the CAN bit clock, which defines the baud rate on the CAN bus. During a message transmission/reception, the timer increments by one for each bit that is received or transmitted. When there is no message on the bus, the timer counts using the previously programmed baud rate. The timer is not incremented during Module Disable, Doze, Deepsleep, and Freeze modes.
The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.
If CAN_CTRL1.TSYN is asserted, the timer is reset whenever a message is received in the first available mailbox, according to the CAN_CTRL2.RFFN setting.
The CPU can write to this register anytime. However, if the write occurs at the same time that the timer is being reset by a reception in the first mailbox, then the write value is discarded.
Reading this register affects the mailbox unlocking procedure, see “Mailbox Lock Mechanism”.

Yuntu Microelectronics

Offset: 8h

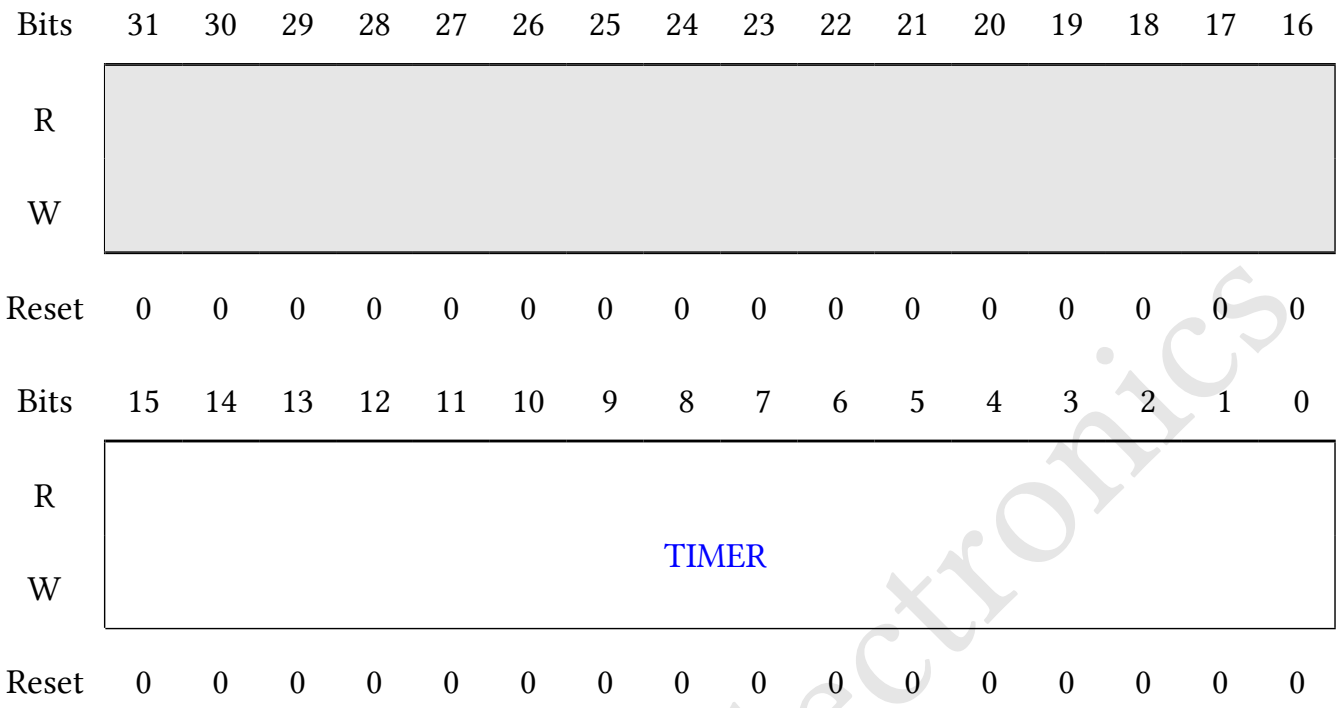


Table 20.7: CAN TIMER Register Description

Field	Function
15 - 0 TIMER	**Timer Value –Contains the free-running counter value.**

20.3.3.4 CAN RXMGMASK Register

The CAN_RXMGMASK register is located in RAM.

CAN_RXMGMASK is provided for legacy application support:

- When the CAN_MCR.IRMQ bit is negated, CAN_RXMGMASK is always in effect (the bits in the MG field will mask the mailbox filter bits).
- When the CAN_MCR.IRMQ bit is asserted, CAN_RXMGMASK has no effect (the bits in the MG field will not mask the mailbox filter bits).

CAN_RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14–15, which have individual mask registers.

CAN_RXMGMASK can only be written in Freeze mode as it is blocked by hardware in other modes.

Yuntu Microelectronics

Table 20.8: CAN Which MG Bits Mask Each Mailbox Filter Field

SMB.RTR[1]	CAN_CTRL2.RRS	CAN_CTRL2.EACEN	Mailbox Filter Feilds			
			MB.RTR	MB.IDE	MB.ID	Reserved
0	-	0	not [2]	note [3]	MG[28:0]	MG[31:29]
0	-	1	MG[31]	MG[30]	MG[28:0]	MG[29]
1	0	-	-	-	-	MG[31:0]
1	1	0	-	-	MG[28:0]	MG[31:29]
1	1	1	MG[31]	MG[30]	MG[28:0]	MG[29]

1. RTR bit of the incoming frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If CAN_CTRL2.EACEN is negated, the RTR bit of the mailbox is never compared with the RTR bit of the incoming frame.
3. If CAN_CTRL2.EACEN is negated, the IDE bit of the mailbox is always compared with the IDE bit of the incoming frame.

Offset: 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.9: CAN RXMGMASK Register Description

Field	Function
31 - 0	**Rx Mailboxes Global Mask Register**

Yuntu Microelectronics

Field	Function
MG	The MG bits mask the mailbox filter bits. Note that the alignment with the ID word of the mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status (C/S) word of the mailbox. Details see table CAN which MG bits mask each mailbox filter field 0b - The corresponding bit in the filter is “don’t care”. 1b - The corresponding bit in the filter is checked.

20.3.3.5 CAN RX14MASK Register

The CAN_RX14MASK register is located in RAM.

CAN_RX14MASK is provided for legacy application support. When the CAN_MCR.IRMQ bit is asserted, CAN_RX14MASK has no effect.

CAN_RX14MASK is used to mask the filter fields of message buffer 14.

CAN_RX14MASK can only be programmed when the FlexCAN is in Freeze mode as it is blocked by hardware in other modes.

Offset: 14h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX14M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX14M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.10: CAN RX14MASK Register Description

Field	Function
31 - 0	**Rx 14 Mask Register**

Yuntu Microelectronics

Field	Function
RX14M	Each mask bit masks the corresponding mailbox 14 filter field in the same way that CAN_RXMGMASK masks other mailboxes' filters. See the description of the CAN_RXMGMASK register. 0b - The corresponding bit in the filter is "don't care". 1b - The corresponding bit in the filter is checked.

20.3.3.6 CAN RX15MASK Register

The CAN_RX15MASK register is located in RAM.

CAN_RX15MASK is provided for legacy application support. When the CAN_MCR.IRMQ bit is asserted, CAN_RX15MASK has no effect.

CAN_RX15MASK is used to mask the filter fields of message buffer 15.

CAN_RX15MASK can be programmed only when the FlexCAN is in Freeze mode because it is blocked by hardware in other modes.

Offset: 18h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RX15M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RX15M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.11: CAN RX15MASK Register Description

Field	Function
31 - 0	**Rx Buffer 15 Mask Bits**

Yuntu Microelectronics

Field	Function
RX15M	Each mask bit masks the corresponding mailbox 15 filter field in the same way that CAN_RXMGMASK masks other mailboxes' filters. See the description of the CAN_RXMGMASK register. 0b - The corresponding bit in the filter is "don't care". 1b - The corresponding bit in the filter is checked.

20.3.3.7 CAN ECR Register

The CAN_ECR register has four 8-bit fields reflecting the value of the FlexCAN error counters:

- Transmit Error Counter (TXERRCNT field)
- Receive Error Counter (RXERRCNT field)
- Transmit Error Counter for errors detected in the Data Phase of CAN FD messages with the BRS bit set (TXERRCNT_FAST field)
- Receive Error Counter for errors detected in the Data Phase of CAN FD messages with the BRS bit set (RXERRCNT_FAST field)

The TXERRCNT and RXERRCNT counters take into account all errors in both CAN FD and non-FD message formats. TXERRCNT_FAST and RXERRCNT_FAST are dedicated to count only the errors that occur in the Data Phase of CAN FD frames with the BRS bit set.

The Fault Confinement State (FLTCONF field in CAN_ESR1) is updated based on the TXERRCNT and RXERRCNT counters only. The TXERRCNT and RXERRCNT counters can be written in Freeze mode only. The TXERRCNT_FAST and RXERRCNT_FAST counters are read-only except in Freeze mode, in which the CPU can write value zero. The rules for incrementing and decrementing these counters are described in the CAN protocol and are completely implemented in the FlexCAN module.

The basic rules for FlexCAN bus state transitions are:

- If the value of TXERRCNT or RXERRCNT increases to be greater than or equal to 128, the CAN_ESR1.FLTCONF field is updated to reflect the Error Passive state.
- If the FlexCAN state is Error Passive, and either TXERRCNT or RXERRCNT decrements to a value less than or equal to 127 while the other already satisfies this condition, the CAN_ESR1.FLTCONF field is updated to reflect the Error Active state.
- If the value of TXERRCNT increases to be greater than 255, the CAN_ESR1.FLTCONF field is updated to reflect the Bus Off state, CAN_ESR1.BOFFINT is set, and an interrupt is issued if CAN_CTRL1.BOFFMSK=1. The value of TXERRCNT is then reset to zero.
- If the FlexCAN is in the Bus Off state, then TXERRCNT is cascaded together with another internal counter to count the 128 occurrences of 11 consecutive recessive bits on the bus. Hence, TXERRCNT is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing TXERRCNT. When TXERRCNT reaches the value of 128, the CAN_ESR1.FLTCONF field is updated to be Error Active and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value. The TXERRCNT_FAST counter is frozen during Bus Off.
- If during system startup, only one node is operating, then its TXERRCNT increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the

Yuntu Microelectronics

CAN_ESR1.ACKERR bit). After the transition to the Error Passive state, TXERRCNT is no longer incremented by acknowledge errors. Therefore, the device never goes to the Bus Off state.

- If RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected while being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to return to the Error Active state.
- The TXERRCNT_FAST and RXERRCNT_FAST error counter values increment and decrement based on errors detected only in the Data Phase of CAN FD frames with the BRS bit set, following the same increment and decrement rules as the TXERRCNT and RXERRCNT counters. These counters do not wrap around and get stuck at their maximum value (255). They stop counting and keep their values frozen when the FlexCAN is in the Bus Off state. They are reset when the FlexCAN leaves the Bus Off state and restart counting after the FlexCAN returns to the Error Active state.
- When the FlexCAN is in Pretended Networking mode, RXERRCNT and RXERRCNT_FAST keep counting errors and error flags are stored, TXERRCNT and TXERRCNT_FAST preserve their values and do not change since no transmission occurs under Pretended Networking mode are updated in CAN_ECR and CAN_ESR1 when the FlexCAN returns to Normal mode. The FAST error flags in CAN_ESR1 are not set if the FlexCAN is in Pretended Networking mode.

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RXERRCNT_FAST								TXERRCNT_FAST							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RXERRCNT								TXERRCNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.12: CAN ECR Register Description

Field	Function
31 - 24	**Receive Error Counter for Fast Bits**

Yuntu Microelectronics

Field	Function
RXERRCNT _FAST	Receive Error Counter for Fast Bits –Receive error counter for errors detected in the Data Phase of received CAN FD messages with the BRS bit set. The RXERRCNT_FAST counter is read-only except in Freeze mode, in which the CPU can write an 8-bit zero value only. RXERRCNT_FAST is not available (reserved) when configuration parameter FD_EN is 0.
23 - 16 TXERRCNT _FAST	**Transmit Error Counter for Fast Bits** Transmit Error Counter for Fast Bits –Transmit error counter for errors detected in the Data Phase of transmitted CAN FD messages with the BRS bit set. The TXERRCNT_FAST counter is read-only except in Freeze mode, in which the CPU can write a 8-bit zero value only. TXERRCNT_FAST is not available (reserved) when configuration parameter FD_EN is 0.
15 - 8 RXERRCNT	**Receive Error Counter** Receive Error Counter –Receive error counter for all errors detected in received messages. The RXERRCNT counter is read-only except in Freeze mode, in which it can be written by the CPU.
7 - 0 TXERRCNT	**Transmit Error Counter** Transmit Error Counter –Transmit error counter for all errors detected in transmitted messages. The TXERRCNT counter is read-only except in Freeze mode, in which it can be written by the CPU.

20.3.3.8 CAN ESR1 Register

The CAN_ESR1 register reports various error conditions detected in the reception and transmission of a CAN frame, some general status of the device, and is the source of some interrupts to the CPU.

The reported error conditions are:

- BIT1ERR, BIT0ERR, ACKKERR, CRCERR, FRMERR, and STFERR, for errors detected in CAN frames of any format.
- BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST, and STFERR_FAST for errors detected in the Data Phase of CAN FD frames with the BRS bit set only.

An error detected in a single CAN frame may be reported by one or more error flags. Also, error reporting is cumulative, in case additional error events occur in subsequent frames before the CPU reads CAN_ESR1.

TXWRN, RXWRN, IDLE, TX, FLTCONF, RX and SYNCH are status bits.

BOFFINT, BOFFDONEINT, ERRINT, ERRINT_FAST, WAKINT, TWRNINT, and RWRNINT are interrupt bits. It is recommended that the CPU use the following procedure when servicing interrupt requests generated by these bits:

Yuntu Microelectronics

1. Read CAN_ESR1 to capture all error condition and status bits. This action clears the respective bits that were set since the last read access.
2. Write 1 to clear the interrupt bit that has triggered the interrupt request.
3. Write 1 to clear the ERR_OVR bit if it is set.

Starting from all error flags cleared, a first error event sets either the ERRINT or the ERRINT_FAST (provided the corresponding mask bit is asserted). If other error events in subsequent frames occur before the CPU serves the interrupt request, the ERR_OVR bit is set to indicate that errors from different frames have accumulated.

Table 20.13: CAN State Related Bit Encodings

SYNCH	IDLE	TX	RX	FlexCAN State
0	0	0	0	Not synchronized to CAN bus
1	1	x	x	Idle
1	0	1	0	Transmitting
1	0	0	1	Receiving

Offset: 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BIT1ERR_FAST	BIT0ERR_FAST		CRCERR_FAST	FRMERR_FAST	STFERR_FAST					ERROVR	ERRINT_FAST	BOFFDONEINT	SYNCH	TWRNINT	RWRNINT
W											W1C	W1C	W1C		W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BIT1ERR	BIT0ERR	ACKERR	CRCERR	FRMERR	STFERR	TXWRN	RXWRN	IDLE	TX	FLTCONF		RX	BOFFINT	ERRINT	WAKINT
W														W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.14: CAN ESR1 Register Description

Field	Function
31 BIT1ERR_FAST	**Bit 1 Error in the Data Phase of CAN FD Frames with BRS bit set.** BIT1ERR_FAST indicates when an inconsistency occurs between the transmitted and the received bit in the Data Phase of CAN FD frames with the BRS bit set. BIT1ERR_FAST is not available (reserved) when configuration parameter FD_EN is 0. 0b - No such occurrence. 1b - At least one bit sent as recessive is received as dominant.
30	**Bit 0 Error in the Data Phase of CAN FD Frames with BRS bit set.**

Yuntu Microelectronics

Field	Function
BIT0ERR_ FAST	BIT0ERR_FAST indicates when an inconsistency occurs between the transmitted and the received bit in the Data Phase of CAN FD frames with the BRS bit set. BIT0ERR_FAST is not available (reserved) when configuration parameter FD_EN is 0. 0b - No such occurrence. 1b - At least one bit sent as recessive is received as dominant.
28 CRCERR_F AST	**CRC Error in the Data Phase of CAN FD Frames with BRS bit set.** CRCERR_FAST indicates that a CRC error has been detected by the receiver node in the CRC field of CAN FD frames with the BRS bit set; that is, the calculated CRC is different from the received CRC. CRCERR_FAST is not available (reserved) when configuration parameter FD_EN is 0. 0b - No such occurrence. 1b - A CRC error occurred since last read of this register.
27 FRMERR_F AST	**Form Error in the Data Phase of CAN FD Frames with BRS bit set.** FRMERR_FAST indicates that a form error has been detected by the receiver node in the Data Phase of CAN FD frames with the BRS bit set; that is, a fixed-form bit field contains at least one illegal bit. FRMERR_FAST is not available (reserved) when configuration parameter FD_EN is 0. 0b - No such occurrence. 1b - A form error occurred since last read of this register.
26 STFERR_F AST	**Stuffing Error in the Data Phase of CAN FD Frames with BRS bit set.** STFERR_FAST indicates that a stuffing error has been detected in the Data Phase of CAN FD frames with the BRS bit set. STFERR_FAST is not available (reserved) when configuration parameter FD_EN is 0. 0b - No such occurrence. 1b - A stuffing error occurred since last read of this register.
21 ERROVR	**Error Overrun –Indicates that an error condition occurred when any error flag is already set.** ERROVR is cleared by writing 1 to it. 0b - Overrun has not occurred. 1b - Overrun has occurred.
20	**Error Interrupt for Errors Detected in the Data Phase of CAN FD Frames with the BRS Bit Set**

Yuntu Microelectronics

Field	Function
ERRINT_FAST	ERRINT_FAST indicates that at least one of the error bits detected in the Data Phase of CAN FD frames with the BRS bit set (BIT1ERR_FAST, BIT0ERR_FAST, CRCERR_FAST, FRMERR_FAST, or STFERR_FAST) is set. If the corresponding mask bit (CAN_CTRL2.ERRMSK_FAST) is set, an interrupt is generated to the CPU. ERRINT_FAST is cleared by writing 1 to it. Writing 0 has no effect. ERRINT_FAST is not available (reserved) when configuration parameter FD_EN is 0. 0b - No such occurrence. 1b - Indicates setting of any error bit detected in the Data Phase of CAN FD frames with the BRS bit set.
19 BOFFDONEINT	**Bus Off Done Interrupt** BOFFDONEINT is set when the Tx Error Counter (TXERRCNT) has finished counting 128 occurrences of 11 consecutive recessive bits on the CAN bus and is ready to leave Bus Off. If the corresponding mask bit (CAN_CTRL2.BOFFDONEMSK) is set, an interrupt is generated to the CPU. BOFFDONEINT is cleared by writing 1 to it. Writing 0 has no effect. 0b - No such occurrence. 1b - FlexCAN module has completed Bus Off process.
18 SYNCH	**CAN Synchronous Status** SYNCH is a read-only flag that indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. See CAN state related bit encodings 0b - FlexCAN is not synchronized to the CAN bus. 1b - FlexCAN is synchronized to the CAN bus.
17 TWRNINT	**TX Warning Interrupt Flag** If CAN_MCR.WRNEN is asserted, the TWRNINT bit is set when the TXWRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit (CAN_CTRL1.TWRNMSK) is set, an interrupt is generated to the CPU. TWRNINT is cleared by writing 1 to it. Writing 0 has no effect. When CAN_MCR.WRNEN is negated, the TWRNINT flag is masked. The CPU must clear the TWRNINT flag before negating CAN_MCR.WRNEN. Otherwise, TWRNINT will be set when CAN_MCR.WRNEN is set again. TWRNINT is not generated during the Bus Off state and is not updated during Freeze mode. When the FlexCAN returns to Normal mode from Pretended Networking mode, TWRNINT is not updated. See "Receive Process under Pretended Networking Mode". 0b - No such occurrence. 1b - The Tx error counter transitioned from less than 96 to greater than or equal to 96.
16	**RX Warning Interrupt Flag**

Yuntu Microelectronics

Field	Function
RWRNINT	If CAN_MCR.WRNEN is asserted, the RWRNINT bit is set when the RXWRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit (CAN_CTRL1.RWRNMSK) is set, an interrupt is generated to the CPU. RWRNINT is cleared by writing it to 1 to it. Writing 0 has no effect. When CAN_MCR.WRNEN is negated, the RWRNINT flag is masked. The CPU must clear the RWRNINT flag before negating CAN_MCR.WRNEN. Otherwise, RWRNINT will be set when CAN_MCR.WRNEN is set again. RWRNINT is not updated during Freeze mode. When the FlexCAN returns to Normal mode from Pretended Networking mode, RWRNINT is updated to reflect the Rx error counter state. See "Receive Process under Pretended Networking Mode" 0b - No such occurrence. 1b - The Rx error counter transitioned from less than 96 to greater than or equal to 96.
15 BIT1ERR	**Bit 1 Error –Indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or Data Phase of a CAN FD message.** NOTE: BIT1ERR is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits. BIT1ERR is updated when the FlexCAN returns to Normal mode from Pretended Networking mode.
14 BIT0ERR	**Bit 0 Error–Indicates when an inconsistency occurs between the transmitted and the received bit in a non-CAN FD message or in the arbitration or Data Phase of a CAN FD message.** BIT0ERR is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - At least one bit sent as dominant is received as recessive.
13 ACKERR	**Acknowledge Error –Indicates that an Acknowledge error was detected by the transmitter node (that is, a dominant bit was detected during the ACK SLOT).** ACKERR is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - An ACK error occurred since last read of this register.
12	**Cyclic Redundancy Check Error –Indicates that a CRC error was detected by the receiver node either in a non-FD message or in the arbitration or Data Phase of a frame in CAN FD format (that is, the calculated CRC is different from the received CRC).**

Yuntu Microelectronics

Field	Function
CRCERR	CRCERR is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - A CRC error occurred since last read of this register.
11 FRMERR	**Form Error –Indicates that a form error was detected by the receiver node in a non-FD message or in a CAN FD message arbitration or Data Phase (that is, a fixed-form bit field contains at least one illegal bit).** FRMERR is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - A form error occurred since last read of this register.
10 STFERR	**Stuffing Error –Indicates that a stuffing error was detected by the receiver node in a non-FD message or in a CAN FD message arbitration or Data Phase.** STFERR is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - A stuffing error occurred since last read of this register.
9 TXWRN	**TX Error Warning –Indicates when repetitive errors are occurring during message transmission.** TXWRN is affected by the value of CAN_ECR.TXERRCNT only. TXWRN is not updated during Freeze mode. TXWRN is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - TXERRCNT is greater than or equal to 96.
8 RXWRN	**RX Error Warning** RXWRN is affected by the value of CAN_ECR.RXERRCNT only. RXWRN is not updated during Freeze mode. RXWRN is updated when the FlexCAN returns to Normal mode from Pretended Networking mode. 0b - No such occurrence. 1b - RXERRCNT is greater than or equal to 96.
7 IDLE	**IDLE** Indicates when the CAN bus is in the IDLE state. See CAN state related bit encodings 0b - No such occurrence. 1b - CAN bus is now IDLE.
6	**FlexCAN in Transmission**

Yuntu Microelectronics

Field	Function
TX	Indicates if the FlexCAN is transmitting a message. See CAN state related bit encodings 0b - FlexCAN is not transmitting a message. 1b - FlexCAN is transmitting a message.
5 - 4 FLTCONF	**FlexCAN Fault Confinement State –Indicates the fault confinement state of the FlexCAN module.** If CAN_CTRL1.LOM is asserted, after a delay that depends on the CAN bit timing, the FLTCONF field will indicate Error Passive. The same delay affects how FLTCONF reflects an update to the CAN_ECR register by the CPU. It may take up to one CAN bit time to get them coherent again. FLTCONF is affected by soft reset. However, if CAN_CTRL1.LOM is asserted, the FLTCONF reset value lasts just one CAN bit. After this time, FLTCONF indicates Error Passive. 00b - Error Active 01b - Error Passive 10b - Bus Off 11b - Bus Off
3 RX	**FlexCAN in Reception** Indicates if the FlexCAN is receiving a message. See CAN state related bit encodings 0b - FlexCAN is not receiving a message. 1b - FlexCAN is receiving a message.
2 BOFFINT	**Bus Off Interrupt** BOFFINT is set when the FlexCAN enters the Bus Off state. If the corresponding mask bit (CAN_CTRL1.BOFFMSK) is set, an interrupt is generated to the CPU. BOFFINT is cleared by writing 1 to it. Writing 0 has no effect. 0b - No such occurrence. 1b - FlexCAN module entered Bus Off state.
1 ERRINT	**Error Interrupt** ERRINT indicates that at least one of the error bits (BIT1ERR, BIT0ERR, ACKERR, CRCERR, FRMERR, or STFERR) is set. If the corresponding mask bit (CAN_CTRL1.ERRMSK) is set, an interrupt is generated to the CPU. ERRINT is cleared by writing 1 to it. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates setting of any error bit in CAN_ESR1.
0	**Wake Up Interrupt –Applies when the FlexCAN is in low-power mode under Self Wake Up mechanism, either Doze mode or Deepsleep mode.**

Yuntu Microelectronics

Field	Function
WAKINT	When a recessive-to-dominant transition is detected on the CAN bus and the CAN_MCR.WAKMSK bit is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When CAN_MCR.SLFWAK is negated, the WAKINT flag is masked. The CPU must clear the WAKINT flag before negating the CAN_MCR.SLFWAK bit. Otherwise, WAKINT will be set when CAN_MCR.SLFWAK is set again. Writing 0 has no effect. WAKINT is cleared by writing 1 to it. Writing 0 has no effect. 0b - No such occurrence. 1b - Indicates a recessive-to-dominant transition was received on the CAN bus.

20.3.3.9 CAN IMASK2 Register

The CAN_IMASK2 register allows any number of the 32 message buffer interrupts to be enabled or disabled for MB63 to MB32. CAN_IMASK2 contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception (as indicated by the corresponding CAN_IFLAG2 bit).
The CAN_IMASK2 register is not available if parameter NUMBER_OF_MB is less than 64.

Offset: 24h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF63TO32M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.15: CAN IMASK2 Register Description

Field	Function
31 - 0	**Buffer MB i Mask**

Yuntu Microelectronics

Field	Function
BUF63TO32M	Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB63 to MB32. NOTE: Setting or clearing a bit in CAN_IMASK2 can assert or negate an interrupt request if the corresponding CAN_IFLAG2 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

20.3.3.10 CAN IMASK1 Register

The CAN_IMASK1 register allows any number of the 32 message buffer interrupts to be enabled or disabled for MB31 to MB. CAN_IMASK1 contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception (as indicated by the corresponding CAN_IFLAG1 bit).

Offset: 28h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUF31TO0M															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.16: CAN IMASK1 Register Description

Field	Function
31 - 0 BUF31TO0M	**Buffer MB i Mask** Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB31 to MB0. NOTE: Setting or clearing a bit in CAN_IMASK1 can assert or negate an interrupt request if the corresponding CAN_IFLAG1 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

Yuntu Microelectronics

20.3.3.11 CAN IFLAG2 Register

The CAN_IFLAG2 register defines the flags for the 32 message buffer interrupts for MB63 to MB32. CAN_IFLAG2 contains one interrupt flag bit per buffer. Each successful transmission or reception sets the respective CAN_IFLAG2 bit. If the corresponding CAN_IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.
Before updating the CAN_MCR.MAXMB field, the CPU must service the CAN_IFLAG2 bits whose MB value is greater than the CAN_MCR.MAXMB to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.
The CAN_IFLAG2 register is not available if parameter NUMBER_OF_MB is less than 64.

Offset: 2ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF63TO32I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.17: CAN IFLAG2 Register Description

Field	Function
31 - 0 BUF63TO32I	**Buffer MB i Interrupt** Each bit flags the corresponding FlexCAN Message Buffer Interrupt for MB63 to MB32. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

20.3.3.12 CAN IFLAG1 Register

The CAN_IFLAG1 register defines the flags for the 32 message buffer interrupts for MB31 to MB0. CAN_IFLAG1 contains one interrupt flag bit per buffer. Each successful transmission or reception sets

Yuntu Microelectronics

the corresponding CAN_IFLAG1 bit. If the corresponding CAN_IMASK1 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

Before updating the CAN_MCR.MAXMB field, the CPU must service the CAN_IFLAG1 bits whose MB value is greater than the CAN_MCR.MAXMB to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

There is an exception when DMA for Legacy Rx FIFO is enabled, as described below.

Legacy Rx FIFO

The BUF7I to BUF5I flags are also used to represent Legacy FIFO interrupts when the Legacy Rx FIFO is enabled. When CAN_MCR.RFEN is set and CAN_MCR.DMA is negated, the function of the 8 least significant interrupt flags changes:

- BUF7I, BUF6I, and BUF5I indicate operating conditions of the Legacy FIFO.
- BUF0I is used to empty the Legacy FIFO.
- BUF4I to BUF1I are reserved.

Before setting CAN_MCR.RFEN, the CPU must service the IFLAG bits asserted in the Legacy Rx FIFO region (see “Legacy Rx FIFO”). Otherwise, these IFLAG bits will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When CAN_MCR.RFEN is negated, the Legacy FIFO flags must be cleared. The same care must be taken when a CAN_CTRL2.RFFN value is selected extending Legacy Rx FIFO filters beyond MB7. For example, when RFFN is 0x8, the MB0–23 range is occupied by Legacy Rx FIFO filters and related IFLAG bits must be cleared.

The Legacy Rx FIFO must be disabled when CAN_MCR.FDEN is set.

Legacy Rx FIFO with DMA

When both CAN_MCR.RFEN and CAN_MCR.DMA are asserted (DMA feature for Legacy Rx FIFO enabled), the function of the 8 least significant interrupt flags (BUF7I–BUF0I) are changed to support the DMA operation:

- BUF7I, BUF6I, and BUF4I–BUF1I are not used.
- BUF5I indicates the operating condition of the Legacy Rx FIFO.
- BUF5I does not generate a CPU interrupt, but generates a DMA request.
- BUF0I is used to empty the Legacy FIFO.

IMASK1 bits in the Legacy Rx FIFO region are not considered when CAN_MCR.DMA is asserted. In addition, the CPU must not clear the flag BUF5I when DMA is enabled. Before setting CAN_MCR.DMA, the CPU must service the IFLAGs asserted in the Rx FIFO region.

When CAN_MCR.DMA is negated, the Legacy Rx FIFO must be empty.

Offset: 30h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF31TO8I															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF31TO8I								BUF7I	BUF6I	BUF5I	BUF4TO1I				BUF0I
W	W1C								W1C	W1C	W1C	W1C				W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.18: CAN IFLAG1 Register Description

Field	Function
31 - 8 BUF31TO8I	**Buffer MBi Interrupt** Each bit flags the corresponding FlexCAN Message Buffer Interrupt for MB31 to MB8. Only the lower 8 bits are available if parameter NUMBER_OF_MB is 16. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.
7 BUF7I	**Buffer MB7 Interrupt or Legacy Rx FIFO Overflow** When CAN_MCR.RFEN is cleared (Legacy Rx FIFO disabled), BUF71 flags the interrupt for MB7. When CAN_MCR.RFEN is set (Legacy Rx FIFO enabled), BUF7I represents Legacy Rx FIFO Overflow. In this case, BUF7I indicates that a message was lost because the Legacy Rx FIFO is full. Note that BUF7I is not asserted when the Legacy Rx FIFO is full and the message was captured by a mailbox. NOTE: BUF7I is cleared by the FlexCAN whenever CAN_MCR.RFEN is changed by a CPU write. 0b - No occurrence of MB7 completing transmission/reception when CAN_MCR.RFEN=0, or of Legacy Rx FIFO overflow when CAN_MCR.RFEN=1. 1b - MB7 completed transmission/reception when CAN_MCR.RFEN=0, or Legacy Rx FIFO overflow when CAN_MCR.RFEN=1.
6	**Buffer MB6 Interrupt or Legacy Rx FIFO Warning**

Yuntu Microelectronics

Field	Function
BUF6I	When CAN_MCR.RFEN is cleared (Legacy Rx FIFO disabled), BUF6I flags the interrupt for MB6. When CAN_MCR.RFEN is set (Legacy Rx FIFO enabled), BUF6I represents Legacy Rx FIFO Warning. In this case, BUF6I indicates when the number of unread messages in the Legacy Rx FIFO increases from 4 to 5 due to the reception of a new message, meaning that the Legacy Rx FIFO is almost full. Note that if BUF6I is cleared when the number of unread messages is greater than 4, BUF6I is not asserted again until the number of unread messages in the Legacy Rx FIFO decreases to 4 or less. NOTE: BUF6I is cleared by the FlexCAN whenever CAN_MCR.RFEN is changed by a CPU write. 0b - No occurrence of MB6 completing transmission/reception when CAN_MCR.RFEN=0, or of Legacy Rx FIFO almost full when CAN_MCR.RFEN=1. 1b - MB6 completed transmission/reception when CAN_MCR.RFEN=0, or Legacy Rx FIFO almost full when CAN_MCR.RFEN=1.
5 BUF5I	**Buffer MB5 Interrupt or Frames Available in Legacy Rx FIFO** When CAN_MCR.RFEN is cleared (Legacy Rx FIFO disabled), BUF5I flags the interrupt for MB5. When CAN_MCR.RFEN is set (Legacy Rx FIFO enabled), BUF5I represents Frames Available in Legacy Rx FIFO and indicates that at least one frame is available to be read from the Legacy Rx FIFO. When CAN_MCR.DMA is set, BUF5I generates a DMA request and the CPU must not clear BUF5I by writing 1 to it. NOTE: BUF5I is cleared by the FlexCAN whenever CAN_MCR.RFEN is changed by a CPU write. 0b - No occurrence of MB5 completing transmission/reception when CAN_MCR.RFEN=0, or of frame(s) available in the FIFO, when CAN_MCR.RFEN=1. 1b - MB5 completed transmission/reception when CAN_MCR.RFEN=0, or frame(s) available in the Legacy Rx FIFO when CAN_MCR.RFEN=1. BUF5I generates a DMA request when both CAN_MCR.RFEN and CAN_MCR.DMA are set.
4 - 1 BUF4TO1I	**Buffer MB Interrupt or Reserved** When CAN_MCR.RFEN is cleared (Legacy Rx FIFO disabled), BUF4TO1I flag the interrupts for MB4 to MB1. When CAN_MCR.RFEN is set (Legacy Rx FIFO enabled), BUF4TO1I flags are reserved. NOTE: BUF4TO1I are cleared by the FlexCAN whenever CAN_MCR.RFEN is changed by a CPU write. 0000b - The corresponding buffer has no occurrence of successfully completed transmission or reception when CAN_MCR.RFEN=0. 0001b - The corresponding buffer has successfully completed transmission or reception when CAN_MCR.RFEN=0.
0	**Buffer MB0 Interrupt or Clear FIFO Bit**

Yuntu Microelectronics

Field	Function
BUF0I	When CAN_MCR.RFEN is set (Legacy Rx FIFO enabled), BUF0I is used to trigger the clear FIFO operation. This operation empties Legacy Rx FIFO contents. Before performing this operation, the CPU must service all FIFO related IFLAGs. When CAN_MCR.DMA is set (DMA is enabled), this operation also clears the BUF5I flag and consequently aborts the DMA request. The clear FIFO operation occurs when the CPU writes 1 to BUF0I, which is only allowed in Freeze mode and is blocked by hardware in other conditions. When CAN_MCR.RFEN is cleared (Legacy Rx FIFO disabled), BUF0I flags the interrupt for MB0: 0b - MB0 has no occurrence of successfully completed transmission or reception when CAN_MCR.RFEN=0. 1b - MB0 has successfully completed transmission or reception when CAN_MCR.RFEN=0.

20.3.3.13 CAN CTRL2 Register

| The CAN_CTRL2 register complements CAN_CTRL1, providing control bits for memory write access in Freeze mode, for extending FIFO filter quantity, and for adjusting the operation of internal FlexCAN processes like matching and arbitration.

The contents of CAN_CTRL2 are not affected by soft reset.

Table 20.19: CAN TASD Reset Value

Number of MBs	TASD reset value(FD_EN=0)	TASD reset value(FD_EN=1)
16	22	Invalid configuration
32	20	20
64	16	16
96	12	12
128	8	2

Table 20.20: CAN Number of Legacy of RX FIFO Filters

RFFN[3:0]	Number of Rx FIFO Filter Elements	Message Buffers Occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks	RX FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask
0x0	8	MB 0-7	MB 8-N	Elements 0-7	None
0x1	16	MB 0-9	MB 10-N	Elements 0-9	Elements 10-15
0x2	24	MB 0-11	MB 12-N	Elements 0-11	Elements 12-23
0x3	32	MB 0-13	MB 14-N	Elements 0-13	Elements 14-31
0x4	40	MB 0-15	MB 16-N	Elements 0-15	Elements 16-39
0x5	48	MB 0-17	MB 18-N	Elements 0-17	Elements 18-47
0x6	56	MB 0-19	MB 20-N	Elements 0-19	Elements 20-55
0x7	64	MB 0-21	MB 22-N	Elements 0-21	Elements 22-63
0x8	72	MB 0-23	MB 24-N	Elements 0-23	Elements 24-71
0x9	80	MB 0-25	MB 26-N	Elements 0-25	Elements 26-79
0xA	88	MB 0-27	MB 28-N	Elements 0-27	Elements 28-87
0xB	96	MB 0-29	MB 30-N	Elements 0-29	Elements 30-95
0xC	104	MB 0-31	MB 32-N	Elements 0-31	Elements 32-103

Yuntu Microelectronics

Table 20.20 continued from previous page

RFFN[3:0]	Number of Rx FIFO Filter Elements	Message Buffers Occupied by Rx FIFO and ID Filter Table	Remaining Available Mailboxes	Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks	RX FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask
0xD	112	MB 0-33	MB 34-N	Elements 0-31	Elements 32-111
0xE	120	MB 0-35	MB 36-N	Elements 0-31	Elements 32-119
0xF	128	MB 0-37	MB 38-N	Elements 0-31	Elements 32-127

Offset: 34h

Table 20.21: CAN CTRL2 Register Description

Field	Function
31 ERRMSK_FAST	**Error Interrupt Mask for errors detected in the Data Phase of fast CAN FD frames** ERRMSK_FAST provides a mask for the ERRINT_FAST Interrupt in CAN_ESR1. ERRMASK_FAST is not available (reserved, write 0) if parameter FD_EN is 0. 0b - ERRINT_FAST error interrupt disabled. 1b - ERRINT_FAST error interrupt enabled.
30 BOFFDONEMSK	**Bus Off Done Interrupt Mask** Provides a mask for the Bus Off done interrupt in CAN_ESR1. 0b - Bus Off done interrupt disabled 1b - Bus Off done interrupt enabled.
29 ECRWRE	**Error-Correction Configuration Register Write Enable –Enables the CAN_MECR register to be updated.** ECRWRE is automatically set to 0 if the protocol described in section "Detection and Correction of Memory Errors" is not followed. ECRWRE is not available (reserved, write 0) if parameter ECC_EN is 0. 0b - Disable update. 1b - Enable update.

Yuntu Microelectronics

Field	Function
28 WRMFRZ	**Write-Access To Memory In Freeze Mode –Enable unrestricted write access to FlexCAN memory in Freeze mode.** WRMFRZ can only be written in Freeze mode and has no effect out of Freeze mode. CAN_MCR.RFEN bit must not be set during FlexCAN memory initialization. WRMFRZ is not available (reserved, write 0) if parameter ECC_EN is 0. 0b - Maintain the write access restrictions. 1b - Enable unrestricted write access to FlexCAN memory.
27 - 24 RFFN	Number of Legacy Rx FIFO Filters –Defines the number of Legacy Rx FIFO filters. The maximum selectable number of filters is a function of configuration parameter NUMBER_OF_MB. RFFN must not be programmed with values that cause the number of message buffers occupied by Legacy Rx FIFO and ID Filter to exceed the number of mailboxes present, defined by CAN_MCR.MAXMB. Each group of eight filters occupies a memory space equivalent to two message buffers, which means that as more filters are implemented fewer mailboxes are available. Considering that the Legacy Rx FIFO occupies the memory space originally reserved for MB0–5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words, which can be calculated as follows: (SETUP_MB - 6) × 4 where SETUP_MB is the lower value between parameter NUMBER_OF_MB and register field CAN_MCR.MAXMB. The number of remaining mailboxes available will be: (SETUP_MB - 8) - (RFFN × 2) If the number of Legacy Rx FIFO filters programmed through RFFN exceeds the SETUP_MB value (memory space available), the exceeding ones will not be functional. NOTE: The number of the last remaining available mailbox is defined by the least value between (NUMBER_OF_MB - 1) and the CAN_MCR.MAXMB field. If Rx Individual Mask registers are not enabled, all Legacy Rx FIFO filters are affected by the Legacy Rx FIFO Global Mask. RFFN can only be written in Freeze mode as it is blocked by hardware in other modes.
23 - 19 TASD	**Tx Arbitration Start Delay –Indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus.** TASD can be written only in Freeze mode because it is blocked by hardware in other modes. See “Tx Arbitration Start Delay” for more details.
18	**Mailboxes Reception Priority**

Yuntu Microelectronics

Field	Function
MRP	If MRP is set, the matching process starts from the mailboxes and if no match occurs the matching continues on the Legacy Rx FIFO. MRP can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Matching starts from Legacy Rx FIFO and continues on mailboxes. 1b - Matching starts from mailboxes and continues on Legacy Rx FIFO.
17 RRS	**Remote Request Storing** If RRS is asserted, a Remote Request Frame is submitted to a matching process and stored in the corresponding message buffer in the same fashion as a Data Frame. No automatic Remote Response Frame will be generated. If RRS is negated, the Remote Request Frame is submitted to a matching process and an automatic Remote Response Frame is generated if a message buffer with CODE=0b1010 is found with the same ID. RRS can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Remote Response Frame is generated. 1b - Remote Request Frame is stored.
16 EACEN	**Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes** EACEN controls the comparison of IDE and RTR bits within Rx mailboxes filters with their corresponding bits in the incoming frame by the matching process. EACEN does not affect matching for Legacy Rx FIFO. EACEN can be written only in Freeze mode because it is blocked by hardware in other modes. 0b - Rx mailbox filter's IDE bit is always compared and RTR is never compared, regardless of mask bits. 1b - Enables the comparison of both the IDE and RTR bits of an Rx mailbox filter with the corresponding bits of the incoming frame. Mask bits do apply.
15 TIMER_SRC	**Timer Source –Selects the time tick source used for incrementing the free-running timer counter.** TIMER_SRC can be written in Freeze mode only. 0b - The free-running timer is clocked by the CAN bit clock, which defines the baud rate on the CAN bus. 1b - The free-running timer is clocked by an external time tick. The period can be either adjusted to be equal to the baud rate on the CAN bus or a different value as required.
14 PREXCEN	**Protocol Exception Enable –Enables the protocol exception feature.** PREXCEN is writable only in Freeze mode. PREXCEN is not available (reserved, write 0) if parameter FE_EN is 0. 0b - Protocol exception is disabled. 1b - Protocol exception is enabled.
13	**Bit Timing Expansion enable**

Yuntu Microelectronics

Field	Function
BTE	This field enables the use of the EPRS, EDCBT and ENCBT registers to configure the CAN bit timing segments, instead of using the bit timing fields of CBT, FDCBT, and CTRL1 registers. If CTRL2[BTE] is set: • PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields in CTRL1 register are read as zero, and a write operation in them has no effect. • EPRESDIV, ERJW, EPROPSEG, EPSEG1, and EPSEG2 fields in the CBT register, and the FDCBT register, are read as zero and a write operation to them has no effect. • The ETDCOFF, ETDCEN, ETDCFAIL, and ETDCVAL fields of the ETDC register are used by FlexCAN instead of TDCOFF, TDCEN, TDCFAIL, and TDCVAL fields in FDCTRL register, which are read as zero, and a write operation in them has no effect. • ETDC[TDMDIS] can be used for disabling transceiver delay measurement. 0b - CAN Bit timing expansion is disabled. 1b - CAN bit timing expansion is enabled.
12 ISOCANFD EN	**ISO CAN FD Enable** ISOCANFDEN enables ISO CAN FD compliant operation by enabling the following features, which are part of the ISO 11898 standard and not included in the original (Bosch) CAN FD protocol specification: • The count of variable stuff bits inserted from the Start of Frame bit to the last bit of Data field. Also, the modulo 8 count of variable stuff bits plus the respective parity bit (even parity calculated over the 3-bit modulo 8 count) are combined as the 4-bit Stuff Count field and inserted before the CRC Sequence field. • CRC calculation extends beyond the end of Data field and takes the Stuff Count field bits into account. ISOCANFDEN is writable only in Freeze mode. ISOCANFDEN is not available (reserved, write 0) if parameter FD_EN is 0. 0b - Disable ISO CAN FD specific features (non-ISO CAN FD operation). 1b - Enable ISO CAN FD compliant operation.
11 EDFLTDIS	**Edge Filter Disable –Disables the edge filter used during the bus integration state.** When the edge filter is enabled, two consecutive nominal time quanta with dominant bus state are required to detect an edge that causes synchronization. When synchronization occurs, the counting of the sequence of eleven consecutive recessive bits is restarted. The edge filter prevents the dominant pulses that are shorter than a nominal bit time (present during the data phase of an FD Frame) from being mistaken for an idle condition. EDFLTDIS is writable only in Freeze mode. EDFLTDIS is not available (reserved, write 0) if parameter FD_EN is 0. 0b - Edge filter is enabled. 1b - Edge filter is disabled.
9 - 8	**Message Buffer Time Stamp Base**

Yuntu Microelectronics

Field	Function
MBTSBASE	MBTSBASE selects which time base is used for capturing the 16-bit TIME STAMP field of the message buffer. MBTSBASE is writable only in Freeze mode. MBTSBASE is not available (reserved, write 0) if parameter TIME_STAMP_EN is 0. 00b - Message buffer time stamp base is CAN_TIMER. 01b - Message buffer time stamp base is lower 16 bits of high-resolution timer. 10b - Message buffer time stamp base is upper 16 bits of high-resolution timer. 11b - Reserved.
7 - 6 TSTAMPCAP	**Time Stamp Capture Point** TSTAMPCAP configures the point in time at which a 32-bit time base is captured during a CAN frame and stored in the high-resolution time stamp register (CAN_HR_TIME_STAMPn). For Classical CAN frames, capture points can be the start of frame (SOF) bit or the point in time when a CAN frame is considered valid, which is the 7th bit of end of frame for transmission and the 6th bit of the end of frame for reception. For CAN FD frames, the high-resolution time stamp can be captured at SOF, the point in time when a CAN FD frame is considered valid, or the res bit. TSTAMPCAP is writable only in Freeze mode. TSTAMPCAP is not available (reserved, write 0) if parameter TIME_STAMP_EN is 0. 00b - The high-resolution time stamp capture is disabled. 01b - The high-resolution time stamp is captured at the end of the CAN frame. 10b - The high-resolution time stamp is captured at the start of the CAN frame. 11b - The high-resolution time stamp is captured at the start of frame for classical CAN frames and at the res bit for CAN FD frames.

Yuntu Microelectronics

20.3.3.14 CAN ESR2 Register

The CAN_ESR2 register reports some general status information

Offset: 38h

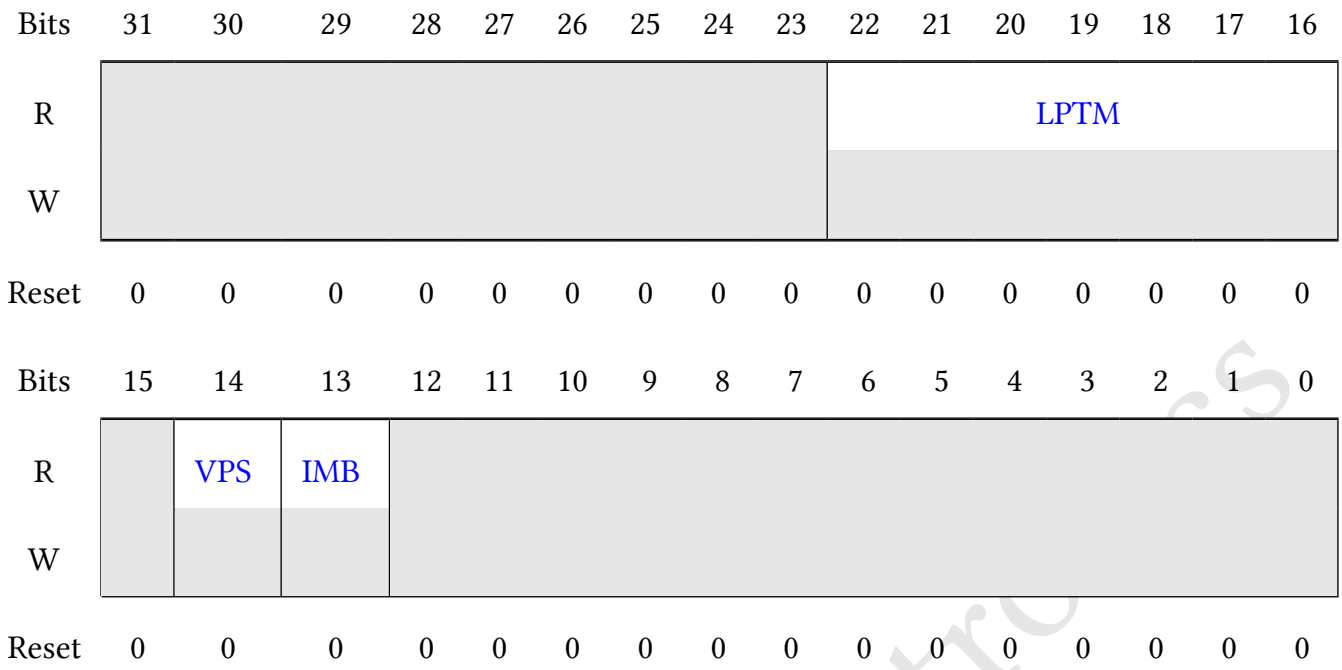


Table 20.22: CAN ESR2 Register Description

Field	Function
22 - 16 LPTM	**Lowest Priority Tx Mailbox** If CAN_ESR2.VPS is asserted, LPTM indicates the lowest number inactive mailbox (see the CAN_ESR2.IMB bit description). If there is no inactive mailbox, then the mailbox indicated depends on the CAN_CTRL1.LBUF bit value: • If CAN_CTRL1.LBUF is negated, the mailbox indicated is the one that has the greatest arbitration value (see "Highest-Priority Mailbox First"). • If CAN_CTRL1.LBUF is asserted, the mailbox indicated is the highest number active Tx mailbox. If a Tx mailbox is being transmitted, it is not considered in LPTM calculation. If CAN_ESR2.IMB is not asserted and a frame is transmitted successfully, LPTM is updated with its mailbox number.
14	**Valid Priority Status –Indicates whether CAN_ESR2.IMB and CAN_ESR2.LPTM contents are currently valid or not.**

Yuntu Microelectronics

Field	Function
VPS	VPS is asserted upon every complete Tx arbitration process unless the CPU writes to the C/S word of a mailbox that has already been scanned (is behind Tx Arbitration Pointer during the Tx arbitration process). If there is no inactive mailbox and only one Tx mailbox that is being transmitted, then VPS is not asserted. VPS is negated upon the start of every Tx arbitration process or upon a write to the C/S word of any mailbox. NOTE: VPS is not affected by any CPU write into the C/S of an MB that is blocked by the abort mechanism. When CAN_MCR.AEN is asserted, the abort code write into the C/S of an MB that is being transmitted (pending abort), or any write attempt into a Tx MB with CAN_IFLAG set is blocked. 0b - Contents of IMB and LPTM are invalid. 1b - Contents of IMB and LPTM are valid.
13 IMB	**Inactive Mailbox** If CAN_ESR2.VPS is asserted, IMB indicates whether there is any inactive mailbox (CODE field is either 0b1000 or 0b0000). IMB is asserted in the following cases: • During arbitration, if a CAN_ESR2.LPTM is found and it is inactive. • If CAN_ESR2.IMB is not asserted and a frame is transmitted successfully. IMB is always cleared at start of arbitration (See "Arbitration Process"). NOTE: If an IMB is successfully transmitted and CAN_ESR2.IMB=0 (no inactive mailbox), then CAN_ESR2.VPS and CAN_ESR2.IMB are asserted and the index related to the MB just transmitted is loaded into CAN_ESR2.LPTM. 0b - If CAN_ESR2.VPS is asserted, CAN_ESR2.LPTM is not an inactive mailbox. 1b - If CAN_ESR2.VPS is asserted, there is at least one inactive mailbox. CAN_ESR2.LPTM content is the number of the first one.

20.3.3.15 CAN CRCR Register

The CAN_CRCR register provides information about the CRC of transmitted messages for non-FD messages. CAN_CRCR only reports the 15 low order bits of CRC calculations for messages in CAN FD format that require either 17 or 21 bits. For CAN FD format frames, the CAN_FDCRC register must be used. CAN_CRCR is updated at the same time the Tx interrupt flag is asserted. The CAN_FDCRC register is not available when parameter FD_EN is 0.

Offset: 44h

Yuntu Microelectronics

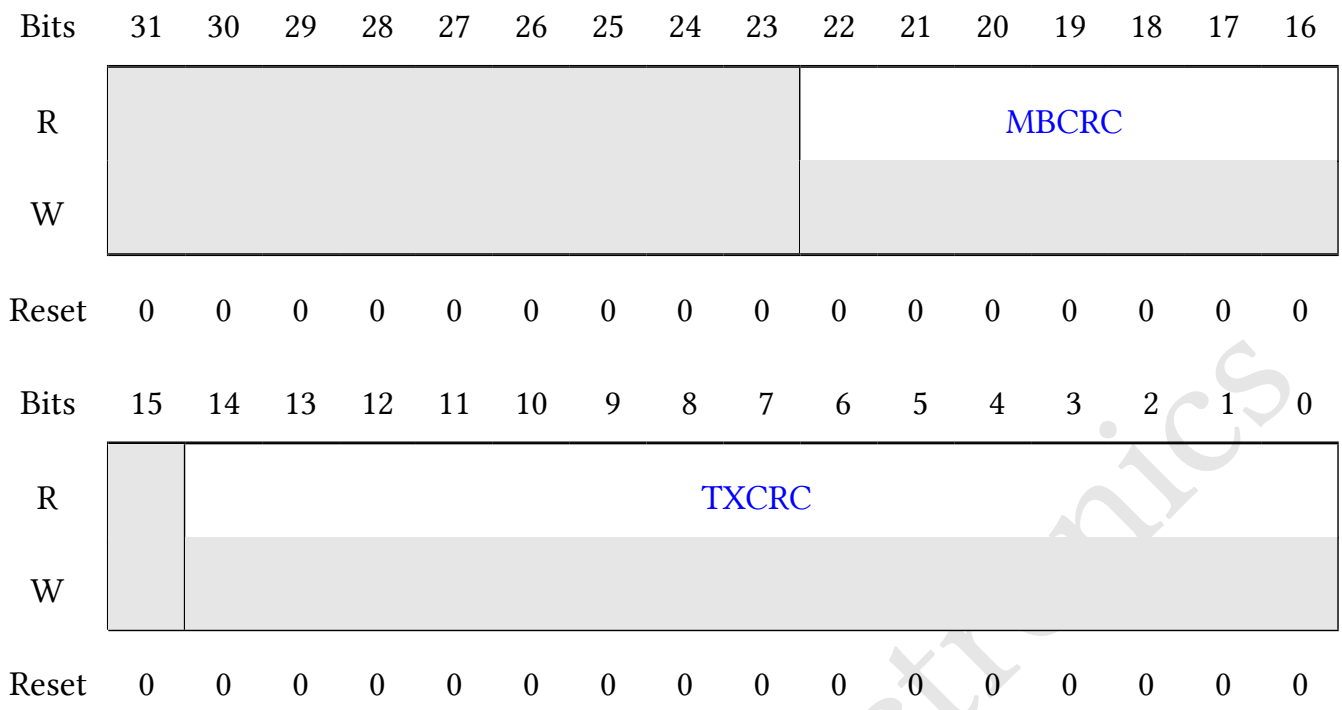


Table 20.23: CAN CRCR Register Description

Field	Function
22 - 16 MBCRC	**CRC Mailbox** Indicates the number of the mailbox corresponding to the value in the CAN_CRCR.TXCRC field.
14 - 0 TXCRC	**Transmitted CRC Value** Indicates the CRC value of the last transmitted message for non-FD frames. For CAN FD frames, the CRC value is reported in CAN_FDCRC.

20.3.3.16 CAN RXFGMASK Register

| The CAN_RXFGMASK register is located in RAM.

If the Rx FIFO is enabled, CAN_RXFGMASK is used to mask the Rx FIFO ID Filter Table elements that do not have a corresponding CAN_RXIMR according to CAN_CTRL2.RFFN field setting.

CAN_RXFGMASK can only be written in Freeze mode as it is blocked by hardware in other modes.

Table 20.24: CAN Correspondence of Legacy RX FIFO Global Mask Bits to IDF Fields

Legacy Rx FIFO ID Filter Table Elements Format (CAN_MCR.IDAM)	Identifier Acceptance Filter Fields					
	RTR	IDE	RXIDA	RXIDB[1]	RXIDC[2]	Reserved
A	FGM[31]	FGM[30]	FGM[29:1]	-		FGM[0]
B	FGM[31], FGM[15]	FGM[30], FGM[14]		FGM[29:16], FGM[13:0]	-	

Table 20.24 continued from previous page

Legacy Rx FIFO ID Filter Table Elements Format (CAN_MCR.IDAM)	Identifier Acceptance Filter Fields					
	RTR	IDE	RXIDA	RXIDB[1]	RXIDC[2]	Reserved
C	-	-	-	-	FGM[31:24], FGM[23:16], FGM[15:8], FGM[7:0]	-

1. If CAN_MCR.IDAM is equivalent to format B, only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Legacy Rx FIFO filter.
2. If CAN_MCR.IDAM field is equivalent to format C, only the eight most significant bits of the Identifier of the incoming frame are compared with the Legacy Rx FIFO filter.

Offset: 48h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FGM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FGM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.25: CAN RXFGMASK Register Description

Field	Function
31 - 0 FGM	**Legacy Rx FIFO Global Mask Bits** These bits mask the ID Filter Table elements bits in a perfect alignment. 0b - The corresponding bit in the filter is “don’t care”. 1b - The corresponding bit in the filter is checked.

Yuntu Microelectronics

20.3.3.17 CAN RXFIR Register

The CAN_RXFIR register provides information on the Legacy Rx FIFO.
CAN_RXFIR is the port through which the CPU accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Legacy Rx FIFO and its output is updated whenever the output of the Legacy Rx FIFO is updated with the next message. See "Legacy Rx FIFO" for instructions on reading CAN_RXFIR.

Offset: 4ch

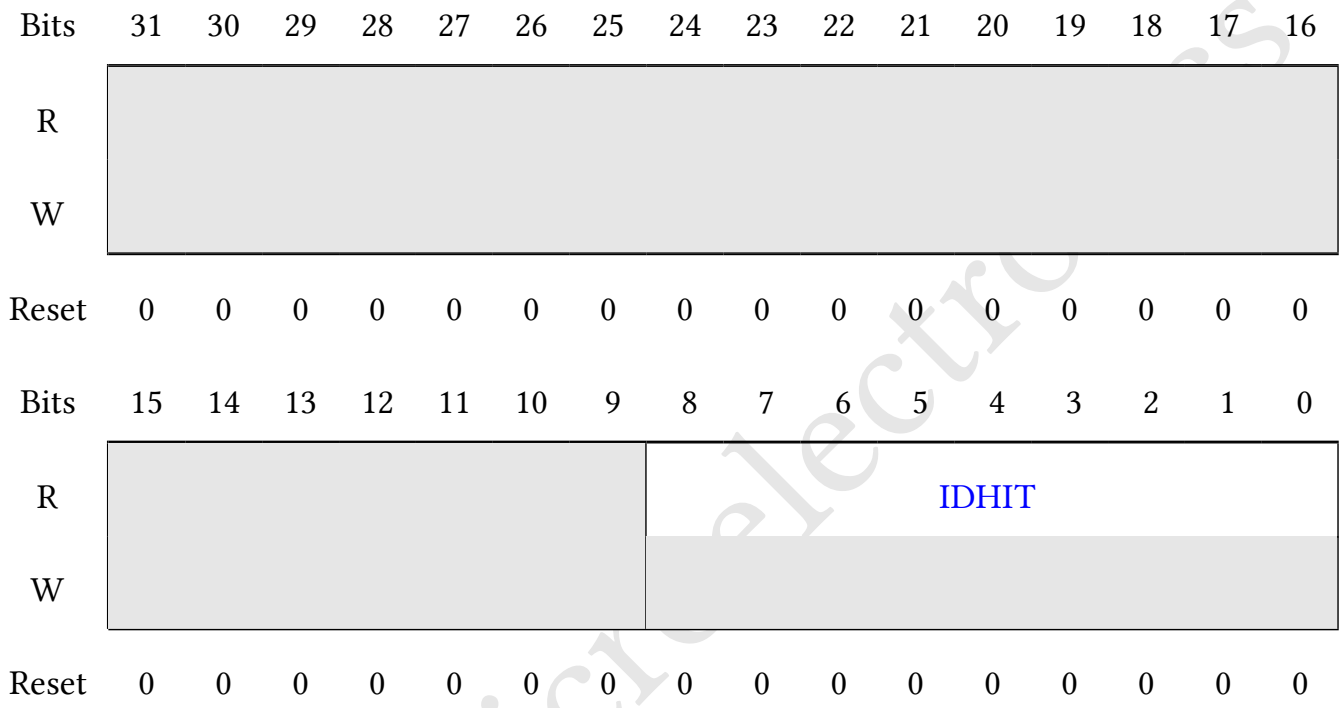


Table 20.26: CAN RXFIR Register Description

Field	Function
8 - 0 IDHIT	**Identifier Acceptance Filter Hit Indicator** Indicates which Identifier Acceptance Filter was hit by the received message that is in the output of the Legacy Rx FIFO. If multiple filters match the incoming message ID, the first matching IDAF found (lowest number) by the matching process is indicated. IDHIT is valid only when CAN_IFLAG1.BUF5I is asserted.

20.3.3.18 CAN CBT Register

The CAN_CBT register is an alternative way to store the CAN bit timing variables described in the CAN_CTRL1 register. EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW are extended versions of CAN_CTRL1 register PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW fields respectively.
The BTF bit selects the use of the timing variables defined in CAN_CBT.
The contents of CAN_CBT are not affected by soft reset.
When CAN_CTRL2.BTE = 1, the EPRESDIV, ERJW, EPROPSEG, EPSEG1, and EPSEG2 fields in CAN_CBT are read as zero and a write operation to them has no effect.

NOTE: The CAN bit variables in CAN_CTRL1 and in CAN_CBT are stored in the same register.
NOTE: When the CAN FD feature is enabled (CAN_MCR.FDEN is set), always set $
$ CAN_CBT.BTF.

Offset: 50h

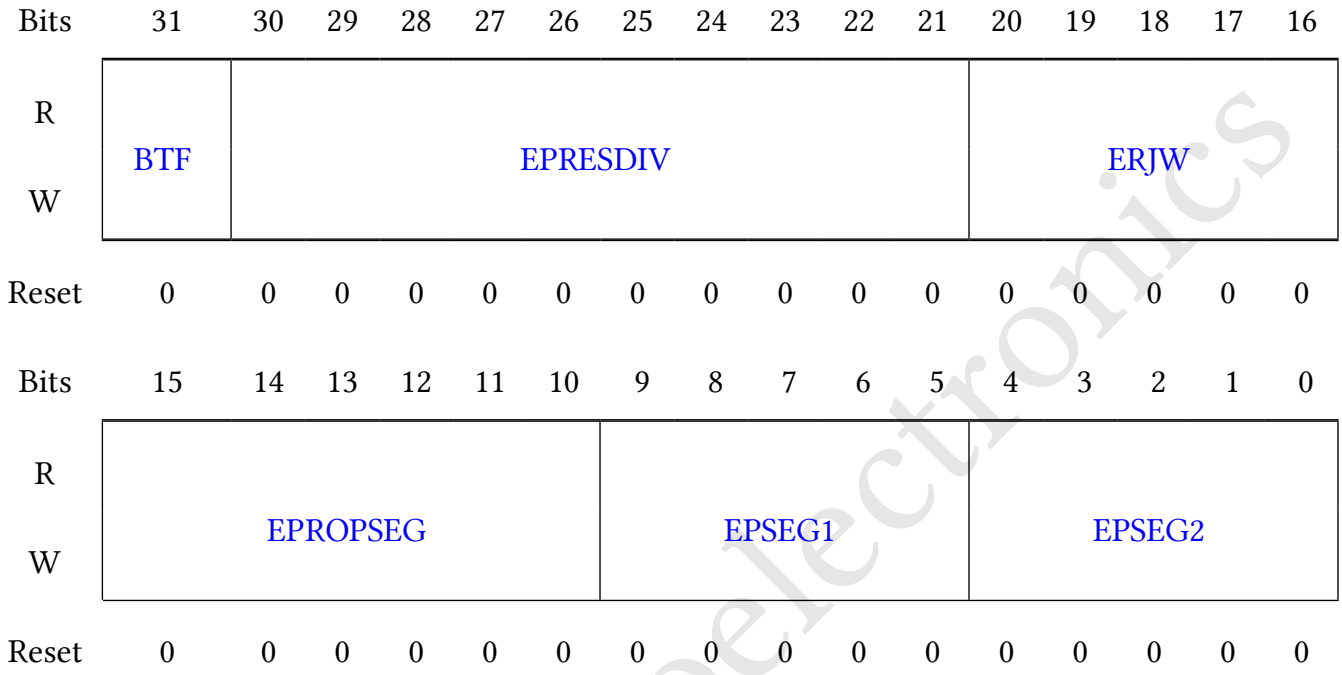


Table 20.27: CAN CBT Register Description

Field	Function
31 BTF	**Bit Timing Format Enable –Enables the use of extended CAN bit timing fields EPRESDIV, EPROPSEG, EPSEG1, EPSEG2, and ERJW, replacing the CAN bit timing variables defined in CAN_CTRL1.** BTF can be written in Freeze mode only. 0b - Extended bit time definitions disabled. 1b - Extended bit time definitions enabled.
30 - 21 EPRESDIV	**Extended Prescaler Division Factor** When CAN_CBT.BTF is asserted, EPRESDIV defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency. EPRESDIV extends the CAN_CTRL1.PRESDIV value range. Sclock frequency = PE clock frequency / (EPRESDIV + 1) The Sclock period defines the time quantum of the CAN protocol. For the reset value, the Sclock frequency is equal to the PE clock frequency (See "Protocol Timing"). When CAN_CBT.BTF is negated, EPRESDIV has no effect. EPRESDIV can be written only in Freeze mode because it is blocked by hardware in other modes.
20 - 16	**Extended Resync Jump Width**

Yuntu Microelectronics

Field	Function
ERJW	When CAN_CBT.BTF is asserted, ERJW defines the maximum number of time quanta that a bit time can be changed by one resynchronization. ERJW extends the CAN_CTRL1.RJW value range. Resync Jump Width = ERJW + 1 One time quantum is equal to the Sclock period. When CAN_CBT.BTF is negated, ERJW has no effect. ERJW can be written only in Freeze mode because it is blocked by hardware in other modes.
15 - 10 EPROPSEG	**Extended Propagation Segment** When CAN_CBT.BTF is asserted, EPROPSEG defines the length of the Propagation Segment in the bit time. EPROPSEG extends the CAN_CTRL1.PROPSEG value range. Propagation Segment Time = (EPROPSEG + 1) × Time-Quanta Time-Quantum = one Sclock period When CAN_CBT.BTF is negated, EPROPSEG has no effect. EPROPSEG can be written only in Freeze mode because it is blocked by hardware in other modes.
9 - 5 EPSEG1	**Extended Phase Segment 1** When CAN_CBT.BTF is asserted, EPSEG1 defines the length of Phase Segment 1 in the bit time. EPSEG1 extends the CAN_CTRL1.PSEG1 value range. Phase Buffer Segment 1 = (EPSEG1 + 1) × Time-Quanta Time-Quantum = one Sclock period When CAN_CBT.BTF is negated, EPSEG1 has no effect. EPSEG1 can be written only in Freeze mode because it is blocked by hardware in other modes.
4 - 0 EPSEG2	**Extended Phase Segment 2** When CAN_CBT.BTF is asserted, EPSEG2 defines the length of Phase Segment 2 in the bit time. EPSEG2 extends the CAN_CTRL1.PSEG2 value range. Phase Buffer Segment 1 = (EPSEG2 + 1) × Time-Quanta Time-Quantum = one Sclock period When CAN_CBT.BTF is negated, EPSEG2 has no effect. EPSEG2 can be written only in Freeze mode because it is blocked by hardware in other modes.

Yuntu Microelectronics

20.3.3.19 CAN IMASK4 Register

The CAN_IMASK4 register allows any number of the 32 message buffer interrupts to be enabled or disabled for MB127 to MB96. CAN_IMASK4 contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception (as indicated by the corresponding CAN_IFLAG4 bit).
The CAN_IMASK4 register is not available if parameter NUMBER_OF_MB is less than 128.

Offset: 68h

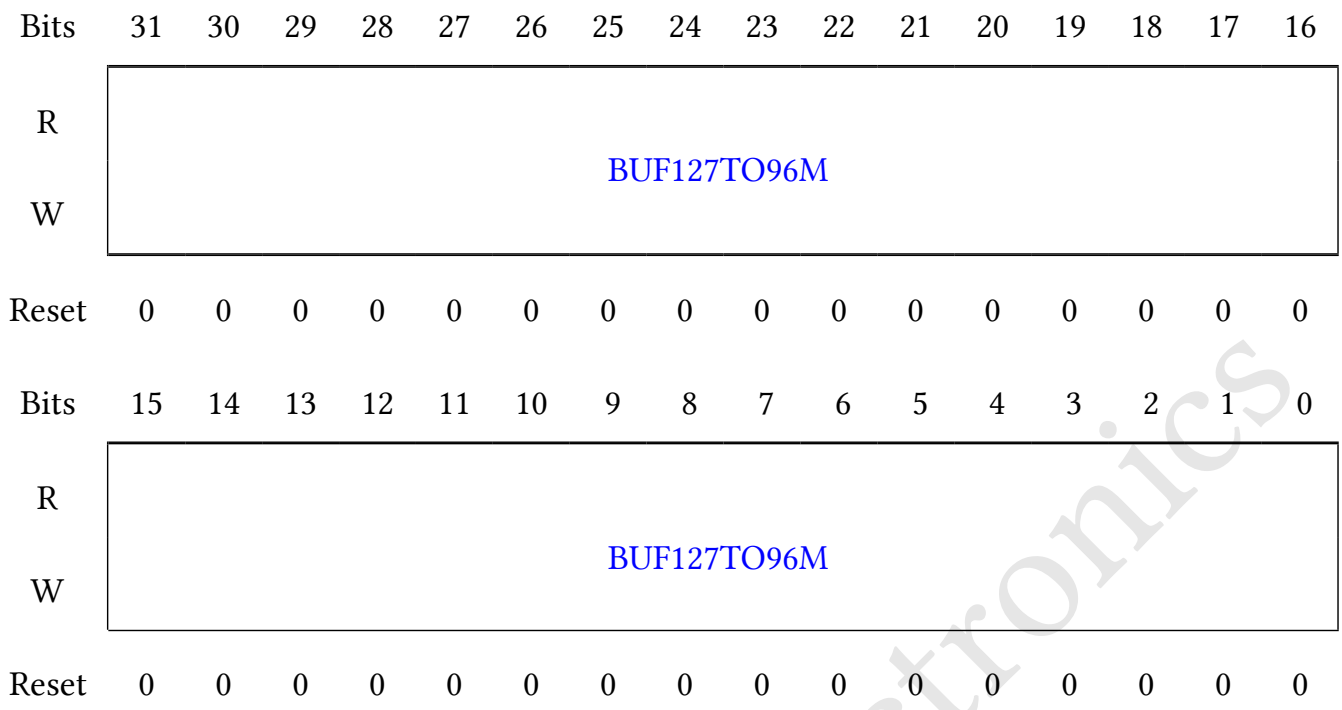


Table 20.28: CAN IMASK4 Register Description

Field	Function
31 - 0 BUF127TO96M	**Buffer MB i Mask** Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB127 to MB96. When CAN FD is enabled, the MB range is defined according to the MBDSRn bit fields of the CAN_FDCTRL register. NOTE: Setting or clearing a bit in CAN_IMASK4 can assert or negate an interrupt request if the corresponding CAN_IFLAG4 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

Yuntu Microelectronics

20.3.3.20 CAN IMASK3 Register

The CAN_IMASK3 register allows any number of the 32 message buffer interrupts to be enabled or disabled for MB95 to MB64. CAN_IMASK3 contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception (as indicated by the corresponding CAN_IFLAG3 bit).
The CAN_IMASK3 register is not available if parameter NUMBER_OF_MB is less than 96.

Offset: 6ch

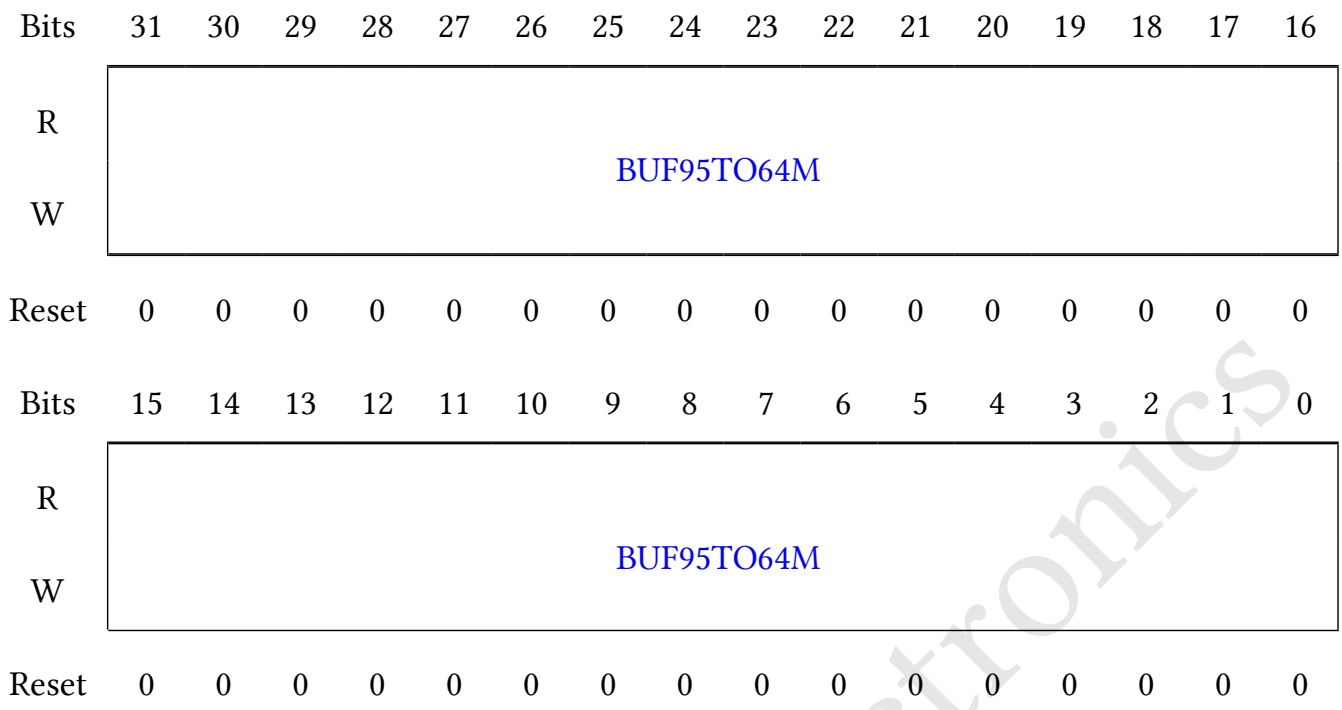


Table 20.29: CAN IMASK3 Register Description

Field	Function
31 - 0 BUF95TO64M	**Buffer MB i Mask** Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB95 to MB64. When CAN FD is enabled, the MB range is defined according to the MBDSRn bit fields of the CAN_FDCTRL register. NOTE: Setting or clearing a bit in CAN_IMASK3 can assert or negate an interrupt request if the corresponding CAN_IFLAG3 bit is set. 0b - The corresponding buffer interrupt is disabled. 1b - The corresponding buffer interrupt is enabled.

20.3.3.21 CAN IFLAG4 Register

The CAN_IFLAG4 register defines the flags for the 32 message buffer interrupts for MB127 to MB96. CAN_IFLAG4 contains one interrupt flag bit per buffer. Each successful transmission or reception sets the respective CAN_IFLAG4 bit. If the corresponding CAN_IMASK4 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.
Before updating the CAN_MCR.MAXMB field, the CPU must service the CAN_IFLAG4 bits whose MB value is greater than the CAN_MCR.MAXMB to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.
The CAN_IFLAG4 register is not available if parameter NUMBER_OF_MB is less than 128.

Offset: 70h

Yuntu Microelectronics

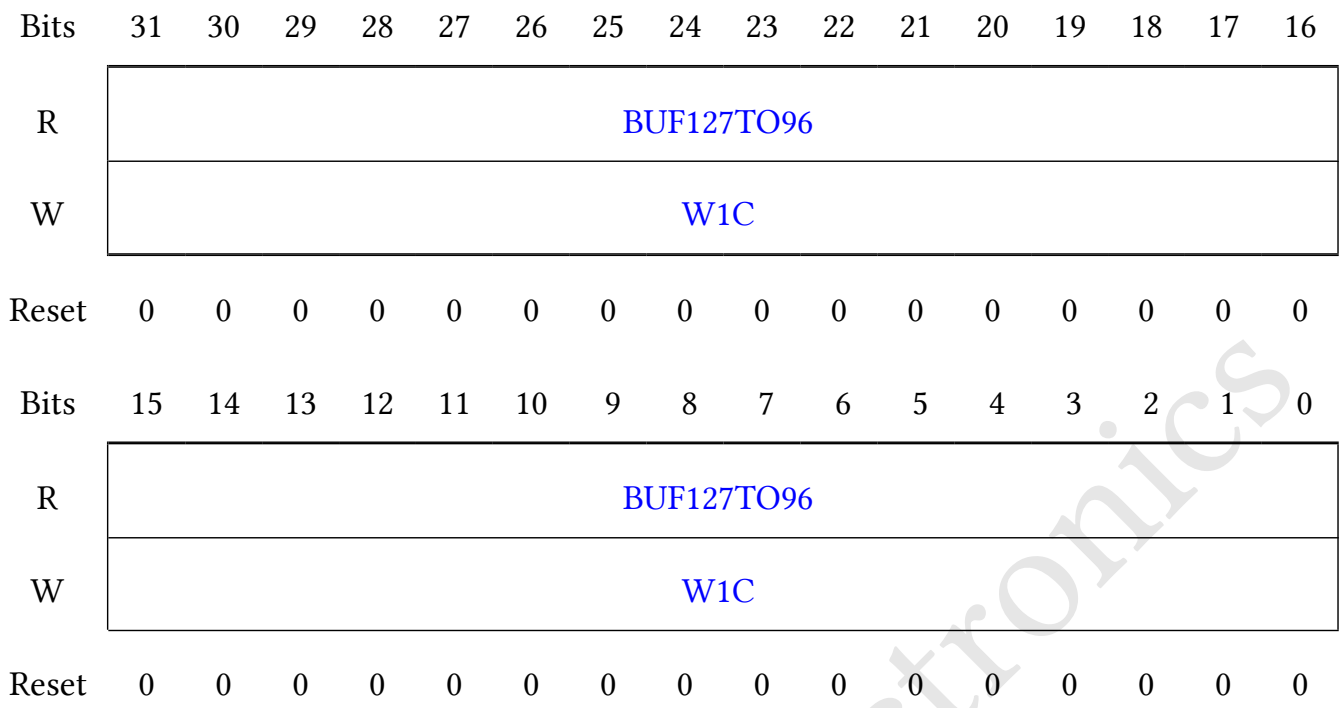


Table 20.30: CAN IFLAG4 Register Description

Field	Function
31 - 0 BUF127TO96	**Buffer MB Interrupt** Each bit flags the corresponding FlexCAN Message Buffer Interrupt for MB127 to MB96. When CAN FD is enabled, the MB range is defined according to the MBDSRn bit fields of the CAN_FDCTRL register. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

20.3.3.22 CAN IFLAG3 Register

The CAN_IFLAG3 register defines the flags for the 32 Message Buffer interrupts for MB95 to MB64. CAN_IFLAG3 contains one interrupt flag bit per buffer. Each successful transmission or reception sets the respective CAN_IFLAG3 bit. If the corresponding CAN_IMASK3 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.
Before updating the CAN_MCR.MAXMB field, the CPU must service the CAN_IFLAG3 bits whose MB value is greater than the CAN_MCR.MAXMB to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.
The CAN_IFLAG3 register is not available if parameter NUMBER_OF_MB is less than 96.

Offset: 74h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	BUF95TO64															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF95TO64															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.31: CAN IFLAG3 Register Description

Field	Function
31 - 0 BUF95TO64	**Buffer MB i Interrupt** Each bit flags the corresponding FlexCAN Message Buffer Interrupt for MB95 to MB64. When CAN FD is enabled, the MB range is defined according to the MBDSRn bit fields of the CAN_FDCTRL register. 0b - The corresponding buffer has no occurrence of successfully completed transmission or reception. 1b - The corresponding buffer has successfully completed transmission or reception.

Yuntu Microelectronics

20.3.3.23 CAN RAM Register

Offset: 80h + n * 4h

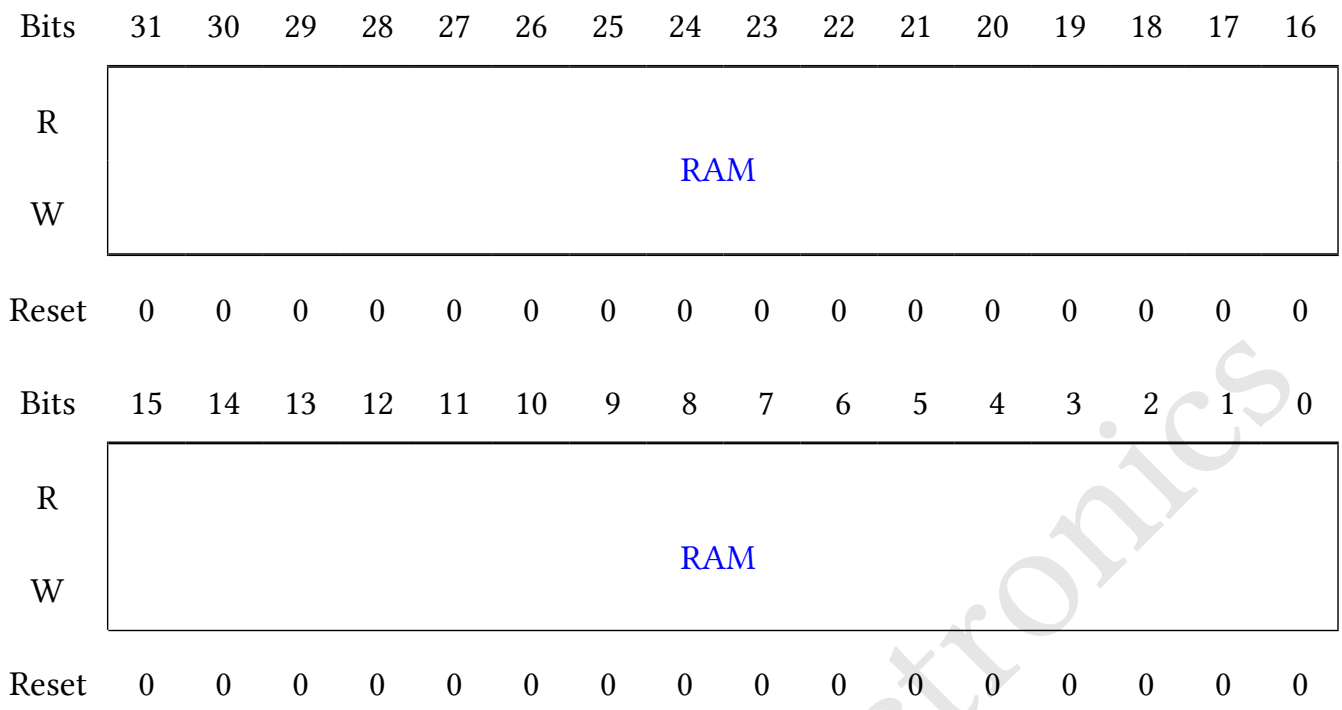


Table 20.32: CAN RAM Register Description

Field	Function
31 - 0 RAM	

20.3.3.24 CAN RXIMR Register

The RX Individual Mask Registers are used to store the acceptance masks for ID filtering in Rx MBs and the Legacy Rx FIFO.
When the Legacy Rx FIFO is disabled (CAN_MCR.RFEN is negated), an individual mask is provided for each available Rx mailbox on a one-to-one correspondence. When the Legacy Rx FIFO is enabled (CAN_MCR.RFEN is asserted), an individual mask is provided for each Legacy Rx FIFO ID Filter Table Element on a one-to-one correspondence depending on the setting of CAN_CTRL2.RFFN (see "Legacy Rx FIFO").
CAN_RXIMR0 stores the individual mask associated with either MB0 or ID Filter Table Element 0, CAN_RXIMR1 stores the individual mask associated with either MB1 or ID Filter Table Element 1, and so on.
CAN_RXIMRn registers can only be accessed by the CPU when the FlexCAN is in Freeze mode; otherwise, they are blocked by hardware. The CAN_RXIMRn registers are not affected by reset. They are located in RAM and must be explicitly initialized prior to any reception.
The CAN_RXIMRn registers are not available if parameter IRMQ_EN is 0.

Offset: 880h + n * 4h

Yuntu Microelectronics

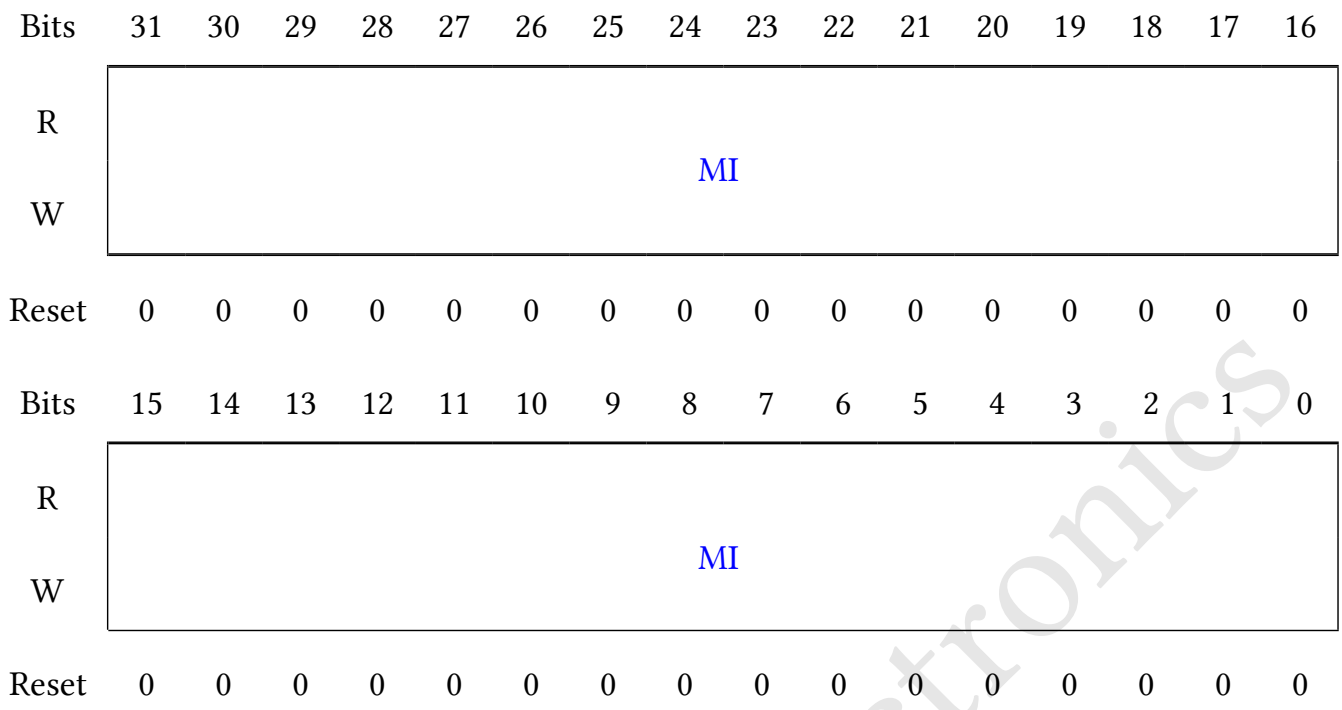


Table 20.33: CAN RXIMR Register Description

Field	Function
31 - 0 MI	**Individual Mask Bits** Each Individual Mask Bit masks the corresponding bit in both the mailbox filter and Legacy Rx FIFO ID Filter Table element in distinct ways. For mailbox filters, see the RXMGMASK register description. For Legacy Rx FIFO ID Filter Table elements, see the RXFGMASK register description. 0b - The corresponding bit in the filter is “don’t care”. 1b - The corresponding bit in the filter is checked.

Yuntu Microelectronics

20.3.3.25 CAN MECR Register

The CAN_MECR register contains control bits for memory error detection and correction (ECC).

NOTE: When CAN_CTRL2.ECRWRE is 0, writes to CAN_MECR are blocked, with the exception of the ECRWRDIS bit.

Offset: ae0h

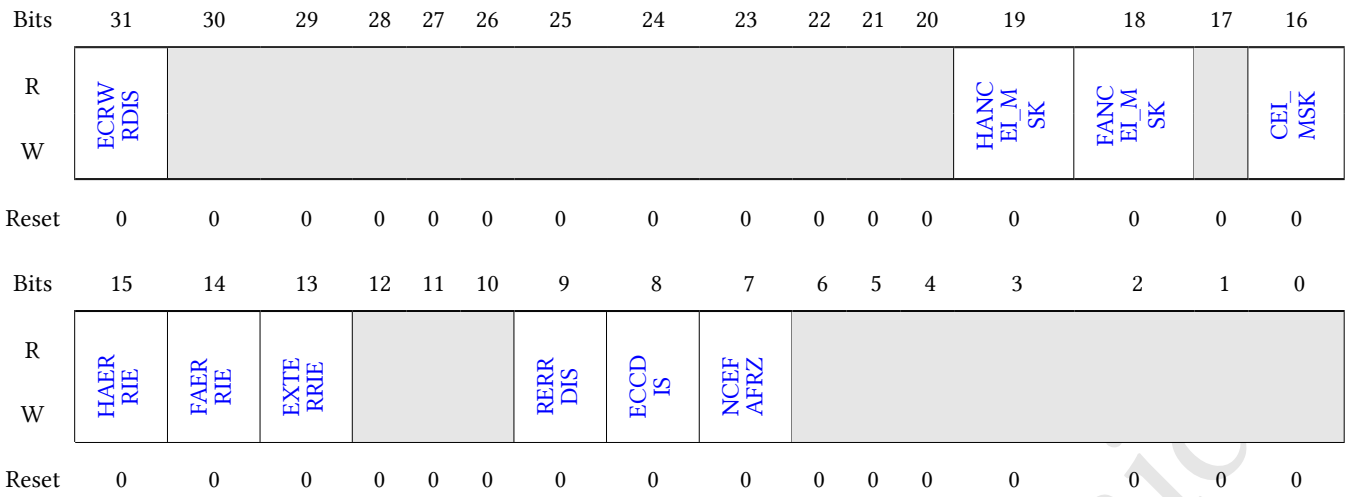


Table 20.34: CAN MECR Register Description

Field	Function
31 ECRWRDIS	**Error Configuration Register Write Disable –Disables writes to this register.** ECRWRDIS is automatically set to 1 (disabled) when CTRL2. ECRWRE is asserted. The protocol described in "Detection and Correction of Memory Errors" must be followed. 0b - Write is enabled. 1b - Write is disabled.
19 HANCEI_MSK	**Host Access With Non-Correctable Errors Interrupt Mask** Enables the interrupt in case of non-correctable errors detected in memory reads issued by the host (CPU). 0b - Interrupt is disabled. 1b - Interrupt is enabled.
18 FANCEI_MSK	**FlexCAN Access With Non-Correctable Errors Interrupt Mask** Enables the interrupt in case of non-correctable errors detected in memory reads issued by the FlexCAN internal processes. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
16 CEI_MSK	**Correctable Errors Interrupt Mask** Enables the interrupt in case of correctable errors detected in memory reads issued by the host or FlexCAN internal processes. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
15	**Host Access Error Injection Enable**

Yuntu Microelectronics

Field	Function
HAERRIE	Enables the injection of errors only in memory reads issued by the host (CPU). 0b - Error injection is disabled. 1b - Error injection is enabled.
14 FAERRIE	**FlexCAN Access Error Injection Enable** Enables the injection of errors only in memory reads issued by the FlexCAN internal processes. 0b - Error injection is disabled. 1b - Error injection is enabled.
13 EXTERRIE	**Extended Error Injection Enable** Memory accesses performed by internal FlexCAN processes are 64-bit accesses. EXTERRIE extends the error injection on 32-bit memory accesses to the complementary 32-bit word using the same 32-bit error injection data and parity words. See 'Error Injection Data Pattern Register (CAN_ERRIDPR)' and 'Error Injection Parity Pattern Register (CAN_ERRIPPR)'. 0b - Error injection is applied only to the 32-bit word. 1b - Error injection is applied to the 64-bit word.
9 RERRDIS	**Error Report Disable –Disables the update of the error report registers.** NOTE: When reading the report registers, RERRDIS must be set to ensure coherence on the consecutive register reads. 0b - Enable updates of the error report registers. 1b - Disable updates of the error report registers. The update of error-related flags and the generation of bus transfer errors are still active.
8 ECCDIS	**Error Correction Disable –Disables the memory detection and correction mechanism.** In addition to disabling the error report mechanism, ECCDIS = 1 also stops the update of the error-related flags and generation of bus transfer errors. The parity bits continue to be calculated and written into memory on write transactions. 0b - Enable memory error correction. 1b - Disable memory error correction.
7 NCEFAFRZ	**Non-Correctable Errors In FlexCAN Access Put Device In Freeze Mode** Determines the response when a non-correctable error is detected in a memory read performed by FlexCAN internal processes. 0b - Keep normal operation. 1b - Put FlexCAN in Freeze mode (according to "Freeze Mode"). In this case, entering Freeze mode prevents corrupted data from being treated as valid by FlexCAN internal processes.

Yuntu Microelectronics

20.3.3.26 CAN ERRIAR Register

The CAN_ERRIAR register holds the address where a memory read error is to be injected, as described in Error Injection .

The size and usage of the FlexCAN RAM depends on the FlexCAN hardware configuration. below table shows how to convert from the memory map address to the location in the physical FlexCAN RAM for the following example FlexCAN configuration:

- NUMBER_OF_MB = 32
- IRMQ_EN = 1
- FD_En = 1
- ENHANCE_MB_MEM = 1
- TIME_STAMP_EN = 1
- ERX_FIFO_EN = 1, ERX_FIFO_SIZE = 32, ERX_FIFO_FLT_ELEM = 128

Where pairs of values are provided, the first is the address for CAN_MCR.FDEN negated, the second is for CAN_MCR.FDEN asserted.

Table 20.35: CAN Error Injection Address Mapping for Maximum Configuration

RAM Contents	Injection Address(1)	Memory Map
FlexCAN Registers	Not mapped	—
MBs	0x0000-0x07FC	0x0080-0x087C
RXIMRs	0x1000-0x11FC	0x0880-0x0A7C
RXFIR_0	0x1200	0x0A80
RXFIR_1	0x1204	0x0A84
RXFIR_2	0x1208	0x0A88
RXFIR_3	0x120C	0x0A8C
RXFIR_4	0x1210	0x0A90
RXFIR_5	0x1214	0x0A94
Reserved	—	0x0A98
RXMGMASK	0x1220	0x0AA0
RXFGMASK	0x1224	0x0AA4
RX14MASK	0x1228	0x0AA8
RX15MASK	0x122C	0x0AAC
Tx_SMB	0x1230	0x0AB0 / 0x0F28
Rx_SMB0	0x1240 / 0x1278	0x0AC0 / 0x0F70
Rx_SMB1	0x1250 / 0x12C0	0x0AD0 / 0x0FB8
Rx_SMB0_TIME_STAMP	0x1308	0x0C20
Rx_SMB1_TIME_STAMP	0x130C	0x0C24
HR_TIME_STAMP	0x1310-0x150C	0x0C30-0xE2C
Enhanced MB Memory	0x0800-0x0FFC	0x1000-0x17FC
Enhanced Rx FIFO	0x1510-0x1F0C	0x2000-0x29FC
Enhanced Rx FIFO Filter Elements	0x1F10-0x210C	0x3000-0x31FC

See "RAM Addresses for Error Injection and Reporting" for the RAM address mapping for all FlexCAN configurations.

The CAN_ERRIAR register is not available when parameter ECC_EN is 0.

Yuntu Microelectronics

Offset: ae4h

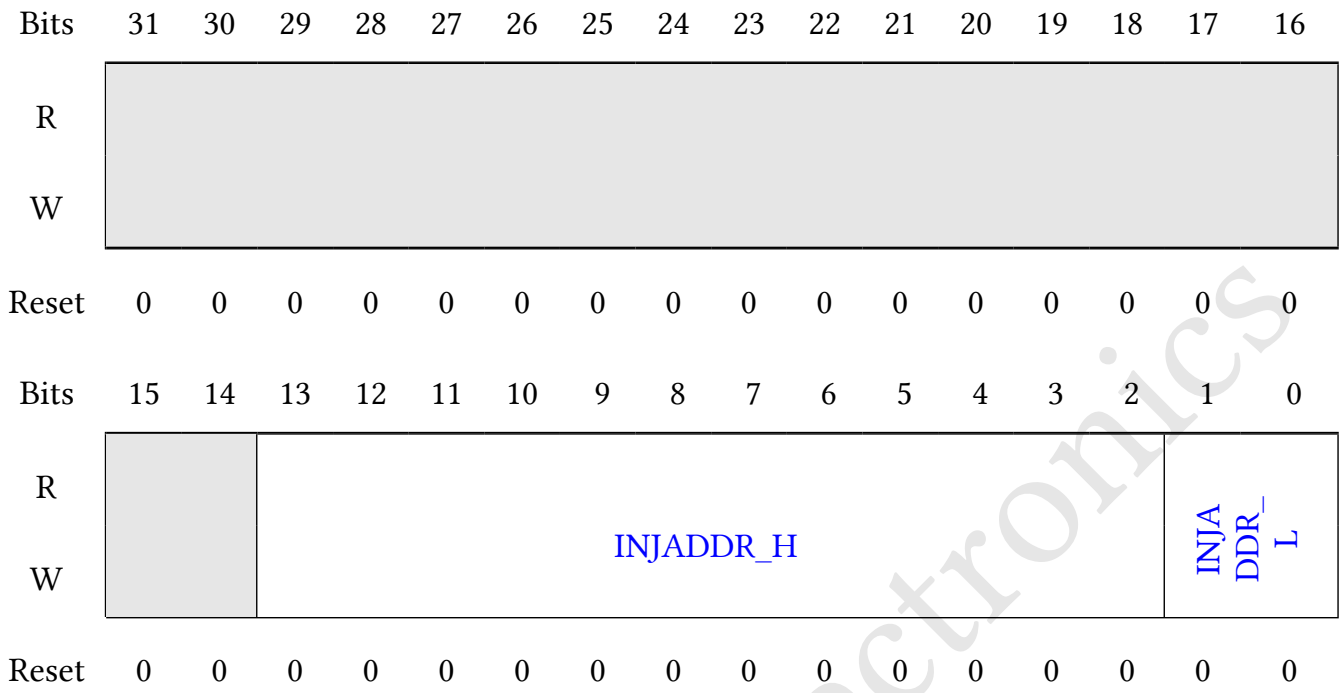


Table 20.36: CAN ERRIAR Register Description

Field	Function
13 - 2 INJADDR_H	**Error Injection Address High** Defines the twelve most significant bits of the physical RAM address where a memory read error is to be injected. See CAN error injection address mapping for maximum configuration
1 - 0 INJADDR_L	**Error Injection Address Low** Read-only field that defines the two least significant bits of the physical RAM address where a memory read error is to be injected. INJADDR_L[1:0] are always zero to ensure that the address is on a 32-bit boundary.

20.3.3.27 CAN ERRIDPR Register

The CAN_ERRIDPR register holds the error pattern to be injected in the data word read from memory. The CAN_ERRIDPR register is not available when parameter ECC_EN is 0.

Offset: ae8h

Yuntu Microelectronics

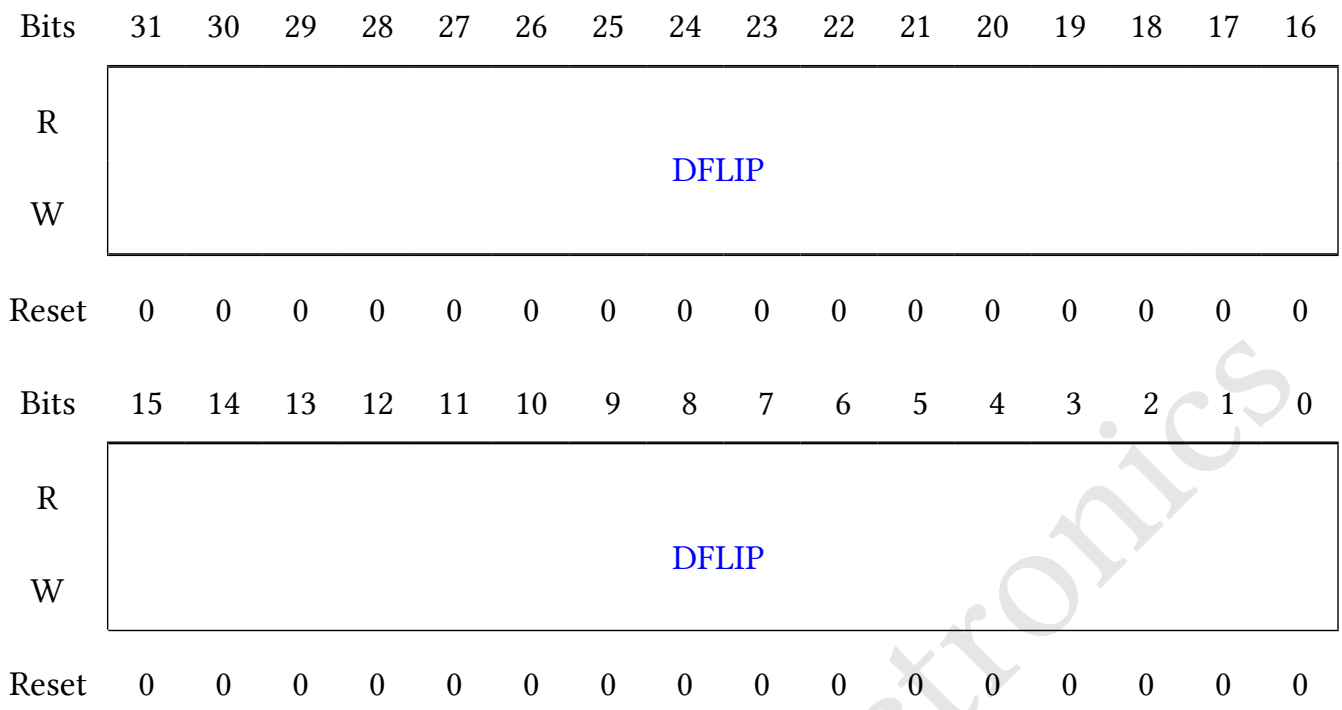


Table 20.37: CAN ERRIDPR Register Description

Field	Function
31 - 0 DFLIP	**Data Flip Pattern** Bits set to 1 in the flip pattern cause the corresponding data bit in the word read from memory to invert.

Yuntu Microelectronics

20.3.3.28 CAN ERRIPPR Register

The CAN_ERRIPPR register holds the error pattern to be injected in parity bits read from memory along with data word.
Bits set to 1 in the flip pattern cause the corresponding parity bit in the word read from memory to invert.
The CAN_ERRIPPR register is not available when parameter ECC_EN is 0.

Offset: aech

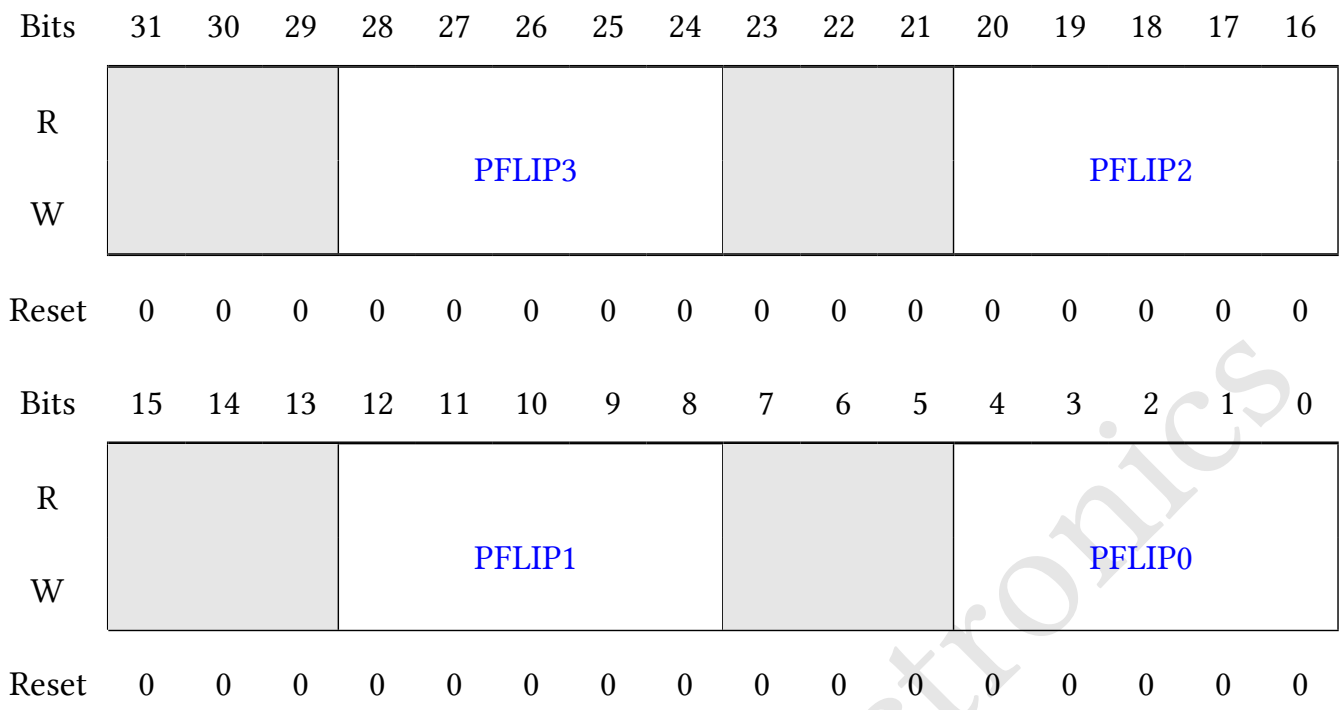


Table 20.38: CAN ERRIPPR Register Description

Field	Function
28 - 24 PFLIP3	**Parity Flip Pattern For Byte 3 (most significant)**
20 - 16 PFLIP2	**Parity Flip Pattern For Byte 2**
12 - 8 PFLIP1	**Parity Flip Pattern For Byte 1**
4 - 0 PFLIP0	**Parity Flip Pattern For Byte 0 (least significant)**

20.3.3.29 CAN RERRAR Register

The CAN_RERRAR register reports the address used for an access in which an error (correctable or non-correctable) was detected, and reports the identification of the source of that access.
The address is always reported using a 32-bit alignment. Non-aligned accesses (CAN_ERRADDR[1:0] are non-0) are reported with the address aligned and data is reported in CAN_RERRDR accordingly shifted. In case of errors detected in accesses larger than 32-bit (as performed by FlexCAN internal processes), the address of the 32-bit word in which the error was detected is reported. In case of errors detected in more than one 32-bit word, only the least significant address is reported.
The CAN_RERRAR register is not available when parameter ECC_EN is 0.

Offset: af0h

Yuntu Microelectronics

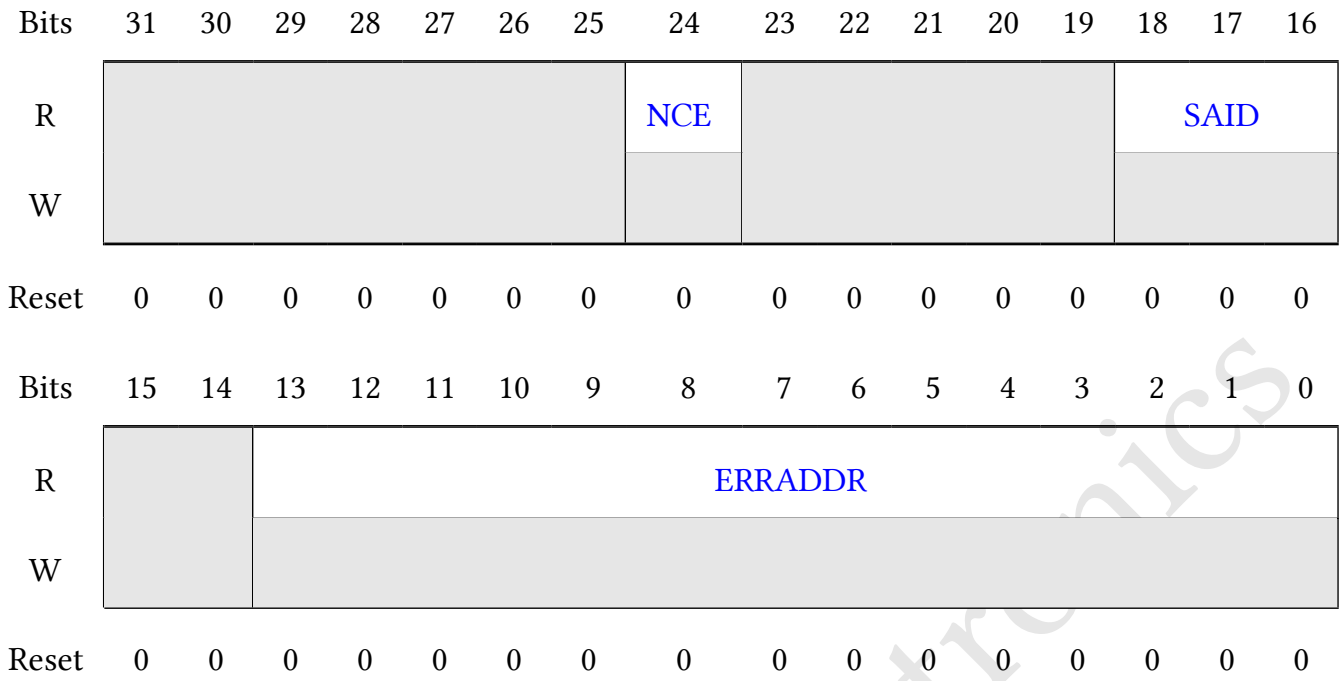


Table 20.39: CAN RERRAR Register Description

Field	Function
24 NCE	**Non-Correctable Error** Indicates that the report is due to a noncorrectable error. 0b - Reporting a correctable-error 1b - Reporting a non-correctable error
18 - 16 SAID	**SAID** SAID[2] - Identification of the requestor of the memory read request: • 0: Requested by FlexCAN internal processes • 1: Requested by host (CPU) SAID[1] - Details of FlexCAN operation: • 0: Move • 1: Scanning SAID[0] - Operation that requested the memory read: • 0: Transmission • 1: Reception
13 - 0 ERRADDR	**Address Where the Error was Detected.** See the description of Error Injection Address Register (CAN_ERRIAR) .

20.3.3.30 CAN RERRDR Register

The CAN_RERRDR register reports the raw data (unmodified by the correction performed by ECC logic) read from memory with error. The value reported does not represent the transient values of the BUSY bit when reading a message buffer (See CAN message buffer code for RX buffers). The CAN_RERRDR register is not available when parameter ECC_EN is 0.

Yuntu Microelectronics

Offset: af4h

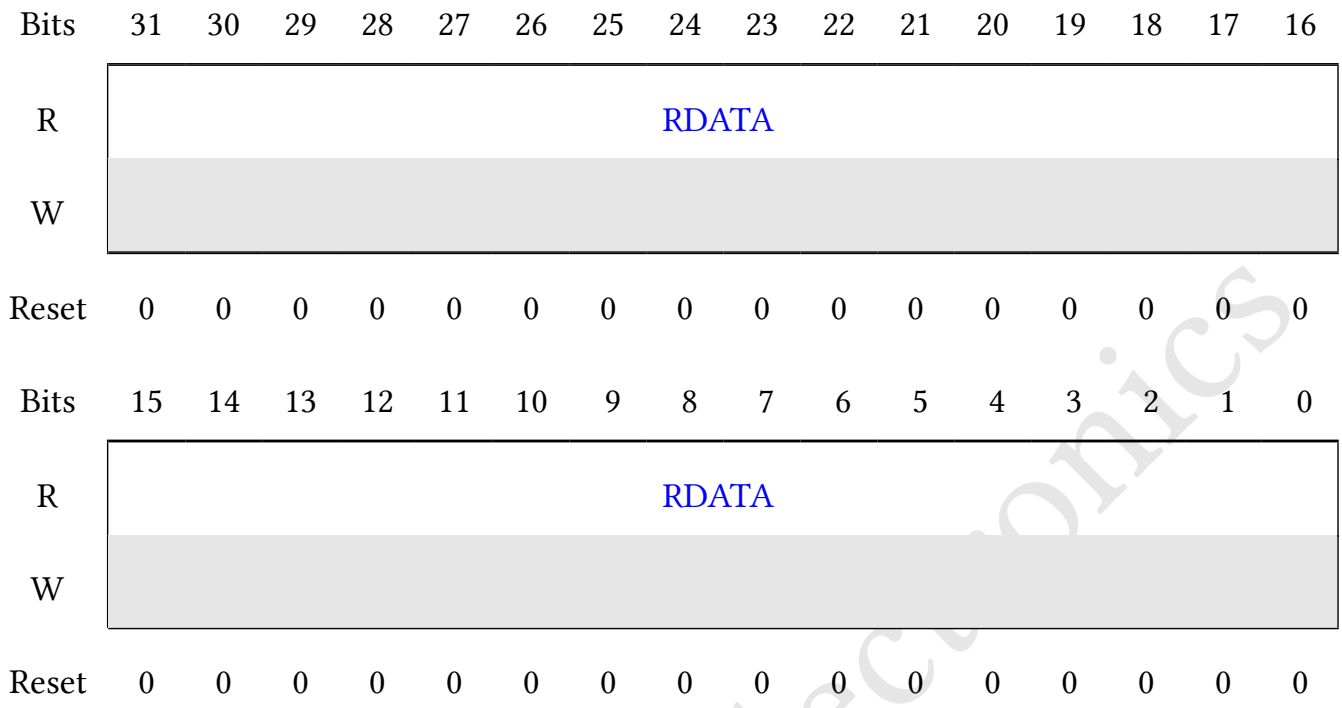


Table 20.40: CAN RERRDR Register Description

Field	Function
31 - 0 RDATA	**Raw data word read from memory with error**

20.3.3.31 CAN RERRSYNR Register

The CAN_RERRSYNR register holds the syndrome detected in a memory read with error. It also reports the bytes that were read in this 32-bit read transaction.

Each SYNDn field indicates the type of error and which bit in byte (n) is affected by the error, as defined in Table 38. (SYND3 corresponds to the most significant byte in the data word read from memory; SYND0 corresponds to the least significant.)

Each BEn field indicates which byte in the 32-bit word reported was effectively read. The syndrome bits are calculated for all bytes, even for the non-read ones. Errors detected in non-read bytes are indicated and reported. See "Error Indication" and "Error Reporting".

Table 20.41: CAN Syndrome Definition

SYNDn(hex)	Type	Bit Affected
00	—	None (no error)
01	Code	0
02	Code	1
04	Code	2
07	Data	5

Yuntu Microelectronics

Table 20.41 continued from previous page

SYNDn(hex)	Type	Bit Affected
08	Code	3
0E	Data	7
10	Code	4
13	Data	2
15	Data	6
16	Data	1
19	Data	3
1A	Data	4
1C	Data	0
06	—	All-zeros non-correctable error
1F	—	All-ones non-correctable error
All others	—	Non-correctable error

Offset: af8h

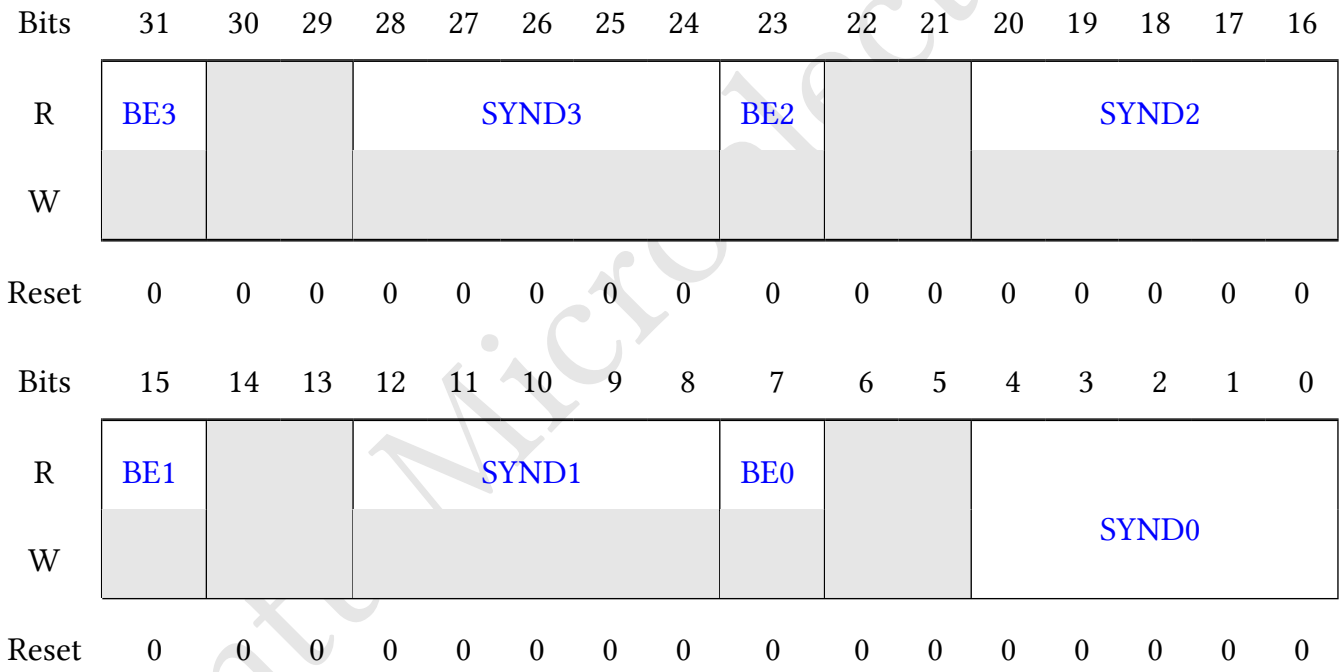


Table 20.42: CAN RERRSYNR Register Description

Field	Function
31 BE3	**Byte Enabled For Byte 3 (most significant).** 0b - The byte was not read. 1b - The byte was read.
28 - 24 SYND3	**Error Syndrome For Byte 3 (most significant)** See CAN syndrome definition.
23	**Byte Enabled For Byte 2**

Yuntu Microelectronics

Field	Function
BE2	0b - The byte was not read. 1b - The byte was read.
20 - 16 SYND2	**Error Syndrome For Byte 2** See CAN syndrome definition.
15 BE1	**Byte Enabled For Byte 1** 0b - The byte was not read. 1b - The byte was read.
12 - 8 SYND1	**Error Syndrome For Byte 1** See CAN syndrome definition.
7 BE0	**Byte Enabled For Byte 0 (least significant)** 0b - The byte was not read. 1b - The byte was read.
4 - 0 SYND0	**Error Syndrome For Byte 0 (least significant)** See CAN syndrome definition.

20.3.3.32 CAN ERRSR Register

The CAN_ERRSR register holds the status bits of the error correction and detection operations. These flags can be cleared by writing 1 to them. Writing 0 has no effect. The CAN_ERRSR register is not available when parameter ECC_EN is 0.

Offset: afch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R													HANCEIF	FANCEIF		CEIF
W													W1C	W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													HANCEIOF	FANCEIOF		CEIOF
W													W1C	W1C		W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 20.43: CAN ERRSR Register Description

Field	Function
19 HANCEIF	**Host Access With Non-Correctable Error Interrupt Flag** HANCEIF = 1 indicates that a non-correctable error was detected in a memory read initiated by the Hhost. A bus transfer error is asserted for that access. If CAN_MECR.HANCEI_MSK is set, the interrupt is asserted. 0b - No non-correctable errors were detected in host accesses so far. 1b - A non-correctable error was detected in a host access.
18 FANCEIF	**FlexCAN Access With Non-Correctable Error Interrupt Flag** FANCEIF = 1 indicates that a non-correctable error was detected in a memory read initiated by FlexCAN internal processes. If CAN_MECR.FANCEI_MSK is set, the interrupt is asserted. 0b - No non-correctable errors were detected in FlexCAN accesses so far. 1b - A non-correctable error was detected in a FlexCAN access.
16 CEIF	**Correctable Error Interrupt Flag** CEIF = 1 indicates that a correctable error was detected in a memory read. If CAN_MECR.CEI_MSK is set, the interrupt is asserted. 0b - No correctable errors were detected so far. 1b - A correctable error was detected.
3 HANCEIOF	**Host Access With Non-Correctable Error Interrupt Overrun Flag** HANCEIOF = 1 indicates that a non-correctable error was detected in a memory read initiated by the host when HANCEIF was set. No interrupt is associated with this flag. See "Error Indication". 0b - No overrun on non-correctable errors in host access. 1b - Overrun on non-correctable errors in host access.
2 FANCEIOF	**FlexCAN Access With Non-Correctable Error Interrupt Overrun Flag** FANCEIOF = 1 indicates that a non-correctable error was detected in a memory read initiated by FlexCAN internal processes when FANCEIF was set. No interrupt is associated with this flag. See "Error Indication". 0b - No overrun on non-correctable errors in FlexCAN access. 1b - Overrun on non-correctable errors in FlexCAN access.
0 CEIOF	**Correctable Error Interrupt Overrun Flag** CEIOF = 1 indicates that a correctable error was detected in a memory read when CEIF was set. No interrupt is associated with this flag. See "Error Indication". 0b - No overrun on correctable errors. 1b - Overrun on correctable errors.

20.3.3.33 CAN CTRL1_PN Register

The CAN_CTRL1_PN register contains control bits for Pretended Networking mode filtering selection. Configure CAN_CTRL1_PN with the filter criteria to be used to receive Wake Up messages.

Yuntu Microelectronics

CAN_CTRL1_PN bits other than WTOF_MSK and WUMF_MSK can be written in Freeze mode only. The CAN_CTRL1_PN register is not available when parameter PNET_EN is 0.

Offset: b00h

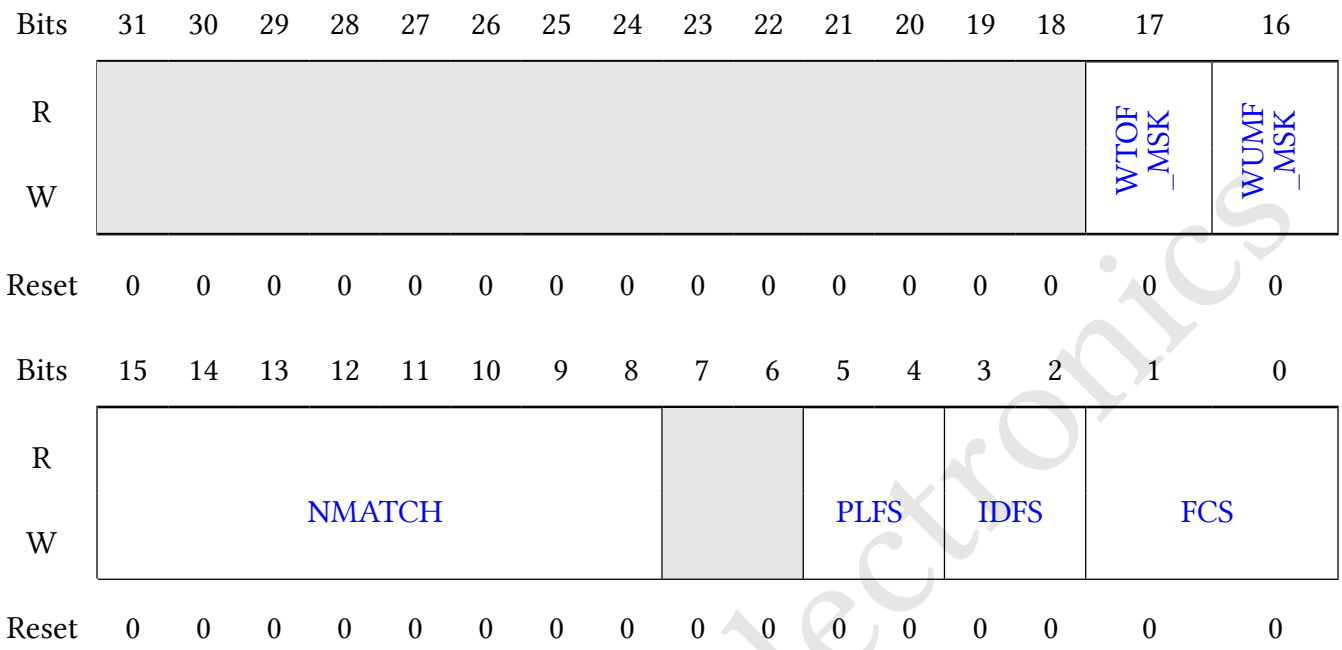


Table 20.44: CAN CTRL1_PN Register Description

Field	Function
17 WTOF_MSK	**Wake Up by Timeout Flag Mask Bit** Masks the generation of a Wake Up event originated by a timeout 0b - Timeout Wake Up event is disabled. 1b - Timeout Wake Up event is enabled.
16 WUMF_MSK	**Wake Up by Match Flag Mask Bit** Masks the generation of a Wake Up event originated by a successful filtered Rx message 0b - Wake Up match event is disabled. 1b - Wake Up match event is enabled.
15 - 8	

Yuntu Microelectronics

Field	Function
NMATCH	Number of Messages Matching the Same Filtering Criteria NMATCH defines the number of times a given message must match the predefined filtering criteria for ID and/or PL before generating a Wake Up event. This quantity can be configured in the 1 to 255 range by using values from 0x01 to 0xFF, respectively 00000001b (0x01) - Received message must match the predefined filtering criteria for ID and/or PL once before generating a Wake Up event. 00000010b (0x02) - Received message must match the predefined filtering criteria for ID and/or PL twice before generating a Wake Up event. ..... 11111111b (0xff) - Received message must match the predefined filtering criteria for ID and/or PL 255 times before generating a Wake Up event.
5 - 4 PLFS	**Payload Filtering Selection** Selects the level of payload filtering to be applied when FlexCAN is under Pretended Networking mode. Filtering does not accept remote messages (RTR=1) when payload filtering is active. 00b - Match upon a payload contents against an exact target value. 01b - Match upon a payload value greater than or equal to a specified target value. 1010b - Match upon a payload value smaller than or equal to a specified target value. 1011b - Match upon a payload value inside a range, greater than or equal to a specified lower limit, and smaller than or equal to a specified upper limit.
3 - 2 IDFS	**ID Filtering Selection –Selects the level of ID filtering to be applied when FlexCAN is under Pretended Networking mode.** In ID filtering, the IDE and RTR bits are also considered as part of the reception filter if the IDE_MSK and RTR_MSK bits in the CAN_FLT_ID2_IDMASK register are set. 00b - Match upon a ID contents against an exact target value. 01b - Match upon a ID value greater than or equal to a specified target value. 1010b - Match upon a ID value smaller than or equal to a specified target value. 1011b - Match upon a ID value inside a range, greater than or equal to a specified lower limit, and smaller than or equal to a specified upper limit.
1 - 0 FCS	**Filtering Combination Selection –Selects the filtering criteria to be applied when FlexCAN is under Pretended Networking mode.** See "Receive Process under Pretended Networking Mode") for additional details. 00b - Message ID filtering only. 01b - Message ID filtering and payload filtering. 1010b - Message ID filtering occurring a specified number of times. 1011b - Message ID filtering and payload filtering a specified number of times.

20.3.3.34 CAN CTRL2_PN Register

The CAN_CTRL2_PN register contains configuration bits for the timeout value under Pretended Networking mode. CAN_CTRL2_PN can be written in Freeze mode only.
The CAN_CTRL2_PN register is not available when parameter PNET_EN is 0.

Yuntu Microelectronics

Offset: b04h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MATCHTO															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.45: CAN CTRL2_PN Register Description

Field	Function
15 - 0 MATCHTO	Timeout for No Message Matching the Filtering Criteria MATCHTO defines a timeout value that generates a Wake Up event if CAN_MCR.PNET_EN is asserted. If the timeout counter reaches the target value when the FlexCAN is under Pretended Networking mode, a Wake Up event is generated. The timeout limit can be configured from 1 to 65535 to control an internal 16-bit up-count timer to produce a trigger upon reaching this configured value. The internal timer is incremented based on periodic time ticks, of which the period is 64 times the CAN Bit Time unit. When MATCHTO is 0x0000, the timeout is disabled.

Yuntu Microelectronics

20.3.3.35 CAN WU_MTC Register

The CAN_WU_MTC register contains Wake Up information related to the matching processes performed while the FlexCAN receives frames under Pretended Networking mode.
The CAN_WU_MTC register is not available when parameter PNET_EN is 0.

Offset: b08h

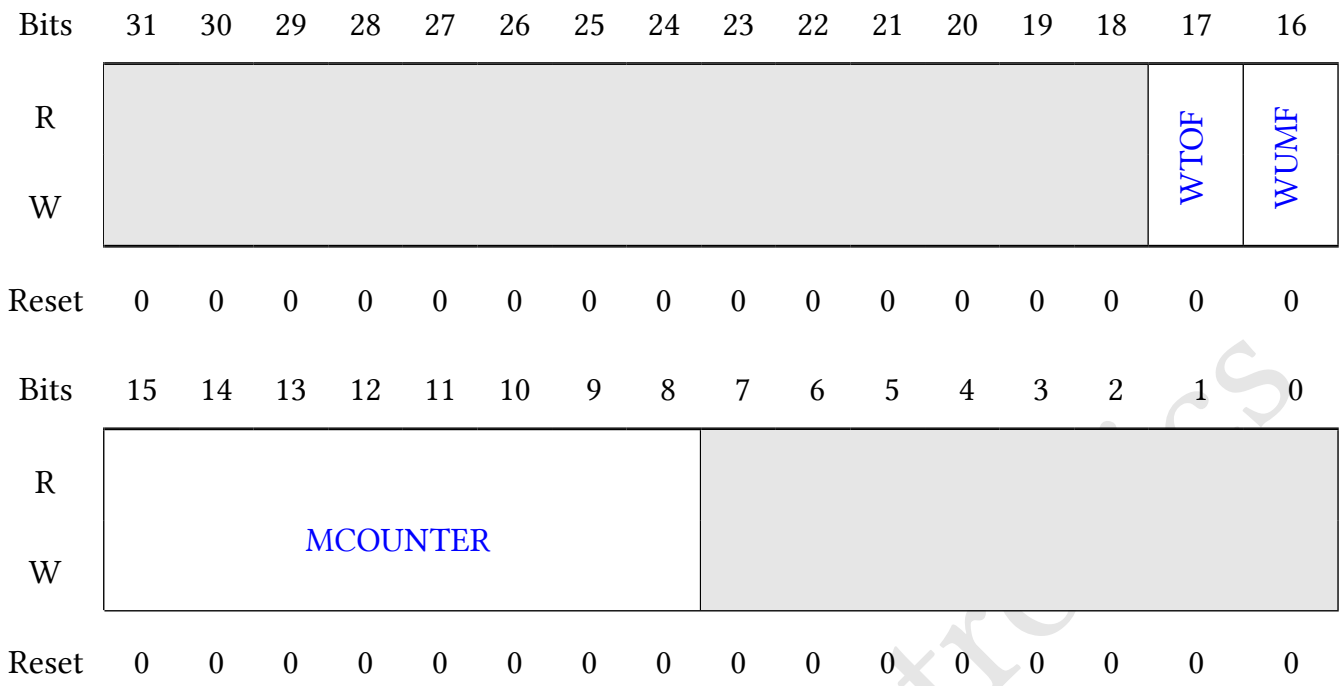


Table 20.46: CAN WU_MTC Register Description

Field	Function
17 WTOF	**Wake Up by Timeout Flag Bit –Indicates whether the FlexCAN has detected a timeout event during a time interval defined by CAN_CTRL2_PN.MATCHTO.** WTOF generates a Wake Up event if CAN_CTRL1_PN.WTOF_MSK is asserted. 0b - No Wake Up by timeout event detected. 1b - Wake Up by timeout event detected.
16 WUMF	**Wake Up by Match Flag Bit** Indicates whether the FlexCAN has detected a matching Rx incoming message that passed the filtering criteria specified in CAN_CTRL1_PN. WUMF generates a Wake Up event if CAN_CTRL1_PN.WUMF_MSK is asserted. 0b - No Wake Up by match event detected. 1b - Wake Up by match event detected.
15 - 8 MCOUNTER	Number of Matches while in Pretended Networking MCOUNTER reports the number of times a given message has matched the predefined filtering criteria for ID and/or PL before a Wake Up event. MCOUNTER is reset by the FlexCAN when it enters in Pretended Networking mode, and is not affected by soft reset.

20.3.3.36 CAN FLT_ID1 Register

The CAN_FLT_ID1 register contains the FLT_ID1 target value and the IDE and RTR target values used to filter an incoming message ID. FLT_ID1 is used either for equal to, smaller than, greater than

comparisons, or as the lower limit value in an ID range detection. CAN_FLT_ID1 can be written in Freeze mode only.
The CAN_FLT_ID1 register is not available when parameter PNET_EN is 0.

Offset: b0ch

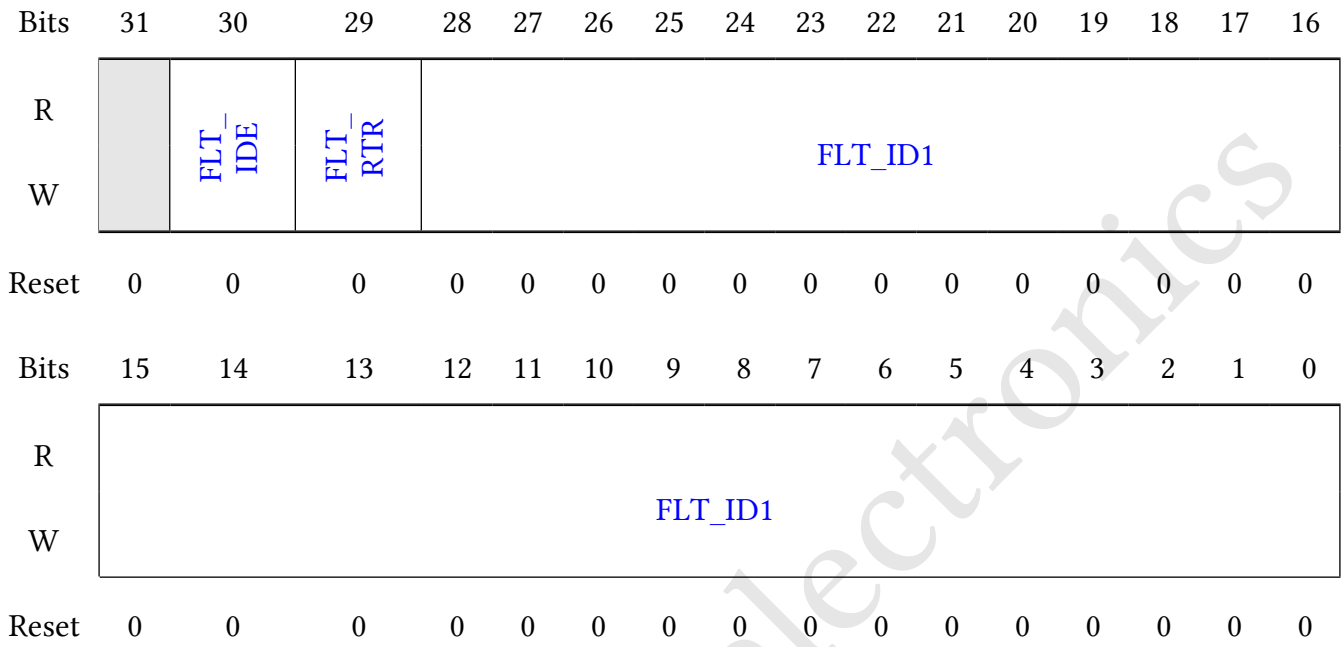


Table 20.47: CAN FLT_ID1 Register Description

Field	Function
30 FLT_IDE	**D Extended Filter –Identifies whether the frame format is standard or extended.** FLT_IDE is used as part of the ID reception filter. 0b - Accept standard frame format. 1b - Accept extended frame format.
29 FLT_RTR	**Remote Transmission Request Filter –Identifies whether the frame is remote or not.** FLT_RTR is used as part of the ID reception filter. 0b - Reject remote frame (accept data frame). 1b - Accept remote frame.
28 - 0 FLT_ID1	**ID Filter 1 for Pretended Networking Filtering** Defines either the 29 bits of an extended frame format, considering all bits, or the 11 bits of a standard frame format, considering just the 11 leftmost bits.

20.3.3.37 CAN FLT_DLC Register

The CAN_FLT_DLC register contains the DLC inside range target values (FLT_DLC_LO and FLT_DLC_HI) used to filter an incoming message. The DLC range is used only for payload filtering. CAN_FLT_DLC can be written in Freeze mode only.

Yuntu Microelectronics

NOTE: When a fixed quantity of data bytes is required, both FLT_DLC_LO and FLT_DLC_HI need to be configured with the same value; otherwise, a range of DLC is considered for filtering (see "Receive Process under Pretended Networking Mode").
The CAN_FLT_DLC register is not available when parameter PNET_EN is 0.

Offset: b10h

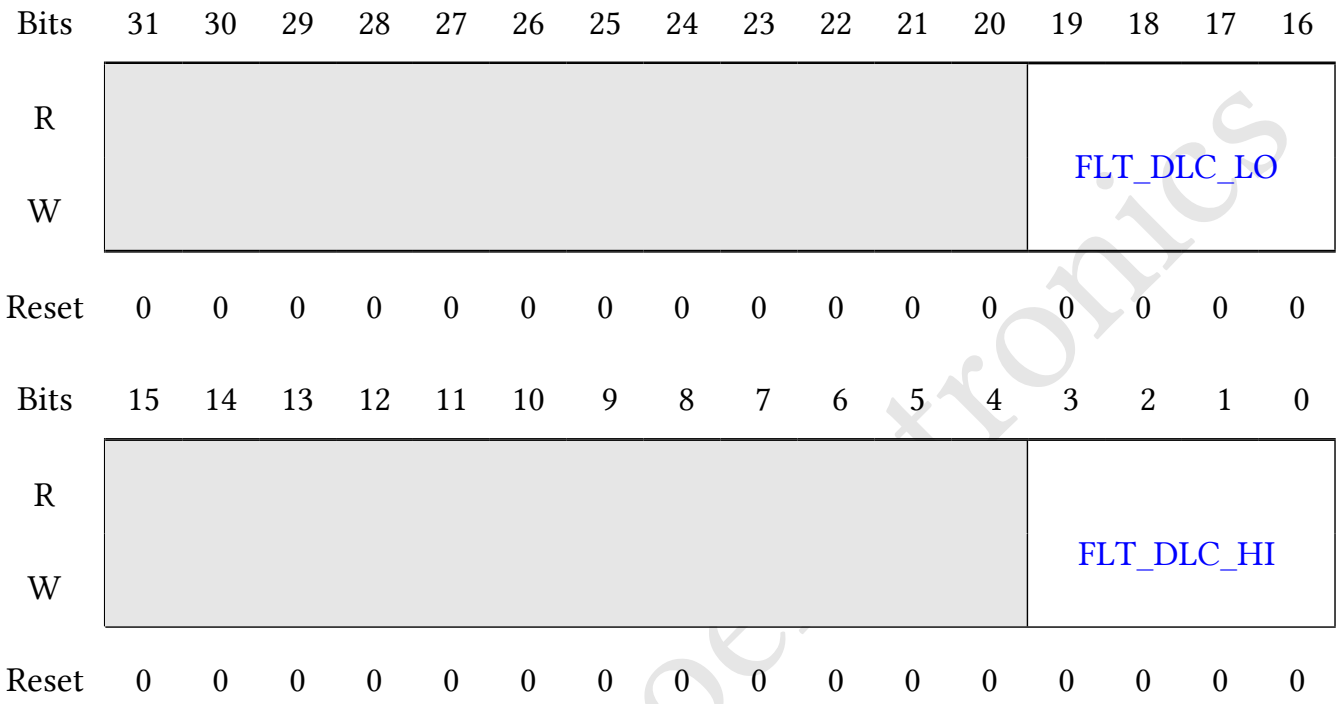


Table 20.48: CAN FLT_DLC Register Description

Field	Function
19 - 16 FLT_DLC_ LO	**Lower Limit for Length of Data Bytes Filter** Specifies the lower limit for the number of data bytes considered valid for payload comparison. FLT_DLC_LO is used as part of payload reception filter.
3 - 0 FLT_DLC_ HI	**Upper Limit for Length of Data Bytes Filter** Specifies the upper limit for the number of data bytes considered valid for payload comparison. FLT_DLC_HI is used as part of payload reception filter.

20.3.3.38 CAN PL1_LO Register

The CAN_PL1_LO register contains Payload Filter 1 low order bits of the target value used to filter incoming message payload. It is used either for "equal to", "smaller than or equal to", "greater than or equal to" comparisons, or as the lower limit value in a payload range detection. CAN_PL1_LO can be written in Freeze mode only.
The CAN_PL1_LO register is not available when parameter PNET_EN is 0.

Offset: b14h

Yuntu Microelectronics

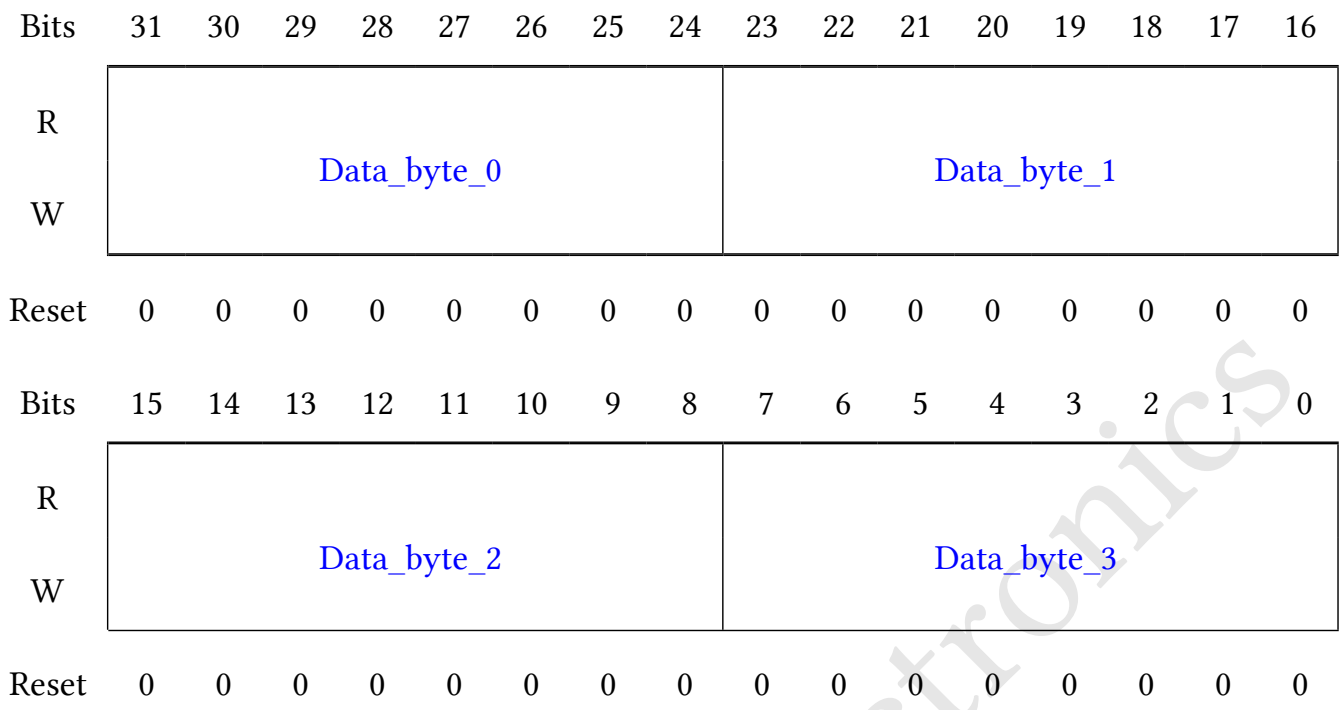


Table 20.49: CAN PL1_LO Register Description

Field	Function
31 - 24 Data_byte_0	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 0.
23 - 16 Data_byte_1	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 1.
15 - 8 Data_byte_2	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 2.
7 - 0 Data_byte_3	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 3.

20.3.3.39 CAN PL1_HI Register

The CAN_PL1_HI register contains Payload Filter 1 high order bits of the target value used to filter incoming message payload. It is used either for "equal to", "smaller than or equal to", "greater than or equal to" comparisons, or as the lower limit value in a payload range detection. CAN_PL1_HI can be written in Freeze mode only.

Yuntu Microelectronics

The CAN_PL1_HI register is not available when parameter PNET_EN is 0.

Offset: b18h

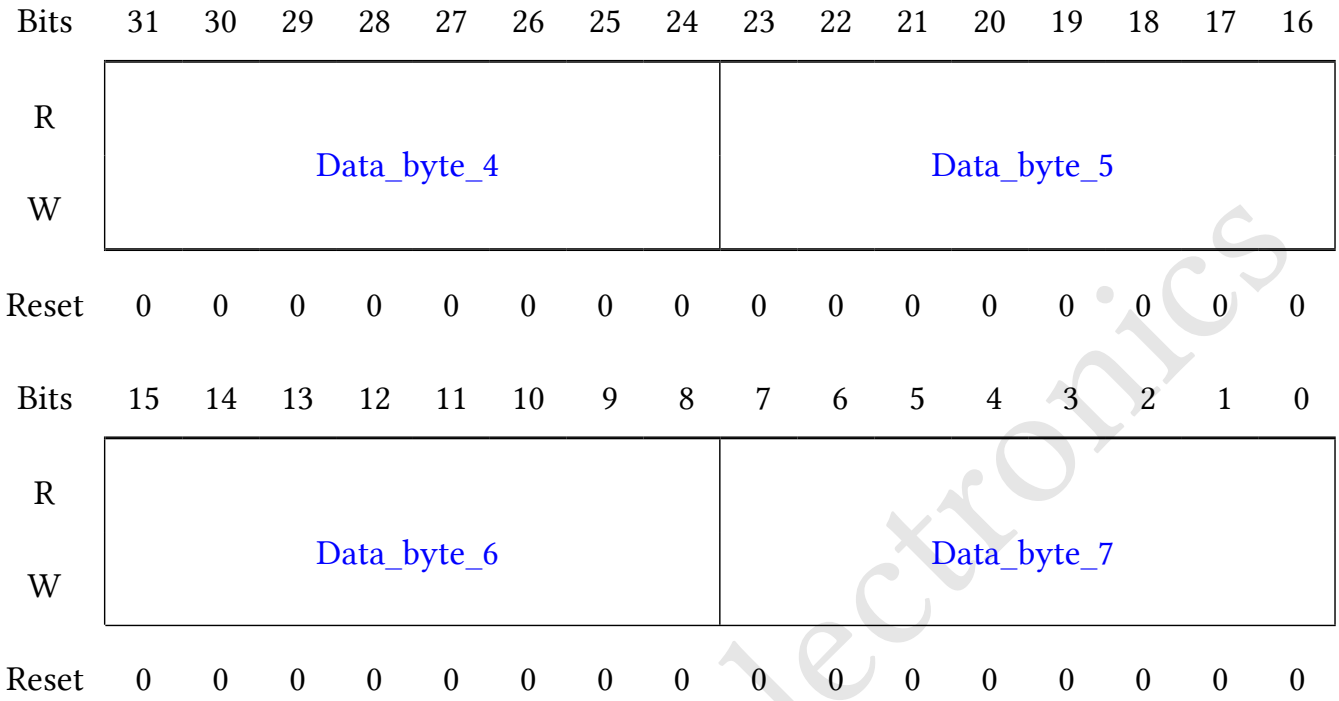


Table 20.50: CAN PL1_HI Register Description

Field	Function
31 - 24 Data_byte_4	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 4.
23 - 16 Data_byte_5	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 5.
15 - 8 Data_byte_6	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 6.
7 - 0 Data_byte_7	Payload Filter 1 low order bits for Pretended Networking payload filtering corresponding to data byte 7.

Yuntu Microelectronics

20.3.3.40 CAN FLT_ID2_IDMASK Register

The CAN_FLT_ID2_IDMASK register contains the FLT_ID2 target value used only as the upper limit value in ID range detection. Also, when exact ID filtering criteria is selected, CAN_FLT_ID2_IDMASK is used to store the ID mask. The IDE_MSK and RTR_MSK bits are used in both ID filtering (exact and range) to enable FLT_IDE and FLT_RTR respectively to be used as part of ID reception filter.
CAN_FLT_ID2_IDMASK can be written in Freeze mode only.
The CAN_FLT_ID2_IDMASK register is not available when parameter PNET_EN is 0.

Offset: b1ch

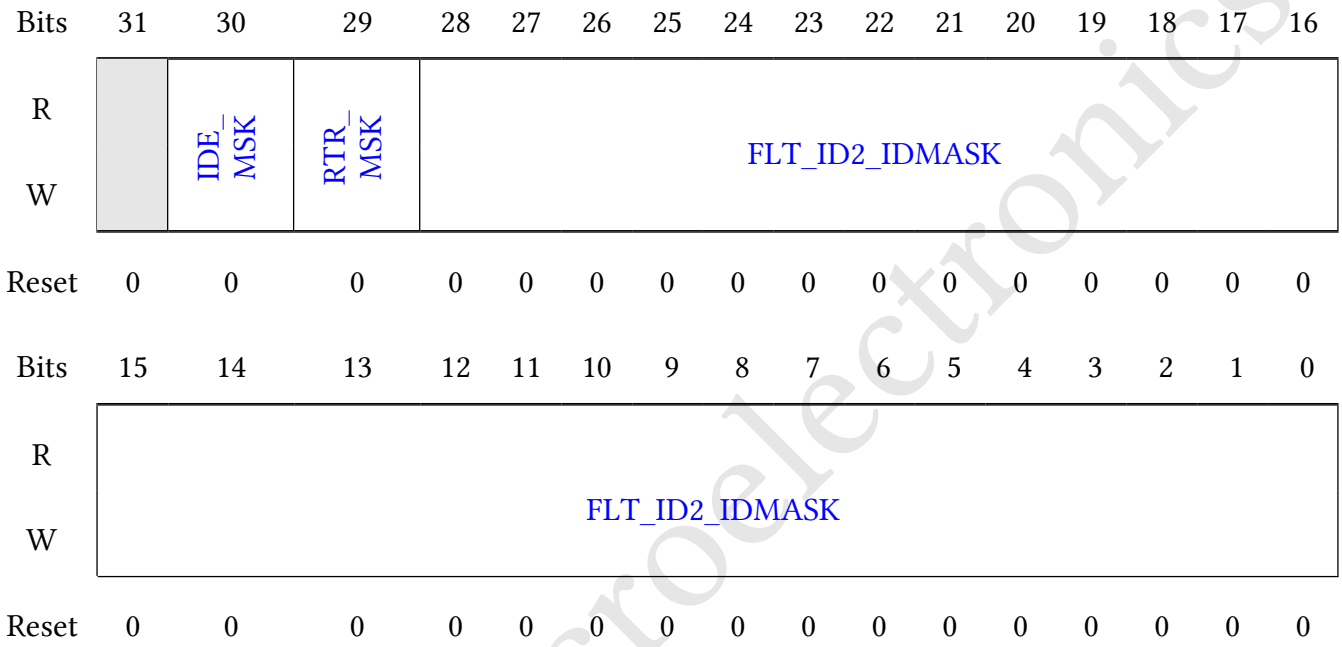


Table 20.51: CAN FLT_ID2_IDMASK Register Description

Field	Function
30 IDE_MSK	**ID Extended Mask Bit –Indicates whether the frame format (standard/extended) is used as part of the ID reception filter.** 0b - The corresponding bit in the filter is "don't care". 1b - The corresponding bit in the filter is checked.
29 RTR_MSK	Remote Transmission Request Mask Bit –Indicates whether the frame type (data/remote) is part of the ID reception filter. 0b - The corresponding bit in the filter is "don't care". 1b - The corresponding bit in the filter is checked.
28 - 0	

Yuntu Microelectronics

Field	Function
FLT_ID2_ IDMASK	ID Filter 2 for Pretended Networking Filtering / ID Mask Bits for Pretended Networking ID Filtering FLT_ID2_IDMASK is used as FLT_ID2 in range of ID filtering to define the ID filter value in either: • Extended frame format (29 bits), considering the FLT_ID2[28:0], or • Standard frame format (11 bits), considering the FLT_ID2[28:18]. Other bits in the [17:0] range have no meaning. FLT_ID2_IDMASK can also be used as IDMASK in exact ID filtering to define the mask value for either: • Extended frame format (29 bits), considering the IDMASK[28:0], or • Standard frame format (11 bits), considering the IDMASK[28:18]. Other bits in the [17:0] range have no meaning.

20.3.3.41 CAN PL2_PLMASK_LO Register

The CAN_PL2_PLMASK_LO register has two functions:
• It contains the low order bits for the Payload Filter 2 used only as the upper limit value in a payload range detection.
• When exact payload filtering criteria is selected, CAN_PL2_PLMASK_LO is used as the payload mask. Otherwise, CAN_PL2_PLMASK_LO is unused. CAN_PL2_PLMASK_LO can be written in Freeze mode only.
The CAN_PL2_PLMASK_LO register is not available when parameter PNET_EN is 0.

Offset: b20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Data_byte_0								Data_byte_1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Data_byte_2								Data_byte_3							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 20.52: CAN PL2_PLMASK_LO Register Description

Field	Function
31 - 24 Data_byte_0	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to data byte 0.
23 - 16 Data_byte_1	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to data byte 1.
15 - 8 Data_byte_2	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to data byte 2.
7 - 0 Data_byte_3	Payload Filter 2 low order bits / Payload Mask low order bits for Pretended Networking payload filtering corresponding to data byte 3.

Yuntu Microelectronics

20.3.3.42 CAN PL2_PLMASK_HI Register

The CAN_PL2_PLMASK_HI register has two functions:
• It contains the high order bits for the Payload Filter 2 used only as the upper limit value in a payload range detection.
• When exact payload filtering criteria is selected, CAN_PL2_PLMASK_HI is used as the payload mask. Otherwise, CAN_PL2_PLMASK_HI is unused. CAN_PL2_PLMASK_HI can be written in Freeze mode only.
The CAN_PL2_PLMASK_HI register is not available when parameter PNET_EN is 0.

Offset: b24h

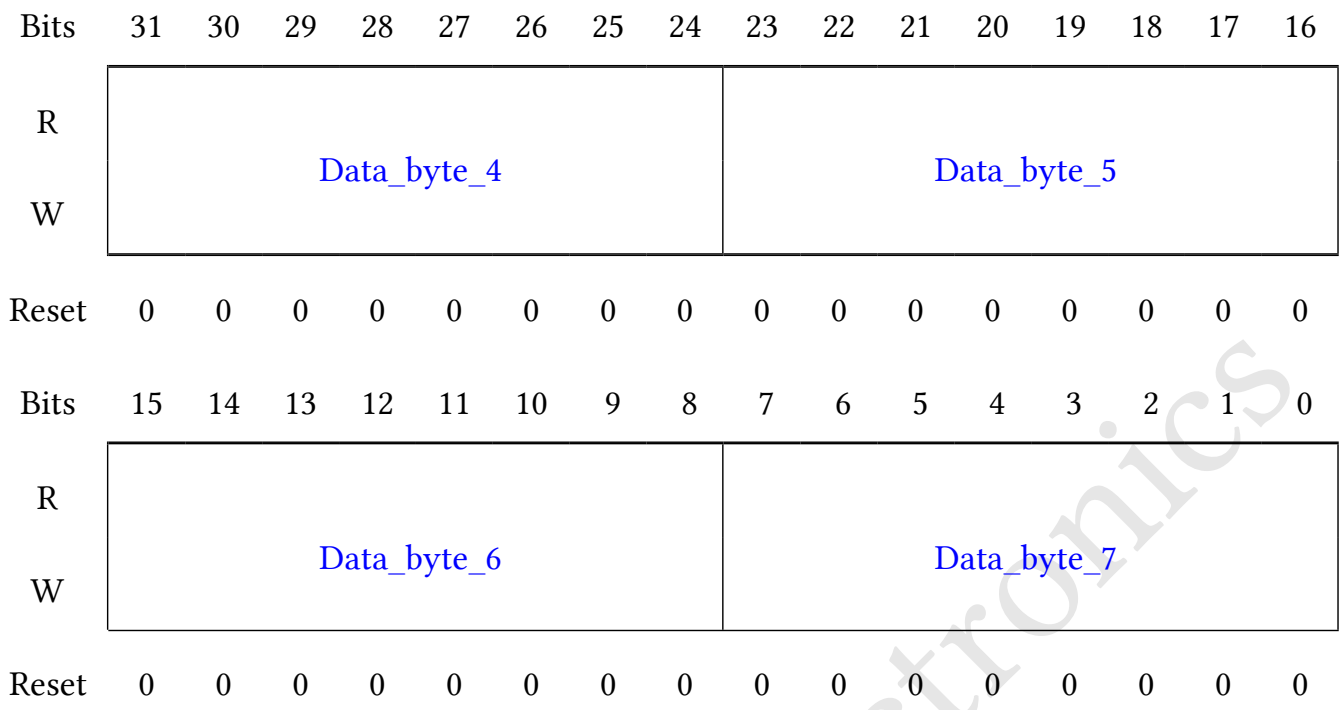


Table 20.53: CAN PL2_PLMASK_HI Register Description

Field	Function
31 - 24 Data_byte_4	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to data byte 4.
23 - 16 Data_byte_5	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to data byte 5.
15 - 8 Data_byte_6	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to data byte 6.
7 - 0 Data_byte_7	Payload Filter 2 high order bits / Payload Mask high order bits for Pretended Networking payload filtering corresponding to data byte 7.

20.3.3.43 CAN WMB_WMBn_CS Register

Each of the four WMBs contains a register to store the Control Status (C/S) information (IDE, RTR, and DLC fields) of an incoming Rx message.
NOTE: The C/S registers are located at 0xB40 for WMB0, 0xB50 for WMB1, 0xB60 for WMB2, and 0xB70 for WMB3.

Yuntu Microelectronics

The CAN_WMBn_CS registers are not available when parameter PNET_EN is 0.

Offset: b40h + n * 10h

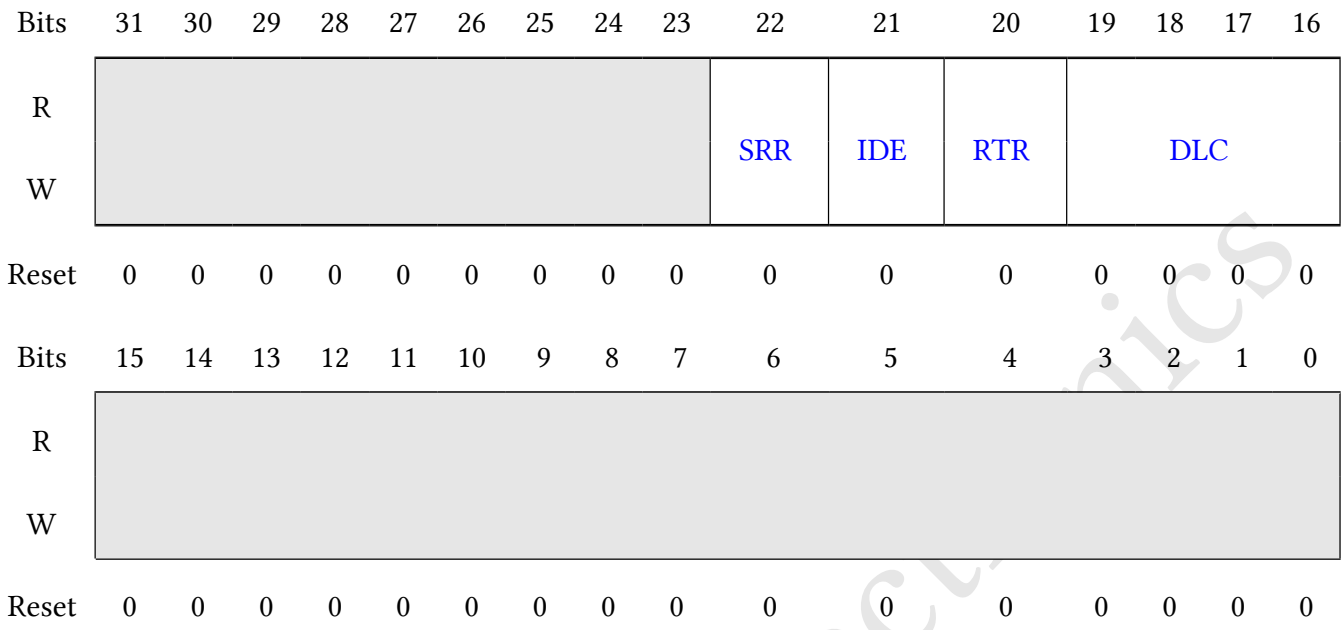


Table 20.54: CAN WMB_WMBn_CS Register Description

Field	Function
22 SRR	**Substitute Remote Request –Can be received either recessive or dominant.**
21 IDE	**ID Extended Bit –Identifies whether the frame format is standard or extended.** 0b - Frame format is standard. 1b - Frame format is extended.
20 RTR	**Remote Transmission Request Bit –Identifies whether the frame is remote or not.** 0b - Frame is data frame (not remote). 1b - Frame is a remote frame.
19 - 16 DLC	Length of Data in Bytes –Length (in bytes) of the Rx data received when FlexCAN is in Pretended Networking mode. DLC is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. The DLC field indicates which data bytes are valid.

Yuntu Microelectronics

20.3.3.44 CAN WMB_WMBn_ID Register

Each of the four WMBs contains a register to store the ID information of an incoming Rx message.
NOTE: The ID registers are located at 0xB44 for WMB0, 0xB54 for WMB1, 0xB64 for WMB2, and 0xB74 for WMB3.
The CAN_WMBn_ID registers are not available when parameter PNET_EN is 0.

Offset: b44h + n * 10h

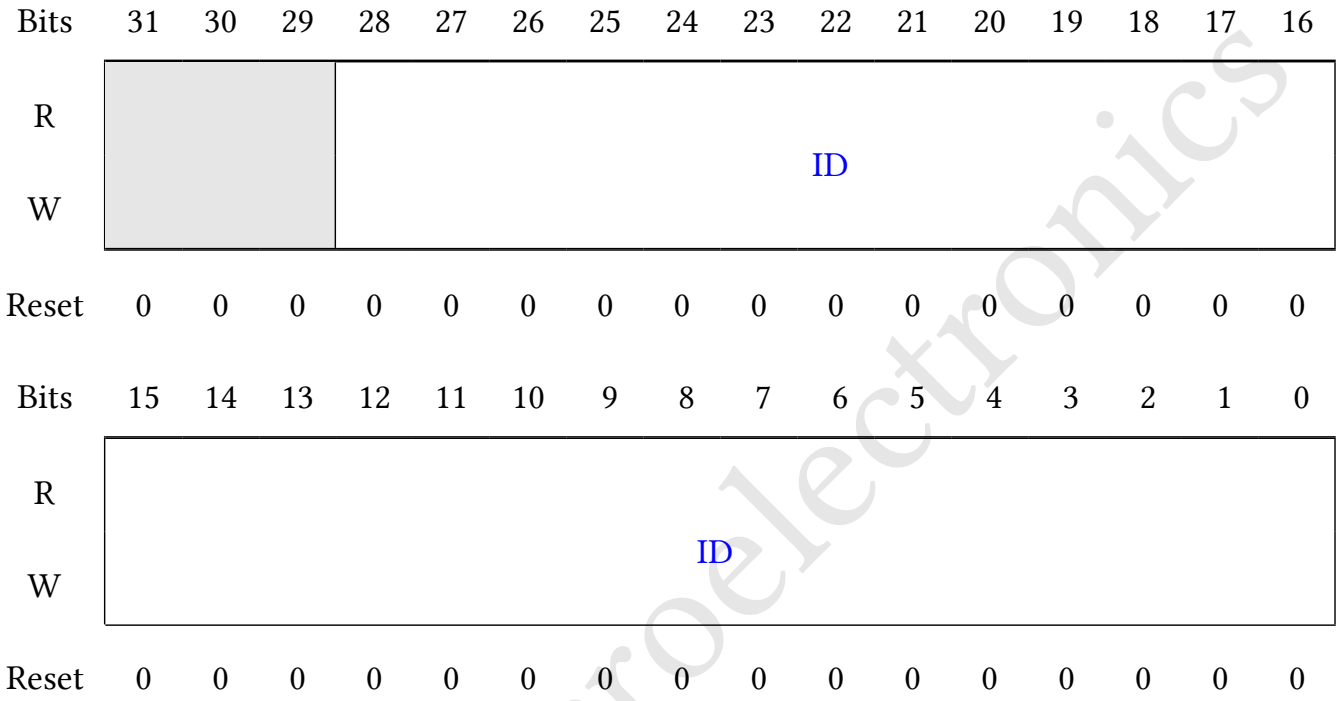


Table 20.55: CAN WMB_WMBn_ID Register Description

Field	Function
28 - 0 ID	Received ID under Pretended Networking Mode –Stores the received ID as either: • The 29 bits of the extended frame format (considering the ID[28:0] field), or • The 11 bits of the standard frame format (considering the ID[28:18] field only. The remaining bits in the ID[17:0] range have no meaning.

20.3.3.45 CAN WMB_WMBn_D03 Register

Each of the four WMBs contains a register to store the data bytes 0 to 3 of the payload information of an incoming Rx message. This register’s content is cleared when the incoming matched message is either a remote frame (RTR=1) or a data frame with DLC=0.
NOTE: The Data 0-3 registers are located at 0xB48 for WMB0, 0xB58 for WMB1, 0xB68 for WMB2, and 0xB78 for WMB3.
The CAN_WMBn_D03 registers are not available when parameter PNET_EN is 0.

Offset: b48h + n * 10h

Yuntu Microelectronics

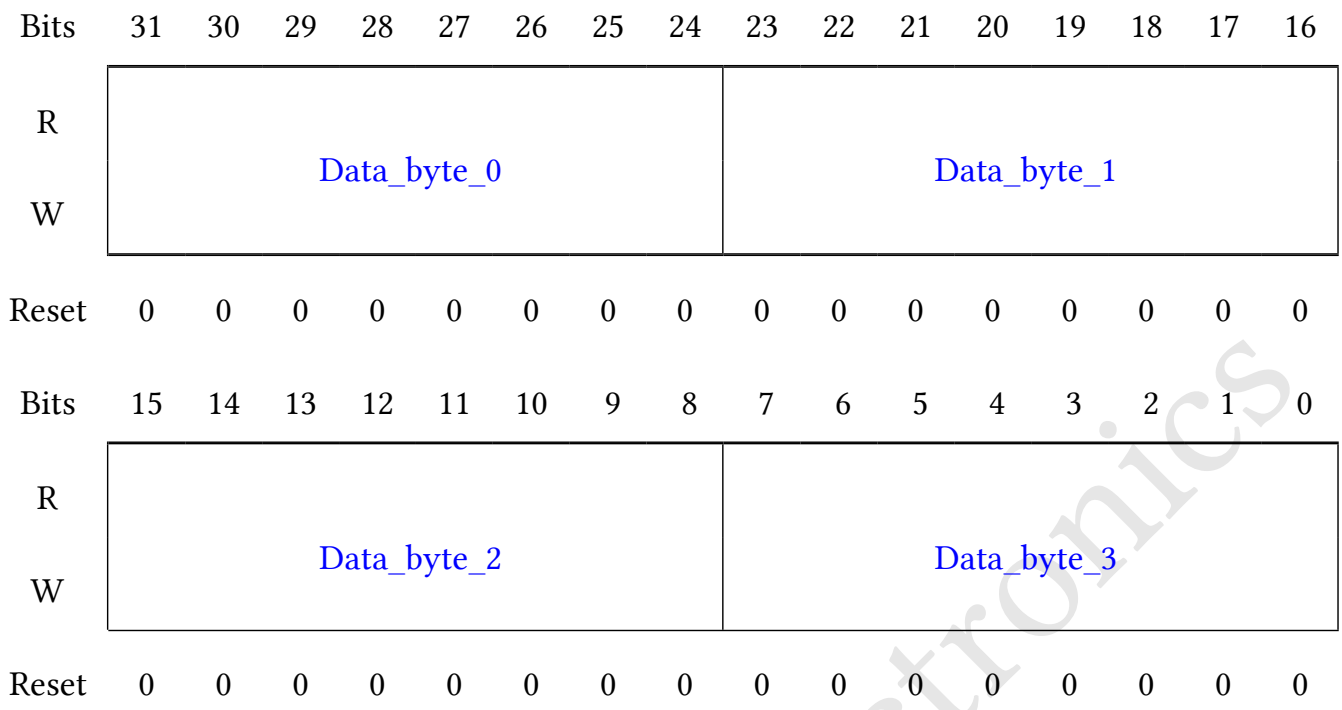


Table 20.56: CAN WMB_WMBn_D03 Register Description

Field	Function
31 - 24 Data_byte_0	Received payload corresponding to data byte 0 under Pretended Networking mode.
23 - 16 Data_byte_1	Received payload corresponding to data byte 1 under Pretended Networking mode.
15 - 8 Data_byte_2	Received payload corresponding to data byte 2 under Pretended Networking mode.
7 - 0 Data_byte_3	Received payload corresponding to data byte 3 under Pretended Networking mode.

20.3.3.46 CAN WMB_WMBn_D47 Register

Each of the four WMBs contains a register to store the data bytes 4 to 7 of the payload information of an incoming Rx message. This register's content is cleared when the incoming matched message is either a remote frame (RTR=1) or a data frame with DLC=0.
NOTE: The Data 4-7 registers are located at 0xB4C for WMB0, 0xB5C for WMB1, 0xB6C for WMB2, and

Yuntu Microelectronics

0xB7C for WMB3.
The CAN_WMBn_D47 registers are not available when parameter PNET_EN is 0.

Offset: b4ch + n * 10h

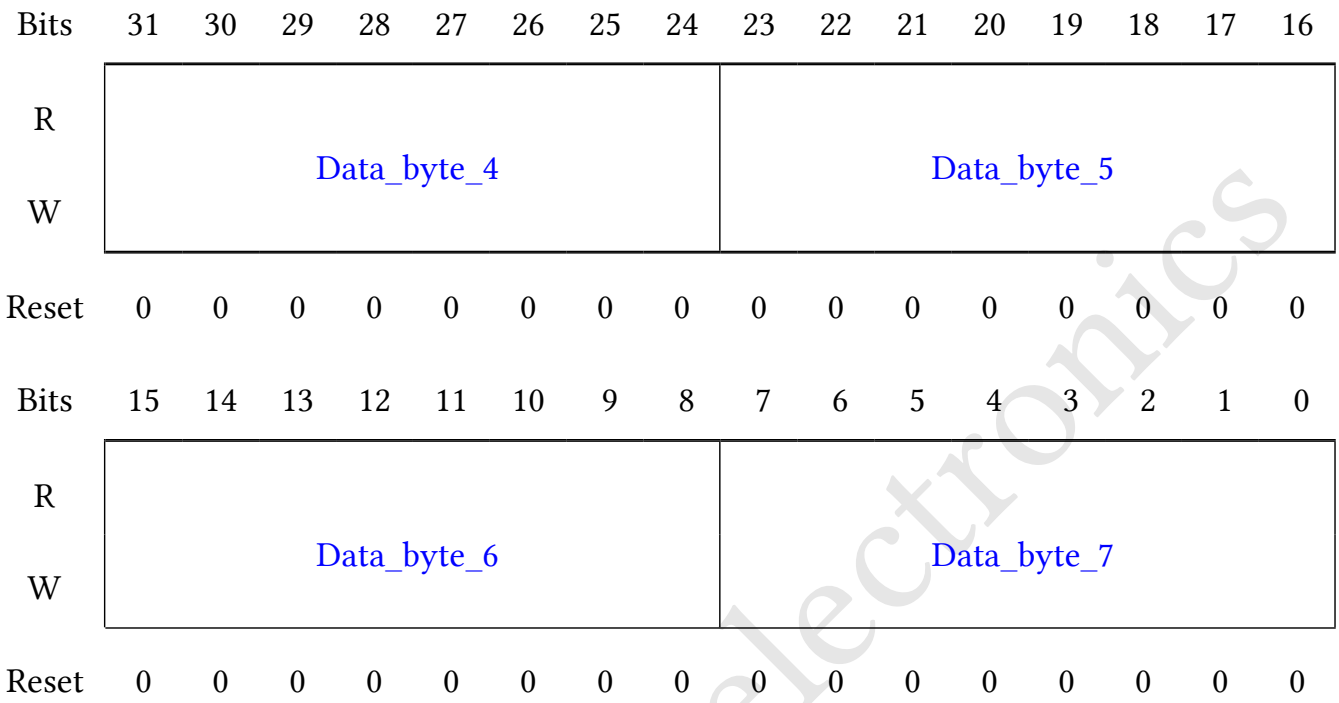


Table 20.57: CAN WMB_WMBn_D47 Register Description

Field	Function
31 - 24 Data_byte_4	Received payload corresponding to data byte 4 under Pretended Networking mode.
23 - 16 Data_byte_5	Received payload corresponding to data byte 5 under Pretended Networking mode.
15 - 8 Data_byte_6	Received payload corresponding to data byte 6 under Pretended Networking mode.
7 - 0 Data_byte_7	Received payload corresponding to data byte 7 under Pretended Networking mode.

Yuntu Microelectronics

20.3.3.47 CAN EPRS Register

The CAN_EPRS register defines the CAN bit timing prescalers for the nominal phase and data phase when CAN_CTRL2.BTE is set.
CAN_EPRS is used by the hardware only if CAN_CTRL2.BTE is set; otherwise a write operation has no effect and all fields are read as zero.
CAN_EPRS can be written only in Freeze mode because it is blocked by hardware in other modes.
The contents of CAN_EPRS are not affected by soft reset.
The CAN_EPRS register is not available when parameter CIA_BIT_TIMING is 0.

Offset: bf0h

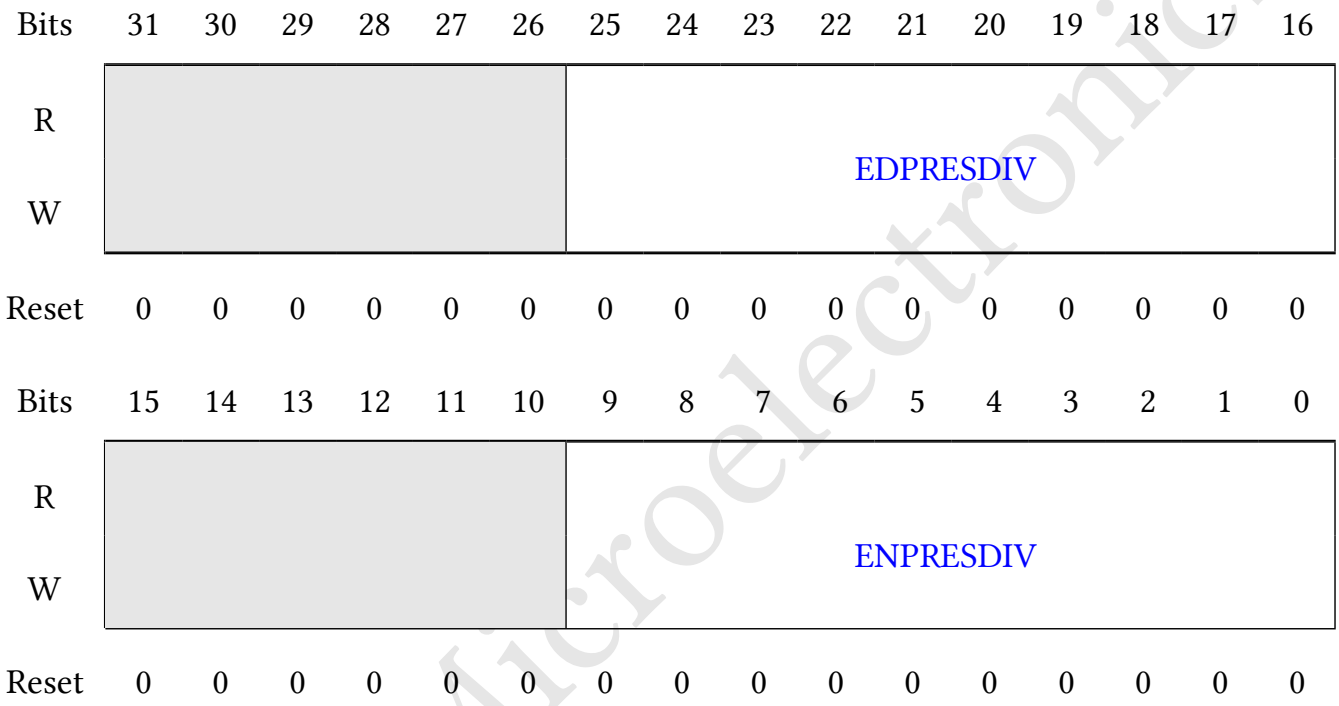


Table 20.58: CAN EPRS Register Description

Field	Function
25 - 16 EDPRESDIV	**Extended Data Phase Prescaler Division Factor** EDPRESDIV defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency in the data bit rate portion of a CAN FD message when CAN_CTRL2.BTE is set. The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. Sclock frequency = PE clock frequency / (EDPRESDIV + 1). NOTE: To minimize errors when processing FD frames, use the same value for ENPRESDIV and EDPRESDIV.
9 - 0	**Extended Nominal Phase Prescaler Division Factor**

Yuntu Microelectronics

Field	Function
ENPRESDIV	ENPRESDIV defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency for the nominal bit rate when CAN_CTRL2.BTE is set. The Sclock period defines the time quantum of the CAN protocol for the nominal phase. Sclock frequency = PE clock frequency / (ENPRESDIV + 1).

20.3.3.48 CAN ENCBT Register

The CAN_ENCBT register provides an alternative way to store the CAN bit timing variables described in the CAN_CTRL1 and CAN_CBT registers, to achieve higher CAN bit timing resolution.
CAN_ENCBT is used by the hardware only if CAN_CTRL2.BTE is set; otherwise a write operation has no effect and all fields are read as zero.
CAN_ENCBT can be written only in Freeze mode because it is blocked by hardware in other modes.
The contents of CAN_ENCBT are not affected by soft reset.
The CAN_ENCBT register is not available when parameter CIA_BIT_TIMING is 0.

Offset: bf4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R				NRJW										NTSEG2		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NTSEG2								NTSEG1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.59: CAN ENCBT Register Description

Field	Function
28 - 22	**Nominal Resynchronization Jump Width**

Yuntu Microelectronics

Field	Function
NRJW	NRJW defines the maximum number of time quanta that a nominal bit time can be changed by one resynchronization when CAN_CTRL2.BTE = 1. When CAN_CTRL2.BTE = 0, NRJW has no effect. One time quantum is equal to one Sclock period. Nominal Resynchronization Jump Width = NRJW + 1.
18 - 12 NTSEG2	**Nominal Time Segment 2** NTSEG2 defines the length of Time Segment 2 in the nominal bit time when CAN_CTRL2.BTE = 1. When CAN_CTRL2.BTE = 0, NTSEG2 has no effect. Nominal Time Segment 2 = (NTSEG2 + 1) × Time-Quanta Time-Quantum = one Sclock period
7 - 0 NTSEG1	**Nominal Time Segment 1** NTSEG1 defines the length of Time Segment 1 in the nominal bit time when CAN_CTRL2.BTE = 1. When CAN_CTRL2.BTE = 0, NTSEG1 has no effect. Nominal Time Segment 1 = (NTSEG1 + 1) × Time-Quanta Time-Quantum = one Sclock period

Yuntu Microelectronics

20.3.3.49 CAN EDCBT Register

The CAN_EDCBT register provides an alternative way to store the data phase CAN bit timing variables described in the CAN_FDCBT register, to achieve higher CAN bit timing resolution.
CAN_EDCBT is used by the hardware only if CAN_CTRL2.BTE is set; otherwise a write operation has no effect and all fields are read as zero.
CAN_EDCBT can be written only in Freeze mode because it is blocked by hardware in other modes.
The contents of CAN_EDCBT are not affected by soft reset.
The CAN_EDCBT register is not available when parameter CIA_BIT_TIMING is 0.
NOTE: CAN_EDCBT.DTSEG1 must be at least two time quanta.

Offset: bf8h

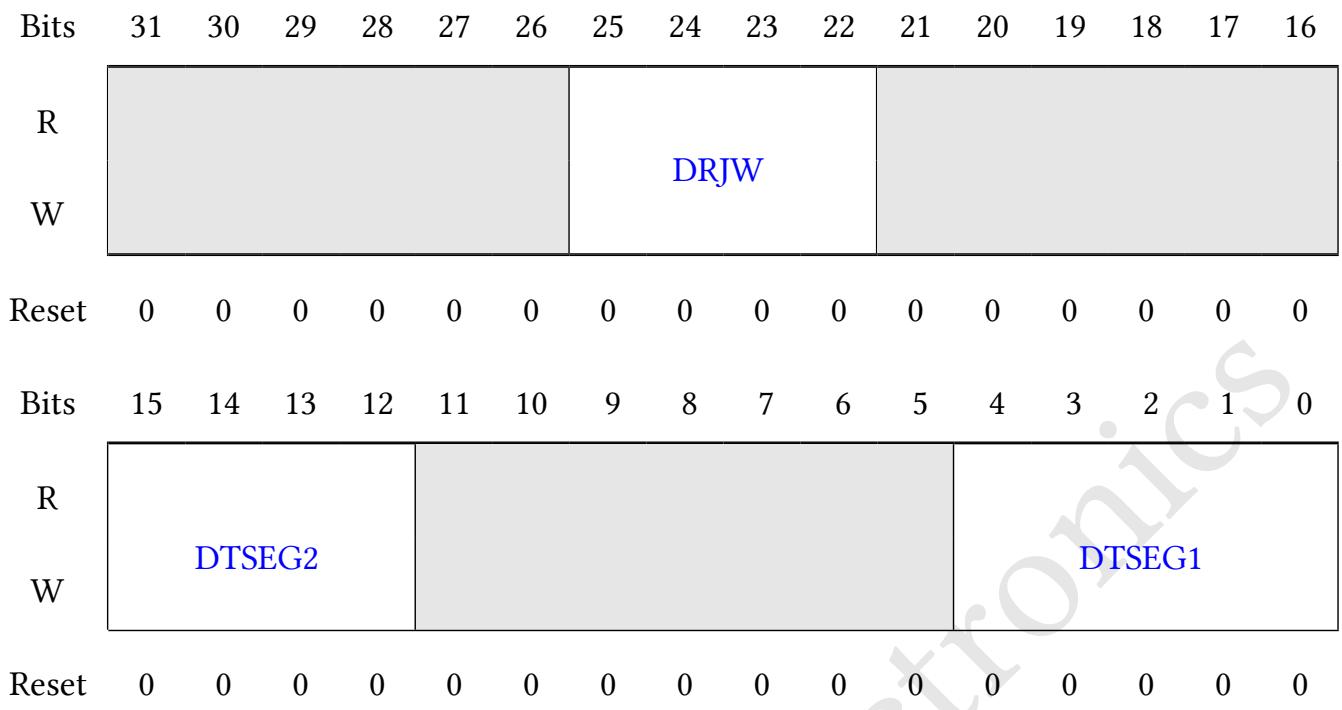


Table 20.60: CAN EDCBT Register Description

Field	Function
25 - 22 DRJW	**Data Phase Resynchronization Jump Width** DRJW defines the maximum number of time quanta that a data phase bit time can be changed by one resynchronization when CAN_CTRL2.BTE = 1. When CAN_CTRL2.BTE = 0, DRJW has no effect. One time quantum is equal to one Sclock period. Data Phase Resynchronization Jump Width = DRJW + 1.
15 - 12 DTSEG2	**Data Phase Time Segment 2** DTSEG2 defines the length of Time Segment 2 in the data phase bit time when CAN_CTRL2.BTE = 1. When CAN_CTRL2.BTE = 0, DTSEG2 has no effect. Data Phase Time Segment 2 = (DTSEG2 + 1) × Time-Quanta Time-Quantum = one Sclock period
4 - 0 DTSEG1	**Data Phase Time Segment 1** DTSEG1 defines the length of Time Segment 1 in the data phase bit time when CAN_CTRL2.BTE = 1. When CAN_CTRL2.BTE = 0, DTSEG1 has no effect. Data Phase Time Segment 1 = (DTSEG1 + 1) × Time-Quanta Time-Quantum = one Sclock period

Yuntu Microelectronics

20.3.3.50 CAN ETDC Register

The CAN_ETDC register contains extended versions of the fields CAN_FDCTRL.TDCOFF and CAN_FDCTRL.TDCVAL.
CAN_ETDC is used by the hardware only if CAN_CTRL2.BTE is set; otherwise a write operation has no effect and all fields are read as zero.
The CAN_ETDC register is not available when parameter CIA_BIT_TIMING is 0.
NOTE: See the transmitter delay compensation section of the CAN Protocol Standard (ISO 11898-1) for additional details.

Offset: bfch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ETDC EN	TDMD IS								ETDCOFF						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ETDC FAIL								ETDCVAL							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.61: CAN ETDC Register Description

Field	Function
31 ETDCEN	**Transceiver Delay Compensation Enable - Enables or disables the TDC feature.** ETDCEN can be written in Freeze mode only. NOTE: TDC must be disabled when Loop-Back mode is enabled (see CAN_CTRL1.LPB). 0b - TDC is disabled. 1b - TDC is enabled.
30	**Transceiver Delay Measurement Disable –Disables or enables transceiver delay measurement.**

Yuntu Microelectronics

Field	Function
TDMDIS	When TDC measurement is disabled, the secondary sample point position is determined only by the enhanced TDC offset (ETDCOFF) field. Otherwise, if TCD measurement is enabled, then the secondary sample point position is determined by the sum of the transceiver delay measurement plus the enhanced TDC offset. TDMDIS is not affected by soft reset. TDMDIS can be enabled only if CAN_CTRL2.BTE is set. 0b - TDC measurement is enabled. 1b - TDC measurement is disabled.
22 - 16 ETDCOFF	**Enhanced Transceiver Delay Compensation Offset** ETDCOFF contains the offset value to be added to the measured transceiver's loop delay in order to define the position of the delayed comparison point when bit rate switching is active. See "Transceiver Delay Compensation" on page 187 for additional details on how the loop delay measurement is performed. ETDCOFF can be written in Freeze mode only. Its value is defined in protocol engine (PE) clock periods and must be selected to be smaller than the CAN bit duration in the data phase bit rate for proper operation. ETDCOFF must not be configured as zero. NOTE: If CAN_CTRL2.BTE is set after a chip-level hard reset, ETCDOFF is read as 0x1.
15 ETDCFAIL	**Transceiver Delay Compensation Fail** ETDCFAIL indicates when the Transceiver Delay Compensation (TDC) mechanism is out of range, unable to compensate the transceiver's loop delay and successfully compare the delayed received bits to the transmitted ones ETDCFAIL is set the first time the FlexCAN detects the out of range condition. To clear ETDCFAIL, write 1 to it. See "Transceiver Delay Compensation". 0b - Measured loop delay is in range. 1b - Measured loop delay is out of range.
7 - 0 ETDCVAL	**Enhanced Transceiver Delay Compensation Value** ETDCVAL contains the ETDCOFF field added to the measured value of the transceiver loop delay in the latest transmitted CAN FD frame with BRS equal to recessive. ETDCVAL is only updated by the hardware if the ETDCEN bit is set. ETDCVAL is affected by soft reset. NOTE: If TDMDIS is set, ETDCVAL stores ETDCOFF only.

20.3.3.51 CAN FDCTRL Register

The CAN_FDCTRL register contains control bits for CAN FD operation. CAN_FDCTRL also defines the data size of message buffers allocated in different partitions of RAM (memory blocks).

When an 8-byte payload is selected:

- Block R0 allocates MB0 to MB31.
- Block R1 allocates MB32 to MB63.

Yuntu Microelectronics

- Block R2 allocates MB64 to MB95.
- Block R3 allocates MB96 to MB127.

When a payload larger than 8 bytes is selected, the maximum number of MBs in a block is limited as described in below table.

Table 20.62: CAN Number of Message Buffers

Payload Size	Maximum Number of Message Buffers per RAM Block
8 bytes	32
16 bytes	21
32 bytes	12
64 bytes	7

NOTE: One memory block fits exactly 32 MBs with 8-byte payload. For other payload sizes, empty memory may exist between the last MB in a block and the beginning of the next block. This empty memory corresponds to less than one MB, and must not be used.

The contents of CAN_FDCTRL are not affected by soft reset. The CAN_FDCTRL register is not available when parameter FD_EN is 0.

Offset: c00h

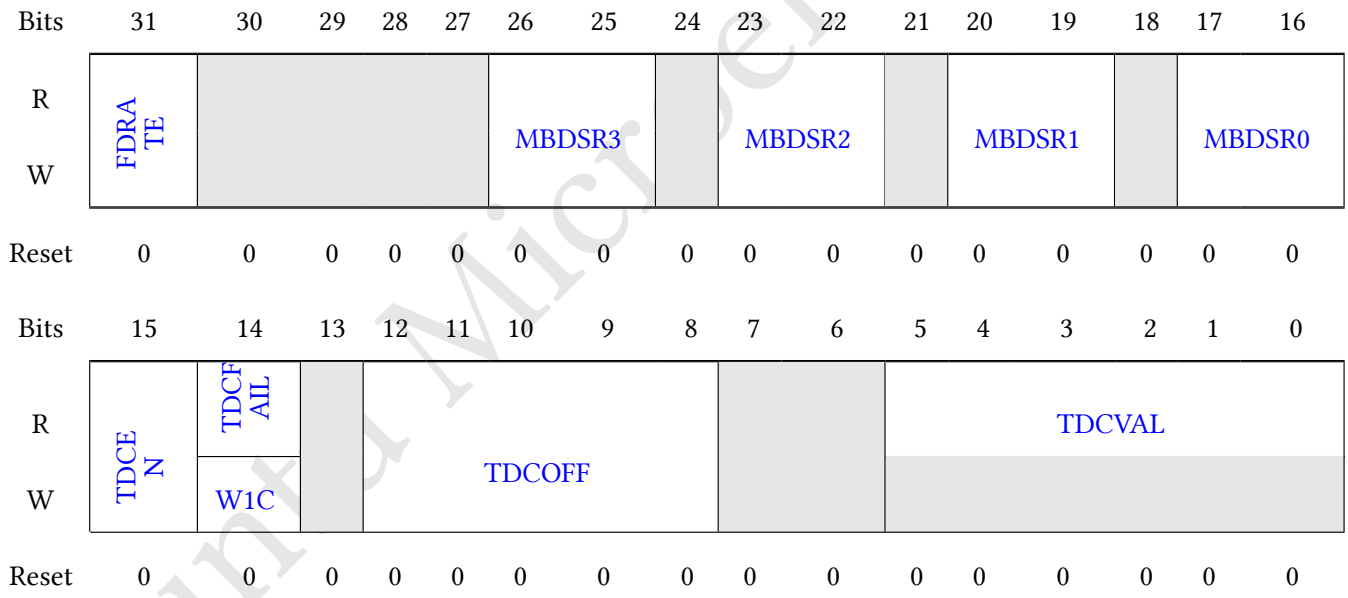


Table 20.63: CAN FDCTRL Register Description

Field	Function
31	**Bit Rate Switch Enable –Enables the effect of the Bit Rate Switch (BRS bit) during the data phase of Tx messages.**

Yuntu Microelectronics

Field	Function
FDRATE	The CPU can write the FDRATE bit any time. However, its effect becomes active only when the CAN bus is in the Wait for Bus Idle, Bus Idle, or Bus Off state, or when the current frame under reception or transmission reaches the interframe space. By negating FDRATE, the CPU can force all bits in CAN FD messages to be transmitted in nominal bit rate, regardless of the value of the BRS bit of the Tx MBs. 0b - Transmit a frame in nominal rate. The BRS bit in the Tx MB has no effect. 1b - Transmit a frame with bit rate switching if the BRS bit in the Tx MB is recessive.
26 - 25 MBDSR3	**Message Buffer Data Size for Region 3 –Selects the data size.** For region R3 of message buffers allocated in RAM: • 00: Selects 8 bytes per message buffer. • 01: Selects 16 bytes per message buffer. • 10: Selects 32 bytes per message buffer. • 11: Selects 64 bytes per message buffer. MBDSR3 can be written in Freeze mode only.
23 - 22 MBDSR2	**Message Buffer Data Size for Region 2 –Selects the data size.** For region R2 of message buffers allocated in RAM: • 00: Selects 8 bytes per message buffer. • 01: Selects 16 bytes per message buffer. • 10: Selects 32 bytes per message buffer. • 11: Selects 64 bytes per message buffer. MBDSR2 can be written in Freeze mode only.
20 - 19 MBDSR1	**Message Buffer Data Size for Region 1 –Selects the data size.** For region R1 of message buffers allocated in RAM: • 00: Selects 8 bytes per message buffer. • 01: Selects 16 bytes per message buffer. • 10: Selects 32 bytes per message buffer. • 11: Selects 64 bytes per message buffer. MBDSR1 can be written in Freeze mode only.
17 - 16 MBDSR0	**Message Buffer Data Size for Region 0 –Selects the data size.** For region R0 of message buffers allocated in RAM: • 00: Selects 8 bytes per message buffer. • 01: Selects 16 bytes per message buffer. • 10: Selects 32 bytes per message buffer. • 11: Selects 64 bytes per message buffer. MBDSR0 can be written in Freeze mode only.
15	**Transceiver Delay Compensation Enable –Enables or disables the TDC feature.**

Yuntu Microelectronics

Field	Function
TDCEN	TDCEN can be written in Freeze mode only. NOTE: TDC must be disabled when Loop-Back mode is enabled (see CAN_CTRL1.LPB). NOTE: If CAN_CTRL2.BTE =1, TDCEN is read as zero and a write operation has no effect. 0b - TDC is disabled. 1b - TDC is enabled.
14 TDCFAIL	**Transceiver Delay Compensation Fail** TDCFAIL indicates when the Transceiver Delay Compensation (TDC) mechanism is out of range, unable to compensate the transceiver’ s loop delay, and successfully compare the delayed received bits to the transmitted ones. TDCFAIL is set the first time the FlexCAN detects the out of range condition. To clear TDCFAIL, write 1 to it. See Transceiver Delay Compensation. NOTE: If CAN_CTRL2.BTE = 1, TDCFAIL is read as zero and a write operation has no effect. 0b - Measured loop delay is in range 1b - Measured loop delay is out of range
12 - 8 TDCOFF	**Transceiver Delay Compensation Offset** TDCOFF contains the offset value to be added to the measured transceiver’ s loop delay in order to define the position of the delayed comparison point when bit rate switching is active. See Transceiver Delay Compensation for more details on how the loop delay measurement is performed. TDCOFF can be written in Freeze mode only. Its value can be defined in PE clock periods and must be selected to be smaller than the CAN bit duration in the data bit rate for proper operation. NOTE: It is not recommended to use TDCOFF equal to zero. NOTE: If CAN_CTRL2.BTE = 1, TDCOFF is read as zero and a write operation has no effect.
5 - 0 TDCVAL	**Transceiver Delay Compensation Value** TDCVAL contains the value of the transceiver loop delay measured from the transmitted EDL to R0 transition edge to the respective received one added to the TDCOFF value. This value is an integer multiple of the PE clock period. See ”Protocol Timing” for more details on how the loop delay measurement is performed. NOTE: If CAN_CTRL2.BTE = 1, TDCVAL is read as zero.

Yuntu Microelectronics

20.3.3.52 CAN FDCBT Register

The CAN_FDCBT register stores the CAN bit timing variables used in the data phase of CAN FD messages when CAN_FDCTRL.FDRATE is set, compatible with CAN FD specification. FPRESDIV, FPROPSEG, FPSEG1, FPSEG2, and FRJW are used to define the time quantum duration, the number of time quanta per CAN bit, and the sample point position for the data bit rate portion of a CAN FD message with the BRS bit set.

The contents of CAN_FDCBT are not affected by soft reset.

NOTE: The sum of the Fast Propagation Segment (FPROPSEG) and Fast Phase Segment 1 (FPSEG1) must be at least two time quanta.

NOTE: If CAN_CTRL2.BTE = 1, the CAN_FDCBT register is read as zero and a write operation has no effect.

The CAN_FDCBT register is not available when parameter FD_EN is 0.

Offset: c04h

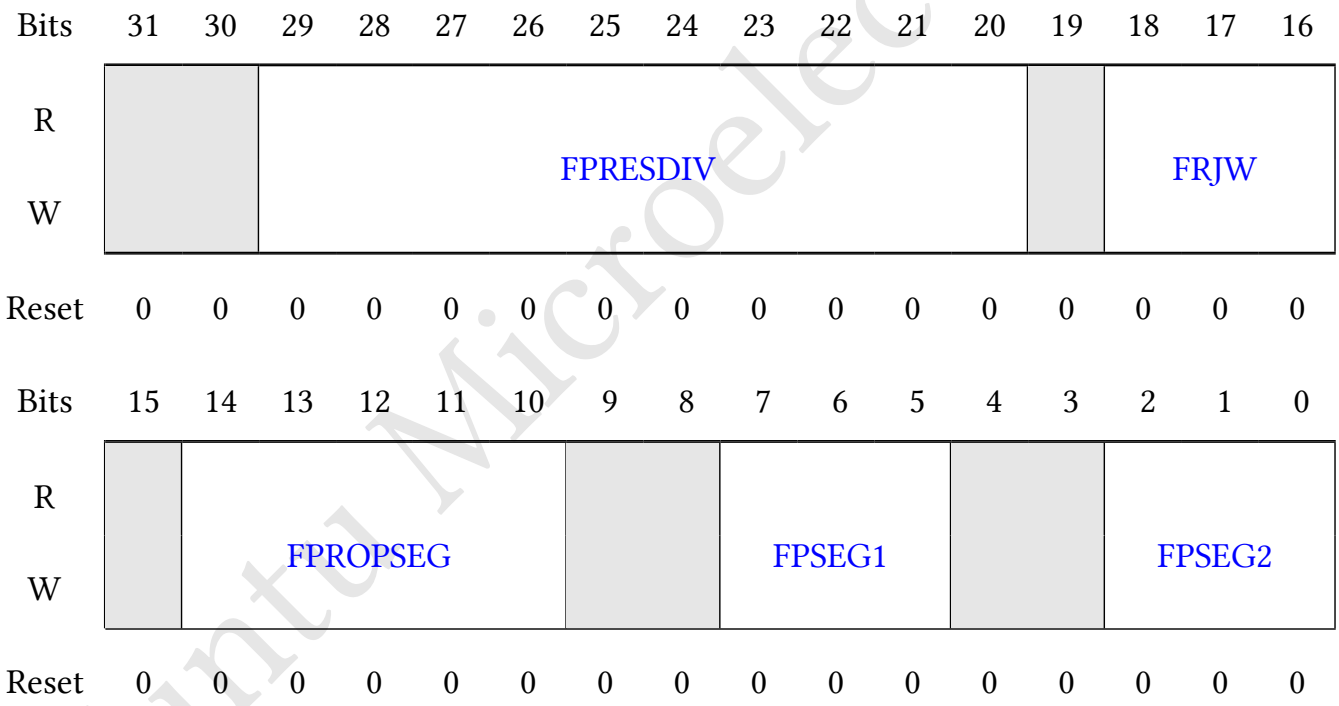


Table 20.64: CAN FDCBT Register Description

Field	Function
29 - 20	**Fast Prescaler Division Factor**

Yuntu Microelectronics

Field	Function
FPRESDIV	FPRESDIV defines the ratio between the PE clock frequency and the Serial Clock (Sclock) frequency in the data bit rate portion of a CAN FD message with the BRS bit set. The Sclock period defines the time quantum of the CAN FD protocol for the data bit rate. Sclock frequency = PE clock frequency / (FPRESDIV + 1). NOTE: To minimize errors when processing FD frames, use the same value for FPRESDIV and PRESDIV (in CAN_CBT or CAN_CTRL1). For more details, see "CAN FD Frames". FPRESDIV can be written only in Freeze mode because it is blocked by hardware in other modes
18 - 16 FRJW	**Fast Resync Jump Width** FRJW defines the maximum number of time quanta that a bit time can be changed by one re-synchronization in the data bit rate portion of a CAN FD message with the BRS bit set. Resync Jump Width = FSJW + 1. One time quantum is equal to the Sclock period. FRJW can be written only in Freeze mode because it is blocked by hardware in other modes.
14 - 10 FPROPSEG	**Fast Propagation Segment** FPROPSEG defines the length of the Propagation Segment in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. Propagation Segment Time = FPROPSEG × Time-Quanta Time-Quantum = one Sclock period FPROPSEG can be written only in Freeze mode because it is blocked by hardware in other modes.
7 - 5 FPSEG1	**Fast Phase Segment 1** FPSEG1 defines the length of Phase Segment 1 in the bit time in the data bit rate portion of a CAN FD message with the BRS bit set. Phase Segment 1 = (FPSEG1 + 1) × Time-Quanta Time-Quantum = one Sclock period FPSEG1 can be written only in Freeze mode because it is blocked by hardware in other modes.
2 - 0 FPSEG2	**Fast Phase Segment 2** FPSEG2 defines the length of Phase Segment 2 in the data bit rate portion of a CAN FD message with the BRS bit set. Phase Segment 2 = (FPSEG2 + 1) × Time-Quanta Time-Quantum = one Sclock period FPSEG2 can be written only in Freeze mode because it is blocked by hardware in other modes.

Yuntu Microelectronics

20.3.3.53 CAN FDCRC Register

The CAN_FDCRC register provides information about the CRC of transmitted messages.

FlexCAN uses different CRC polynomials for different frame formats:

- The CRC_15 polynomial is used for all frames in CAN format.
- The CRC_17 polynomial is used for frames in CAN FD format with a data field up to sixteen bytes.
- The CRC_21 polynomial is used for frames in CAN FD format with a data field longer than sixteen bytes.

Each polynomial shown below results in a Hamming Distance of 6. CAN_FDCRC is updated at the same time the Tx Interrupt Flag is asserted.

- CRC_15 = 0xC599: $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$
- CRC_17 = 0x3685B: $x^{17} + x^{16} + x^{14} + x^{13} + x^{11} + x^6 + x^4 + x^3 + x^1 + 1$
- CRC_21 = 0x302899: $x^{21} + x^{20} + x^{13} + x^{11} + x^7 + x^4 + x^3 + 1$

The CAN_FDCRC register is not available when parameter FD_EN is 0.

Offset: c08h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		FD_MBCRC										FD_TXCRC				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FD_TXCRC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.65: CAN FDCRC Register Description

Field	Function
30 - 24 FD_MBCRC	**CRC Mailbox Number for FD_TXCRC** FD_MBCRC indicates the number of the mailbox corresponding to the value in the FD_TXCRC field, for both FD and non-FD frames. FD_MBCRC reports the same information as in the CAN_CRCR.MBCRC bit field.

Yuntu Microelectronics

Field	Function
20 - 0 FD_TXCRC	**Extended Transmitted CRC value** FD_TXCRC contains the CRC value calculated over the most recent transmitted message. Different CRC polynomials are used for different frame formats. A 15-bit polynomial, CRC_15, is used for all frames in CAN format. The second 17-bit polynomial, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long. The third 21-bit polynomial, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes. For CRC_15 and CRC_17, the 6 most significant bits and the 4 most significant bits are reported as zeros, respectively. For CRC_15, this register has the same content as CAN_CRCR.

20.3.3.54 CAN ERFCR Register

The CAN_ERFCR register is not available when parameter ERX_FIFO_EN is 0. The CAN_ERFCR register is not available when parameter ERX_FIFO_EN is 0.

Table 20.66: CAN DMALW Value and Number of Enhanced RX FIFO Elements Transferred

DMALW	Number of 32-bit Words Transferred	Last FIFO Address
0	1	0x2000
1	2	0x2004
2	3	0x2008
3	4	0x200C
4	5	0x2010
5	6	0x2014
6	7	0x2018
7	8	0x201C
8	9	0x2020
9	10	0x2024
10	11	0x2028
11	12	0x202C
12	13	0x2030
13	14	0x2034
14	15	0x2038
15	16	0x203C
16	17	0x2040
17	18	0x2044
18	19	0x2048
19	20	0x204C

Yuntu Microelectronics

Table 20.67: CAN NEXIF Value and Number of Extended or Standard ID Filter Elements

NEXIF	NFE	Number of Extended ID Filter Elements	Number of Standard ID Filter Elements
0	63	0	128
1	63	1	126
2	63	2	124
3	63	3	122
4	63	4	120
5	63	5	118
6	63	6	116
7	63	7	114
8	63	8	112
9	63	9	110
10	63	10	108
11	63	11	106
12	63	12	104
13	63	13	102
14	63	14	100
15	63	15	98
16	63	16	96
17	63	17	94
18	63	18	92
19	63	19	90
20	63	20	88
21	63	21	86
22	63	22	84
23	63	23	82
24	63	24	80
25	63	25	78
26	63	26	76
27	63	27	74
28	63	28	72
29	63	29	70
30	63	30	68
31	63	31	66
32	63	32	64
33	63	33	62
34	63	34	60
35	63	35	58
36	63	36	56
37	63	37	54
38	63	38	52
39	63	39	50
40	63	40	48
41	63	41	46
42	63	42	44

Yuntu Microelectronics

Table 20.67 continued from previous page

NEXIF	NFE	Number of Extended ID Filter Elements	Number of Standard ID Filter Elements
43	63	43	42
44	63	44	40
45	63	45	38
46	63	46	36
47	63	47	34
48	63	48	32
49	63	49	30
50	63	50	28
51	63	51	26
52	63	52	24
53	63	53	22
54	63	54	20
55	63	55	18
56	63	56	16
57	63	57	14
58	63	58	12
59	63	59	10
60	63	60	8
61	63	61	6
62	63	62	4
63	63	63	2
64	63	64	0

Table 20.68: CAN NFE Value and Number of Enhanced RX FIFO Filter Elements

NFE	Maximum Number of Extended ID Filter Elements (NEXIF = NFE + 1)	Maximum Number of Standard ID Filter Elements (NEXIF = 0)
0	1	2
1	2	4
2	3	6
3	4	8
4	5	10
5	6	12
6	7	14
7	8	16
8	9	18
9	10	20
10	11	22
11	12	24
12	13	26
13	14	28
14	15	30
15	16	32
16	17	34

Yuntu Microelectronics

Table 20.68 continued from previous page

NFE	Maximum Number of Extended ID Filter Elements (NEXIF = NFE + 1)	Maximum Number of Standard ID Filter Elements (NEXIF = 0)
17	18	36
18	19	38
19	20	40
20	21	42
21	22	44
22	23	46
23	24	48
24	25	50
25	26	52
26	27	54
27	28	56
28	29	58
29	30	60
30	31	62
31	32	64
32	33	66
33	34	68
34	35	70
35	36	72
36	37	74
37	38	76
38	39	78
39	40	80
40	41	82
41	42	84
42	43	86
43	44	88
44	45	90
45	46	92
46	47	94
47	48	96
48	49	98
49	50	100
50	51	102
51	52	104
52	53	106
53	54	108
54	55	110
55	56	112
56	57	114
57	58	116
58	59	118
59	60	120

Yuntu Microelectronics

Table 20.68 continued from previous page

NFE	Maximum Number of Extended ID Filter Elements (NEXIF = NFE + 1)	Maximum Number of Standard ID Filter Elements (NEXIF = 0)
60	61	122
61	62	124
62	63	126
63	64	128

Offset: c0ch

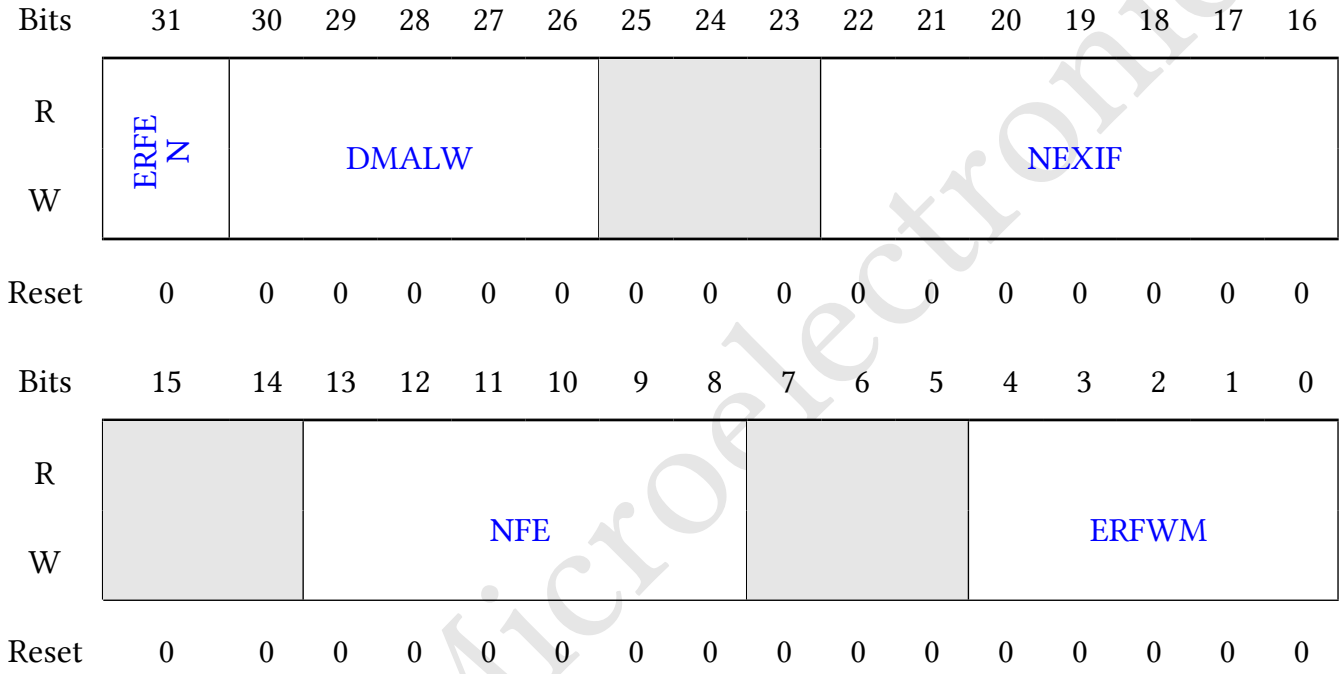


Table 20.69: CAN ERFCR Register Description

Field	Function
31 ERFEN	**Enhanced Rx FIFO Enable - Enables or disables the Enhanced Rx FIFO.** ERFEN can be written only in Freeze mode because it is blocked by hardware in other modes. NOTE: ERFEN must not be set if CAN_MCR.RFEN is set. 0b - Disable Enhanced Rx FIFO. 1b - Enable Enhanced Rx FIFO.
30 - 26	**DMA Last Word - Defines the last DMA address for each Enhanced Rx FIFO element.**

Yuntu Microelectronics

Field	Function
DMALW	Table ”CAN DMALW value and number of enhanced RX FIFO elements transferred” shows the number of elements and the last address for each Enhanced Rx FIFO element according to the value of DMALW. DMALW values not shown in Table ”CAN DMALW value and number of enhanced RX FIFO elements transferred” are reserved and must not be used. DMALW can be written only in Freeze mode because it is blocked by hardware in other modes.
22 - 16 NEXIF	**Number of Extended ID Filter Elements - Defines the number of extended ID filter elements used during the Enhanced Rx FIFO matching process.** NEXIF must be less than or equal to NFE + 1. The number of standard ID filter elements can be determined by the following equation: Number of Standard ID Filter Elements = 2 x (NFE - NEXIF +1) ”CAN NEXIF value and number of extended or standard ID filter elements” shows the number of extended ID filters and standard ID filters available for Enhanced Rx FIFO if all the filter elements are used. NEXIF can be written only in Freeze mode because it is blocked by hardware in other modes.
13 - 8 NFE	**Number of Enhanced Rx FIFO Filter Elements - Defines the total number of filter elements used during the Enhanced Rx FIFO matching process, as shown in Table 68.** NFE can be written only in Freeze mode because it is blocked by hardware in other modes.
4 - 0 ERFWM	**Enhanced Rx FIFO Watermark - Defines the minimum number of CAN messages stored in the Enhanced Rx FIFO.** Minimum number of CAN messages = ERFWM + 1. When ERFWM + 1 is reached, ERFSR.ERFWMI is set. ERFWM can be written only in Freeze mode because it is blocked by hardware in other modes.

Yuntu Microelectronics

20.3.3.55 CAN ERFIER Register

The CAN_ERFIER register is not available when parameter ERX_FIFO_EN is 0.

Offset: c10h

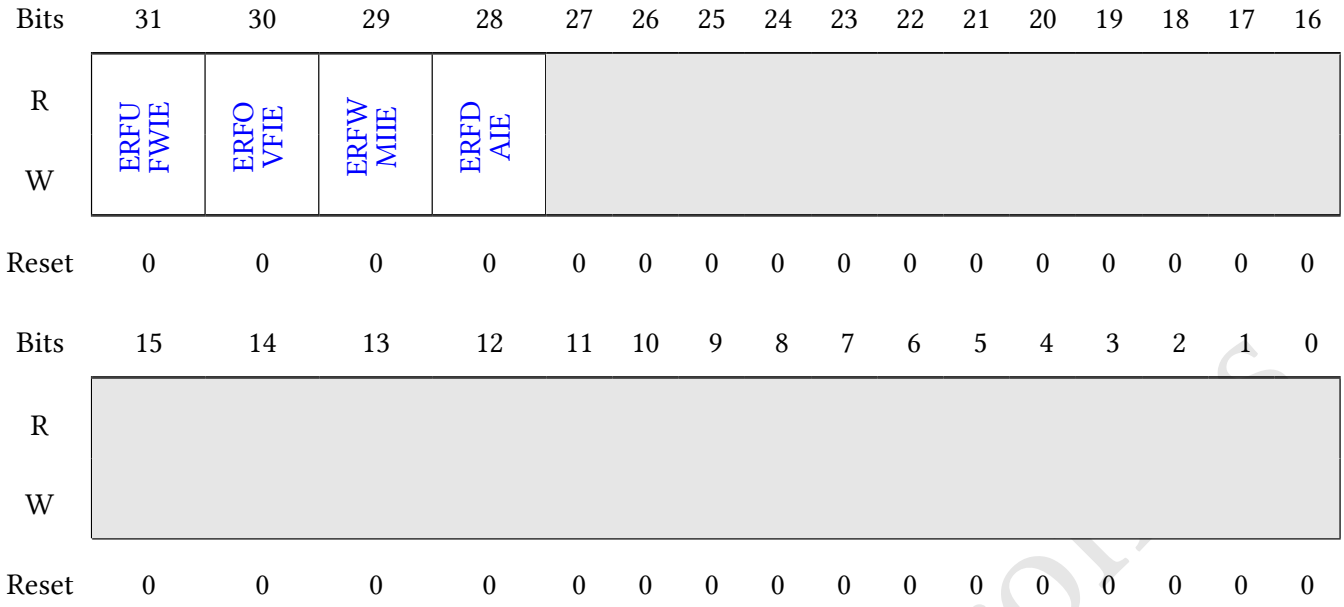


Table 20.70: CAN ERFIER Register Description

Field	Function
31 ERFUFWIE	**Enhanced Rx FIFO Underflow Interrupt Enable –Enables or disables the interrupt for Enhanced Rx FIFO underflow (ERFSR.ERFUFW).** 0b - Disable Enhanced Rx FIFO underflow interrupt. 1b - Enable Enhanced Rx FIFO underflow interrupt.
30 ERFOVFIE	**Enhanced Rx FIFO Overflow Interrupt Enable –Enables or disables the interrupt for Enhanced Rx FIFO overflow (ERFSR.ERFOVF).** 0b - Disable Enhanced Rx FIFO overflow interrupt. 1b - Enable Enhanced Rx FIFO overflow interrupt.
29 ERFWMIIE	**Enhanced Rx FIFO Watermark Indication Interrupt Enable –Enables or disables the interrupt for Enhanced Rx FIFO watermark (ERFSR.ERFWMI).** 0b - Disable Enhanced Rx FIFO watermark interrupt. 1b - Enable Enhanced Rx FIFO watermark interrupt.
28 ERFDAIE	**Enhanced Rx FIFO Data Available Interrupt Enable –Enables or disables the interrupt for Enhanced Rx FIFO data available (ERFSR.ERFDA).** 0b - Disable Enhanced Rx FIFO data available interrupt. 1b - Enable Enhanced Rx FIFO data available interrupt.

20.3.3.56 CAN ERFSR Register

The CAN_ERFSR register is not available when parameter ERX_FIFO_EN is 0.

Offset: c14h

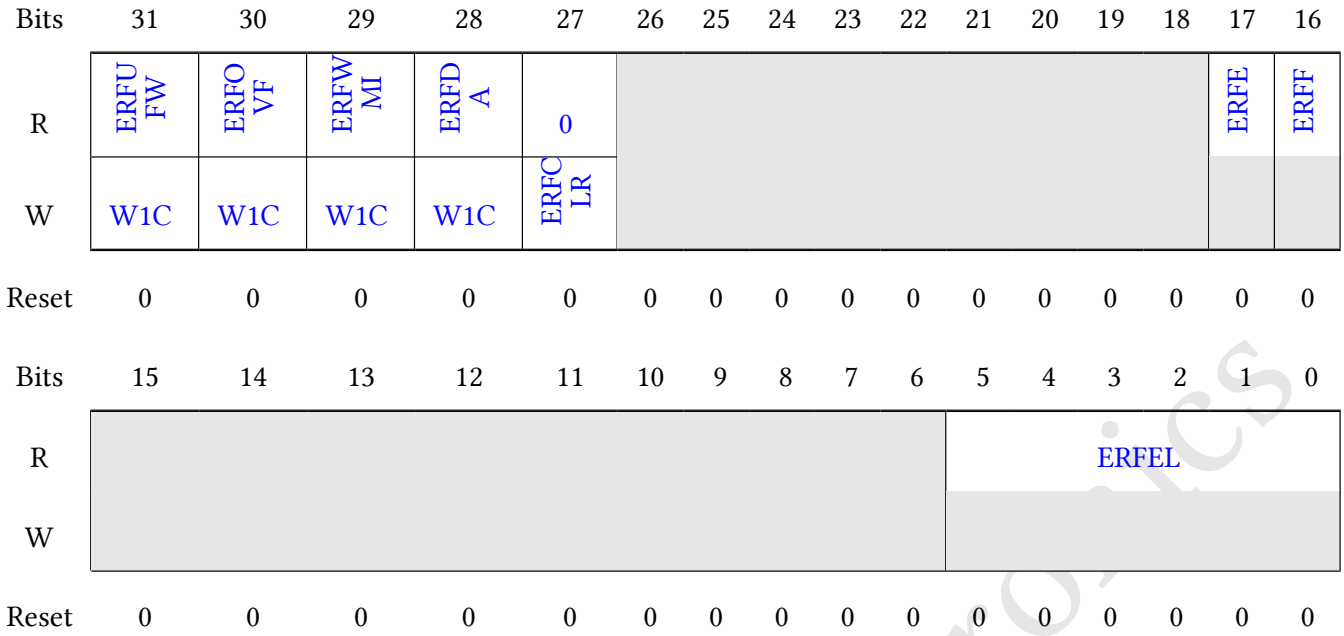


Table 20.71: CAN ERFSR Register Description

Field	Function
31 ERFUFW	**Enhanced Rx FIFO Underflow –Indicates that an underflow condition occurred in the Enhanced Rx FIFO.** ERFUFW is cleared by writing 1 to it. 0b - Enhanced Rx FIFO underflow has not occurred. 1b - Enhanced Rx FIFO underflow has occurred. Interrupt ipi_int_erfufw is asserted if CAN_ERFIER.ERFUFWIE = 1.
30 ERFOVF	**Enhanced Rx FIFO Overflow –Indicates that an overflow condition occurred in the Enhanced Rx FIFO.** ERFOVF is cleared by writing 1 to it. 0b - Enhanced Rx FIFO overflow has not occurred. 1b - Enhanced Rx FIFO overflow has occurred. Interrupt ipi_int_erfovf is asserted if CAN_ERFIER.ERFOVFIE = 1.
29 ERFWMI	**Enhanced Rx FIFO Watermark Indication –Set by hardware if the number of messages in the Enhanced Rx FIFO is greater than the watermark defined in CAN_ERFCR.ERFWM.** ERFWMI is cleared by writing 1 to it. 0b - The number of messages in the Enhanced Rx FIFO is not greater than the watermark. 1b - The number of messages in the Enhanced Rx FIFO is greater than the watermark. Interrupt ipi_int_erfwm is asserted if CAN_ERFIER.ERFWMIIE = 1.
28	**Enhanced Rx FIFO Data Available –Set by hardware if there is at least one message stored in the Enhanced Rx FIFO.**

Yuntu Microelectronics

Field	Function
ERFDA	ERFDA is cleared by writing 1 to it. 0b - There is no message stored in the Enhanced Rx FIFO. 1b - There is at least one message stored in the Enhanced Rx FIFO. Interrupt ipi_int_erfda is asserted if CAN_ERFIER.ERFDAIE = 1.
27 ERFCLR	**Enhanced Rx FIFO Clear** Writing 1 to ERFCLR when the FlexCAN is in Freeze mode clears the Enhanced Rx FIFO content. Writing 0 to ERFCLR or writing to ERFCLR when the FlexCAN is not in Freeze mode has no effect.
17 ERFE	**Enhanced Rx FIFO Empty** 0b - The Enhanced Rx FIFO is not empty. 1b - The Enhanced Rx FIFO is empty.
16 ERFF	**Enhanced Rx FIFO Full** 0b - The Enhanced Rx FIFO is not full. 1b - The Enhanced Rx FIFO is full.
5 - 0 ERFEL	**Enhanced Rx FIFO Elements –Indicates the number of CAN messages stored in the Enhanced Rx FIFO.**

20.3.3.57 CAN HR_TIME_STAMPn Register

Each High Resolution Time Stamp register (CAN_HR_TIME_STAMPn) stores a copy of the 32-bit timer at the start or end of a CAN frame for the associated message buffer.

CAN_HR_TIME_STAMP0 stores the 32-bit time stamp associated with MB0;

CAN_HR_TIME_STAMP1 stores the 32-bit time stamp associated with MB1; and so on.

The CAN_HR_TIME_STAMPn registers are not affected by reset and must be written only in Freeze mode.

The CAN_HR_TIME_STAMPn registers are not available if parameter TIME_STAMP_EN is 0.

Table 20.72: CAN High Resolution Time Stamp Register Operation

TSTAMPCAP	Captured Time Base	Capture Point
00	None	None
01	32 bits of high-resolution on-chip timer	Seventh bit of the end of frame field for transmission and sixth bit of the end of frame field for reception.
10	32 bits of high-resolution on-chip timer	Start of frame.
11	32 bits of high-resolution on-chip timer	Start of frame for Classical CAN frame format and res bit for CAN FD frame format.

Offset: c30h + n * 4h

Yuntu Microelectronics

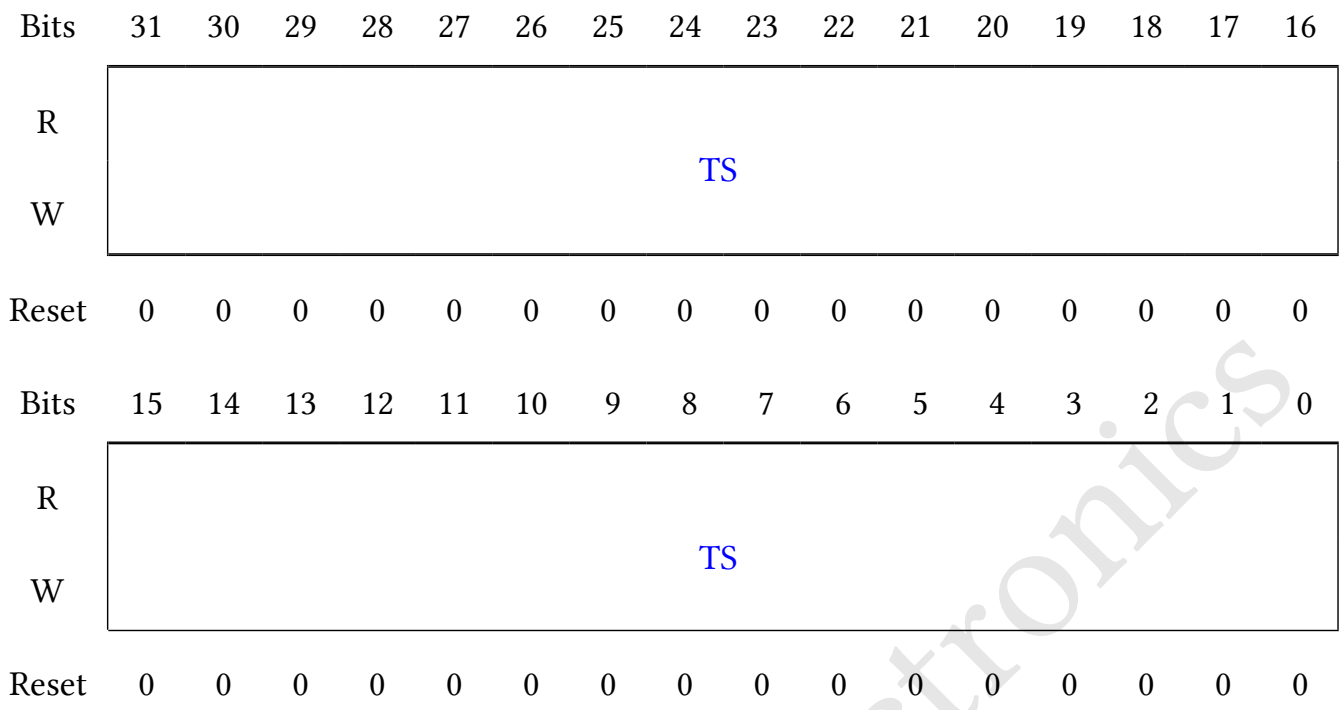


Table 20.73: CAN HR_TIME_STAMPn Register Description

Field	Function
31 - 0 TS	**High Resolution Time Stamp** Capture of 32-bit timer value for the corresponding message buffer. The capture point depends on the value of CAN_CTRL2.TSTAMPCAP, as shown in ”CAN high resolution time stamp register operation”.

Yuntu Microelectronics

20.3.3.58 CAN ERFFELn Register

The Enhanced Rx FIFO Filter Element Registers are used to store the filter elements of the Enhanced Rx FIFO.
For standard ID filtering, each CAN_ERFFELn register stores one filter element. For extended ID filtering, each pair of CAN_ERFFELn registers stores one filter element.
CAN_ERFFELn registers can only be accessed by the CPU when the FlexCAN is in Freeze mode; otherwise, they are blocked by hardware. The CAN_ERFFELn registers are not affected by reset. They are located in RAM and must be explicitly initialized prior to any reception.
The CAN_ERFFELn registers are not available if parameter ERX_FIFO_EN is 0.

Offset: 3000h + n * 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 20.74: CAN ERFFELn Register Description

Field	Function
31 - 0 FEL	**Filter Element Bits** Each filter element is used during the match process and, if the matching criteria are met, a message is stored in the Enhanced Rx FIFO.

20.3.4 Message Buffer Structure

Below figure shows the message buffer structure used by the FlexCAN module. Both Extended (29-bit identifier) and Standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual MB is formed by 16, 24, 40, or 72 bytes, depending on the quantity of data bytes allocated for the message payload: 8, 16, 32, or 64 data bytes, respectively.

The memory area from 0x80 to 0x87F is used by the mailboxes. When CAN FD is enabled, the exact address for each MB depends on the size of its payload. See “FlexCAN Memory Partition for CAN FD” for more detailed information.

When CAN FD is enabled, the memory area from 0x1000 to 0x17FF can be used for implementing up to 28 64-byte additional MBs. This additional MB region (enhanced MB memory) can be used partially or completely only if the total number of MBs configured by CAN_MCR.MAXMB is 128 or less (CAN_MCR.MAXMB = 127 or less). This means that the maximum number of MBs in the area from 0x80 to 0x87F and area from 0x1000 to 0x17FF is 128. See “FlexCAN Memory Partition for CAN FD” for more detailed information.

Yuntu Microelectronics

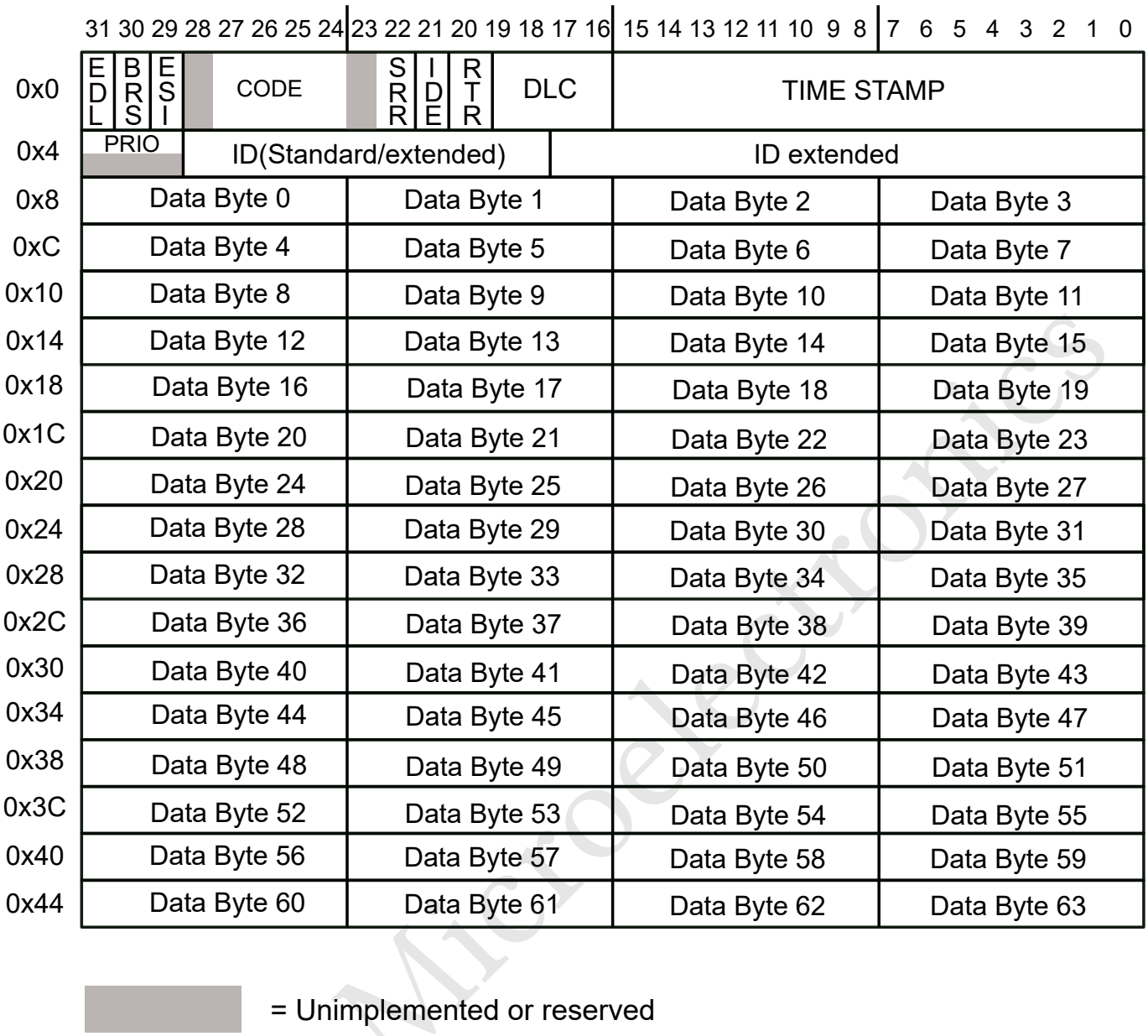


Figure 20.2: Message Buffer Structure Example with 64 Bytes Payload

EDL – Extended Data Length

- The EDL bit distinguishes between CAN format and CAN FD format frames. EDL must not be set for message buffers configured to RANSWER with code field 0b1010. See “CAN message buffer code for RX buffers”).

BRS – Bit Rate Switch

- The BRS bit defines whether the bit rate is switched inside a CAN FD format frame.

ESI – Error State Indicator

- The ESI bit indicates if the transmitting node is error active or error passive.

CODE – Message Buffer Code

- The CODE field can be accessed (read or write) by the CPU and by the FlexCAN module, as part of the message buffer matching and arbitration process. The encoding is shown in “CAN message

Yuntu Microelectronics

buffer code for RX buffers") and "CAN message buffer code for TX buffers"). See "Functional Description" for additional information.

Table 20.75: CAN Message Buffer Code for RX Buffers

CODE Description	Rx Code BEFORE Receive New Frame	SRV[1]	Rx Code AFTER Successful Reception[2]	RRS[3]	Comment
0000: INACTIVE - MB is not active	INACTIVE	–	–	–	MB does not participate in the matching process
0100: EMPTY - MB is active and empty	EMPTY	–	FULL	–	When a frame is received successfully (after the Move-in process), the CODE field is automatically updated to FULL.
0010: FULL - MB is full	FULL	Yes	FULL	–	The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a new frame is moved to the MB after the MB was serviced, the code still remains FULL. See "Matching Process" for matching details related to FULL code.
		No	OVERRUN	–	If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. See "Matching Process" for details about overrun behavior
0110: OVERRUN - MB is being overwritten into a full buffer.	OVERRUN	Yes	FULL	–	If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB, the code returns to FULL.
		No	OVERRUN	–	If the CODE field already indicates OVERRUN, and another new frame must be moved, the MB will be overwritten again, and the code will remain OVERRUN. See "Matching Process" for details about overrun behavior
1010: RANSWER[4] - A frame was configured to recognize a Remote Request Frame and transmit a Response Frame in return.[5]	RANSWER	–	TANSWER (1110)	0	A Remote Answer was configured to recognize a remote request frame received. After that, an MB is set to transmit a response frame. The code is automatically changed to TANSWER (1110). See "Matching Process" for details. If CAN_CTRL2.RRS is negated, transmit a response frame whenever a remote request frame with the same ID is received.

Yuntu Microelectronics

Table 20.75 continued from previous page

CODE Description	Rx Code BEFORE Receive New Frame	SRV[1]	Rx Code AFTER Successful Reception[2]	RRS[3]	Comment
		–	–	1	This code is ignored during matching and arbitration process. See "Matching Process" for details.
CODE[0]=1: BUSY - FlexCAN is updating the contents of the MB. The CPU must not access the MB	BUSY[6]	–	FULL	–	Indicates that the MB is being updated. It will be negated automatically and does not interfere with the next CODE.
		–	OVERRUN	–	

1. SRV: Serviced MB. MB was read and unlocked by reading CAN_TIMER or other MB.
2. A frame is considered a successful reception after the frame is moved to an MB(move-in process). See "Move-in" for details.
3. Remote Request Stored bit, see "Control 2 Register (CAN_CTRL2)" for details.
4. Code 1010 is not considered Tx and an MB with this code should not be aborted.
5. Code 1010 must be used in message buffers configured in CAN FD format, having the EDL bit set.
6. For Tx MBs, the BUSY bit should be ignored upon read, except when CAN_MCR.AEN is set. If this bit is asserted, the corresponding MB does not participate in the matching process.

Table 20.76: CAN Message Buffer Code for TX Buffers

CODE Description	Tx Code BEFORE Tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
1000: INACTIVE - MB is not active.	INACTIVE	–	–	MB does not participate in arbitration process
1001: ABORT - MB is aborted	ABORT	–	–	MB does not participate in arbitration process
1100: DATA - MB is a Tx Data Frame (MB RTR must be 0)	DATA	0	INACTIVE	Transmit data frame unconditionally once. After transmission, the MB automatically returns to the INACTIVE state.
1100: REMOTE – MB is a Tx Remote Request Frame (MB RTR must be 1).	REMOTE	1	EMPTY	Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID.

Yuntu Microelectronics

Table 20.76 continued from previous page

CODE Description	Tx Code BEFORE Tx frame	MB RTR	Tx Code AFTER successful transmission	Comment
1110: TANSWER – MB is a Tx Response Frame from an incoming Remote Request Frame.	TANSWER	–	RANSWER	This is an intermediate code that is automatically written to the MB by the CHI as a result of a match to a remote request frame. The remote response frame will be transmitted unconditionally once, and then the code will automatically return to RANSWER (1010). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. See ”Matching Process” and ”Arbitration Process” for details.

SRR – Substitute Remote Request

SRR is a fixed recessive bit, used only in extended format. SRR must be set to one by the user for transmission (Tx buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If the FlexCAN receives this bit as dominant, then it is interpreted as an arbitration loss.

- 1: Recessive value is compulsory for transmission in extended format frames.
- 0: Dominant is not a valid value for transmission in extended format frames.

IDE – ID Extended Bit

IDE identifies whether the frame format is standard or extended:

- 1: Frame format is extended.
- 0: Frame format is standard.

RTR – Remote Transmission Request

The RTR bit affects the behavior of remote frames and is part of the reception filter:

- 1: The current MB may have a remote request frame to be transmitted if MB is Tx. If the MB is Rx, then incoming remote request frames may be stored.
- 0: The current MB has a data frame to be transmitted. In an Rx MB, it may be considered in matching processes.

See “CAN message buffer code for RX buffers”) and ”CAN message buffer code for TX buffers”), and the description of the CAN_CTRL2.RRS bit for additional details.

If the FlexCAN transmits RTR as 1 (recessive) and receives it as 0 (dominant), it is interpreted as an arbitration loss. If RTR is transmitted as 0 (dominant) and received as 1 (recessive), the FlexCAN treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission.

NOTE: When configuring CAN FD frames, the RTR bit must be negated.

DLC – Length of Data in Bytes

Yuntu Microelectronics

The 4-bit DLC field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x8 through 0xF of the MB space(see: “message buffer structure example with 64 bytes payload”). In reception, DLC is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, DLC is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field(see ”CAN data bytes validity”).

TIME STAMP – Free-Running Counter Time Stamp

The 16-bit TIME STAMP field is a copy of the Free-Running Timer or 16 bits of the 32-bit onchip timer, captured for Tx and Rx frames as described in below Table.

Table 20.77: CAN Time Stamp Operation

CAN_CTRL2			Captured time base	Capture point
TSTAMPCAP	MBTSBASE	TIMER_SRC		
00	xx	0	CAN_TIMER incremented by CAN bit clock	Second bit of identifier field
00	xx	1	CAN_TIMER incremented by tmr_tick_ext	Second bit of identifier field
xx	00	0	CAN_TIMER incremented by CAN bit clock	Second bit of identifier field
xx	00	1	CAN_TIMER incremented by tmr_tick_ext	Second bit of identifier field
01	01	x	Lower 16 bits of high-resolution on-chip timer	Seventh bit of End of Frame field for transmission; sixth bit of End of Frame field for reception
01	10	x	Upper 16 bits of high-resolution on-chip timer	Seventh bit of End of Frame field for transmission; sixth bit of End of Frame field for reception
10	01	x	Lower 16 bits of high-resolution on-chip timer	Start of Frame
10	10	x	Upper 16 bits of high-resolution on-chip timer	Start of Frame
11	01	x	Lower 16 bits of high-resolution on-chip timer	Start of Frame for Classical CAN frame format; res bit for CAN FD frame format
11	10	x	Upper 16 bits of high-resolution on-chip timer	Start of Frame for Classical CAN frame format; res bit for CAN FD frame format

Yuntu Microelectronics

PRIO – Local Priority

The 3-bit PRIO field is used only when the CAN_MCR.LPRIO_EN bit is set and only makes sense for Tx mailboxes. The PRIO field is not transmitted. It is appended to the regular ID to define the transmission priority. See “Arbitration Process”.

ID – Frame Identifier

In standard frame format, only the 11 most significant bits (28 to 18) are used for frame identification for both receive and transmit, and the 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification for both receive and transmit.

DATA BYTE 0 to 63 – Data Field

Up to 64 bytes can be used for a data frame, depending on the size of payload selected for the Message Buffers.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (n) is valid only if n is less than DLC as shown in below Table: “CAN data bytes validity”).

Table 20.78: CAN Data Bytes Validity

DLC	Valid Data Bytes
0	None
1	DATA BYTE 0
2	DATA BYTE 0 to 1
3	DATA BYTE 0 to 2
4	DATA BYTE 0 to 3
5	DATA BYTE 0 to 4
6	DATA BYTE 0 to 5
7	DATA BYTE 0 to 6
8	DATA BYTE 0 to 7
9	DATA BYTE 0 to 11
10	DATA BYTE 0 to 15
11	DATA BYTE 0 to 19
12	DATA BYTE 0 to 23
13	DATA BYTE 0 to 31
14	DATA BYTE 0 to 47
15	DATA BYTE 0 to 63

20.3.5 FlexCAN Memory Partition for CAN FD

When CAN FD is enabled, the FlexCAN RAM can be partitioned in blocks of 512 bytes. Each block can accommodate a number of message buffers that depends on the configuration provided by CAN_FDCTRL.MBDSRn bit fields as shown in Table: “CAN RAM partition”.

Table 20.79: CAN RAM Partition

RAM Block	Number of MBS with8 Bytes (default range)	Size Control Bit Field in CAN_FDCTRL	Number of MBs of Different Sizes, per Block
0	0 to 31	MBDSR0	MBDSR0=00, 32 MBs with 8 bytes payload MBDSR0=01, 21 MBs with 16 bytes payload MBDSR0=10, 12 MBs with 32 bytes payload MBDSR0=11, 7 MBs with 64 bytes payload
1	32 to 63	MBDSR1	MBDSR1=00, 32 MBs with 8 bytes payload MBDSR1=01, 21 MBs with 16 bytes payload MBDSR1=10, 12 MBs with 32 bytes payload MBDSR1=11, 7 MBs with 64 bytes payload

Yuntu Microelectronics

Table 20.79 continued from previous page

RAM Block	Number of MBS with8 Bytes (default range)	Size Control Bit Field in CAN_FDCTRL	Number of MBs of Different Sizes, per Block
2	64 to 95	MBDSR2	MBDSR2=00, 32 MBs with 8 bytes payload MBDSR2=01, 21 MBs with 16 bytes payload MBDSR2=10, 12 MBs with 32 bytes payload MBDSR2=11, 7 MBs with 64 bytes payload
3	96 to 127	MBDSR3	MBDSR3=00, 32 MBs with 8 bytes payload MBDSR3=01, 21 MBs with 16 bytes payload MBDSR3=10, 12 MBs with 32 bytes payload MBDSR3=11, 7 MBs with 64 bytes payload
Blocks 4-7 exist when configuration parameter ENHANCE_MB_MEM =1			
4	–	–	7 MBs with 64 bytes payload (if the total number of MBs in blocks 0-3 is 121 or less)
5	–	–	7 MBs with 64 bytes payload (if the total number of MBs in blocks 0-3 is 114 or less)
6	–	–	7 MBs with 64 bytes payload (if the total number of MBs in blocks 0-3 is 107 or less)
7	–	–	7 MBs with 64 bytes payload (if the total number of MBs in blocks 0-3 is 100 or less)

When payload sizes of 16, 32, or 64 bytes are configured in some or all RAM blocks, the total number of MBs and the respective number order may differ from the default configuration of 8 bytes. For example, “CAN RAM partition example” indicates how the message buffers will be arranged into RAM when:

- Block 0 is configured to 8 bytes payload.
- Block 1 is configured to 16 bytes payload.
- Block 2 is configured to 32 bytes payload.
- Block 3 is configured to 64 bytes payload.
- CAN_MCR.MAXMB = 95 (96 message buffers are used).
- Configuration parameter ENHANCE_MB_MEM = 1.

Table 20.80: CAN RAM Partition Example

RAM block	Payload size	Number of MBs in the RAM block	Message Buffer range
0	CAN_FDCTRL[MBDSR0]=00, 8 bytes payload	32	0 to 31
1	CAN_FDCTRL[MBDSR1]=01, 16 bytes payload	21	32 to 52
2	CAN_FDCTRL[MBDSR2]=10, 32 bytes payload	12	53 to 64
3	CAN_FDCTRL[MBDSR2]=11, 64 bytes payload	7	65 to 71
4	64 bytes payload	7	72 to 78
5	64 bytes payload	7	79 to 85
6	64 bytes payload	7	86 to 92
7	64 bytes payload	3	93 to 95

Yuntu Microelectronics

20.3.6 FlexCAN Message Buffer Memory Map

The FlexCAN memory buffers are allocated in memory according to the following tables.

Table 20.81: CAN 8-byte Message Buffers

Address Offset (hex)	MBDSR = 00 (8-Byte Payload)
0080	MB0
0090	MB1
00A0	MB2
00B0	MB3
00C0	MB4
00D0	MB5
00E0	MB6
00F0	MB7
0100	MB8
0110	MB9
0120	MB10
0130	MB11
0140	MB12
0150	MB13
0160	MB14
0170	MB15
0180	MB16
0190	MB17
01A0	MB18
01B0	MB19
01C0	MB20
01D0	MB21
01E0	MB22
01F0	MB23
0200	MB24
0210	MB25
0220	MB26
0230	MB27
0240	MB28
0250	MB29
0260	MB30
0270	MB31
0280	MB32
0290	MB33
02A0	MB34
02B0	MB35
02C0	MB36
02D0	MB37
02E0	MB38
02F0	MB39
0300	MB40

Yuntu Microelectronics

Table 20.81 continued from previous page

Address Offset (hex)	MBDSR = 00 (8-Byte Payload)
0310	MB41
0320	MB42
0330	MB43
0340	MB44
0350	MB45
0360	MB46
0370	MB47
0380	MB48
0390	MB49
03A0	MB50
03B0	MB51
03C0	MB52
03D0	MB53
03E0	MB54
03F0	MB55
0400	MB56
0410	MB57
0420	MB58
0430	MB59
0440	MB60
0450	MB61
0460	MB62
0470	MB63
0480	MB64
0490	MB65
04A0	MB66
04B0	MB67
04C0	MB68
04D0	MB69
04E0	MB70
04F0	MB71
0500	MB72
0510	MB73
0520	MB74
0530	MB75
0540	MB76
0550	MB77
0560	MB78
0570	MB79
0580	MB80
0590	MB81
05A0	MB82
05B0	MB83
05C0	MB84
05D0	MB85

Yuntu Microelectronics

Table 20.81 continued from previous page

Address Offset (hex)	MBDSR = 00 (8-Byte Payload)
05E0	MB86
05F0	MB87
0600	MB88
0610	MB89
0620	MB90
0630	MB91
0640	MB92
0650	MB93
0660	MB94
0670	MB95
0680	MB96
0690	MB97
06A0	MB98
06B0	MB99
06C0	MB100
06D0	MB101
06E0	MB102
06F0	MB103
0700	MB104
0710	MB105
0720	MB106
0730	MB107
0740	MB108
0750	MB109
0760	MB110
0770	MB111
0780	MB112
0790	MB113
07A0	MB114
07B0	MB115
07C0	MB116
07D0	MB117
07E0	MB118
07F0	MB119
0800	MB120
0810	MB121
0820	MB122
0830	MB123
0840	MB124
0850	MB125
0860	MB126
0870	MB127

Yuntu Microelectronics

Table 20.82: CAN 16-byte Message Buffers

Address Offset (hex)	MBDSR = 01 (16-Byte Payload)
0080	MB0
0098	MB1
00B0	MB2
00C8	MB3
00E0	MB4
00F8	MB5
0110	MB6
0128	MB7
0140	MB8
0158	MB9
0170	MB10
0188	MB11
01A0	MB12
01B8	MB13
01D0	MB14
01E8	MB15
0200	MB16
0218	MB17
0230	MB18
0248	MB19
0260	MB20
0280	MB21
0298	MB22
02B0	MB23
02C8	MB24
02E0	MB25
02F8	MB26
0310	MB27
0328	MB28
0340	MB29
0358	MB30
0370	MB31
0388	MB32
03A0	MB33
03B8	MB34
03D0	MB35
03E8	MB36
0400	MB37
0418	MB38
0430	MB39
0448	MB40
0460	MB41
0480	MB42
0498	MB43

Yuntu Microelectronics

Table 20.82 continued from previous page

Address Offset (hex)	MBDSR = 01 (16-Byte Payload)
04B0	MB44
04C8	MB45
04E0	MB46
04F8	MB47
0510	MB48
0528	MB49
0540	MB50
0558	MB51
0570	MB52
0588	MB53
05A0	MB54
05B8	MB55
05D0	MB56
05E8	MB57
0600	MB58
0618	MB59
0630	MB60
0648	MB61
0660	MB62
0680	MB63
0698	MB64
06B0	MB65
06C8	MB66
06E0	MB67
06F8	MB68
0710	MB69
0728	MB70
0740	MB71
0758	MB72
0770	MB73
0788	MB74
07A0	MB75
07B8	MB76
07D0	MB77
07E8	MB78
0800	MB79
0818	MB80
0830	MB81
0848	MB82
0860	MB83

Yuntu Microelectronics

Table 20.83: CAN 32-byte Message Buffers

Address Offset (hex)	MBDSR = 10 (32-Byte Payload)
0080	MB0
00A8	MB1
00D0	MB2
00F8	MB3
0120	MB4
0148	MB5
0170	MB6
0198	MB7
01C0	MB8
01E8	MB9
0210	MB10
0238	MB11
0280	MB12
02A8	MB13
02D0	MB14
02F8	MB15
0320	MB16
0348	MB17
0370	MB18
0398	MB19
03C0	MB20
03E8	MB21
0410	MB22
0438	MB23
0480	MB24
04A8	MB25
04D0	MB26
04F8	MB27
0520	MB28
0548	MB29
0570	MB30
0598	MB31
05C0	MB32
05E8	MB33
0610	MB34
0638	MB35
0680	MB36
06A8	MB37
06D0	MB38
06F8	MB39
0720	MB40
0748	MB41
0770	MB42
0798	MB43

Yuntu Microelectronics

Table 20.83 continued from previous page

Address Offset (hex)	MBDSR = 10 (32-Byte Payload)
07C0	MB44
07E8	MB45
0810	MB46
0838	MB47

Table 20.84: CAN 64-byte Message Buffers

Address Offset (hex)	MBDSR = 11 (64-Byte Payload)
0080	MB0
00C8	MB1
0110	MB2
0158	MB3
01A0	MB4
01E8	MB5
0230	MB6
0280	MB7
02C8	MB8
0310	MB9
0358	MB10
03A0	MB11
03E8	MB12
0430	MB13
0480	MB14
04C8	MB15
0510	MB16
0558	MB17
05A0	MB18
05E8	MB19
0630	MB20
0680	MB21
06C8	MB22
0710	MB23
0758	MB24
07A0	MB25
07E8	MB26
0830	MB27

Yuntu Microelectronics

20.3.7 Legacy Rx FIFO Structure

When the CAN_MCR.RFEN bit is set, the memory area from 0x80 to 0xDC (which is normally occupied by MBs 0–5) is used by the Legacy Rx FIFO engine.

The region 0x80–0x8C contains the output of the Legacy Rx FIFO, which must be read by the CPU as a message buffer. The Legacy Rx FIFO output contains the oldest message that has been received but not yet read. The region 0x90–0xDC is reserved for internal use of the Legacy Rx FIFO engine.

An additional memory area, which starts at 0xE0 and may extend up to 0x2DC (normally occupied by MBs 6–37) depending on the CAN_CTRL2.RFFN field setting, contains the ID filter table (configurable from 8 to 128 table elements), which specifies filtering criteria for accepting frames into the Legacy Rx FIFO.

Out of reset, the ID filter table flexible memory area defaults to 0xE0 and extends only to 0xFC, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of the FlexCAN.

below table shows the Legacy Rx FIFO data structure.

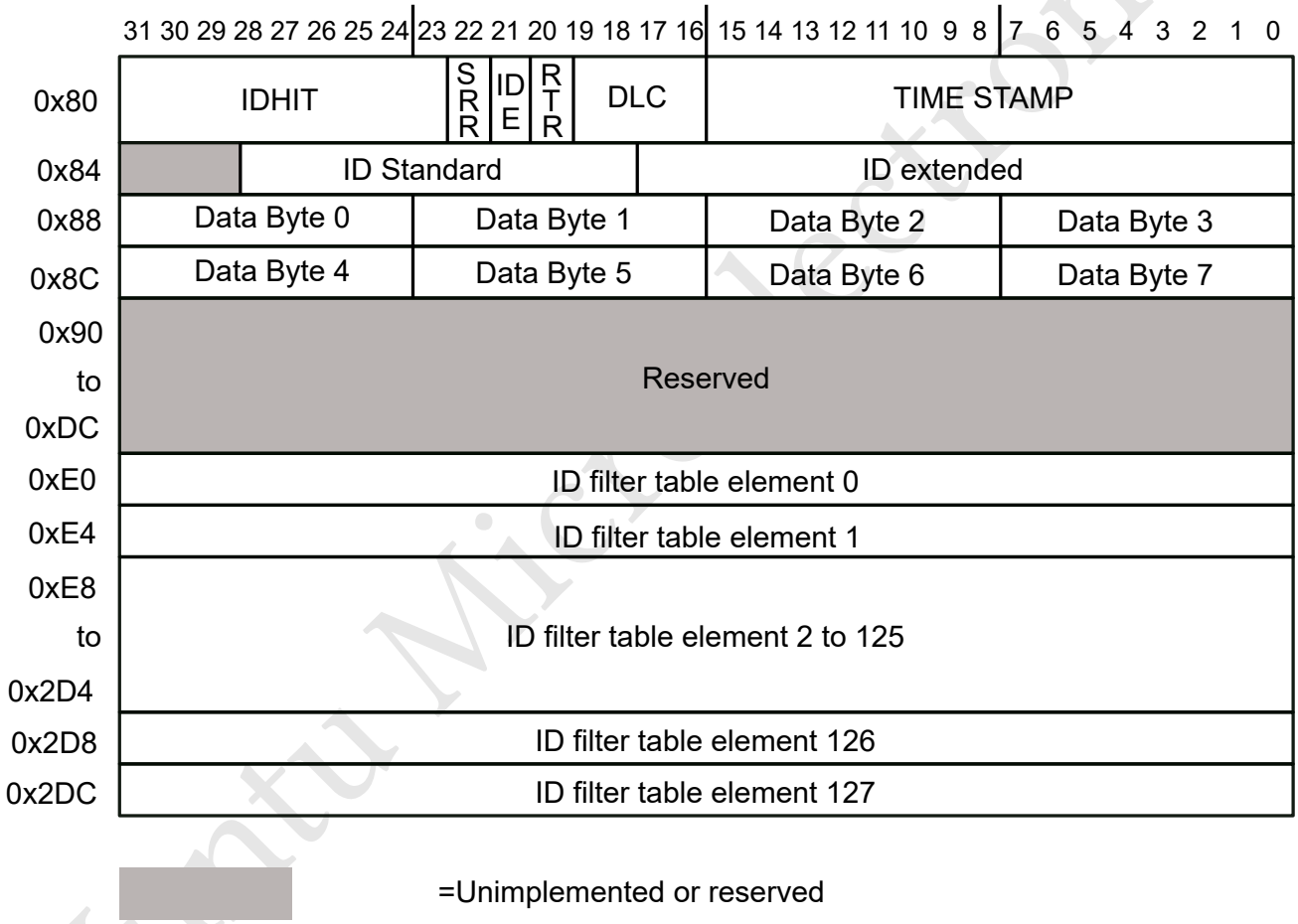


Figure 20.3: Legacy RX FIFO Structure

Each ID filter table element occupies an entire 32-bit word and can be compounded by one, two, or four Identifier Acceptance Filters (IDAF) depending on the CAN_MCR.IDAM field setting.

Below table shows the three different formats of the ID table elements. All elements of the table must have the same format, as selected by CAN_MCR.IDAM. See "Legacy Rx FIFO" for more information.

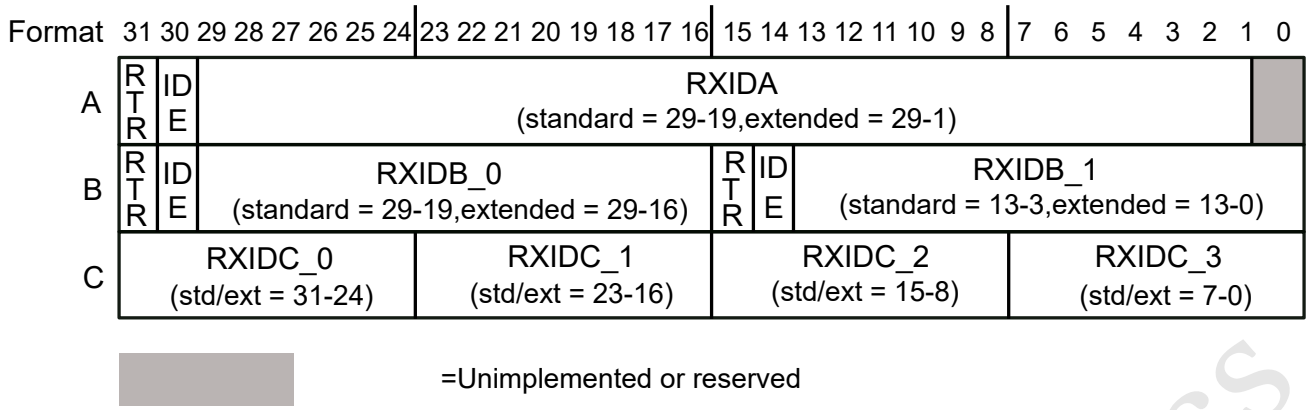


Figure 20.4: ID Filter Table Structure

RTR – Remote Frame

The RTR bit specifies if remote frames are accepted into the Legacy Rx FIFO if they match the target ID:

- 1 = Remote frames can be accepted and data frames are rejected.
- 0 = Remote frames are rejected and data frames can be accepted. IDE – Extended Frame

IDE – Extended Frame

The IDE bit specifies whether extended or standard frames are accepted into the Legacy Rx FIFO if they match the target ID:

- 1 = Extended frames can be accepted and standard frames are rejected.
- 0 = Extended frames are rejected and standard frames can be accepted.

RXIDA – Rx Frame Identifier (Format A)

Specifies an ID to be used as acceptance criteria for the Legacy Rx FIFO. In the standard frame format, only the 11 most significant bits (29 to 19) are used for frame identification. In the extended frame format, all bits are used.

RXIDB_0, RXIDB_1 – Rx Frame Identifier (Format B)

Specifies an ID to be used as acceptance criteria for the Legacy Rx FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (29 to 19 and 13 to 3) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

RXIDC_0, RXIDC_1, RXIDC_2, RXIDC_3 – Rx Frame Identifier (Format C)

Specifies an ID to be used as acceptance criteria for the Legacy Rx FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

IDHIT – Identifier Acceptance Filter Hit Indicator

The 9-bit IDHIT field indicates which Identifier Acceptance Filter was hit by the received message that is in the output of the Legacy Rx FIFO. See “Legacy Rx FIFO” for more information.

Yuntu Microelectronics

20.3.8 Enhanced Rx FIFO Structure

When CAN_ERFCR.ERFEN is set, the Enhanced Rx FIFO is enabled. The region 0x2000-0x204C contains the output of the Enhanced Rx FIFO which must be read by the CPU as a message buffer. This output contains the oldest message that has been received but not yet read. "enhanced rx fifo structure" shows the Enhanced Rx FIFO data structure.

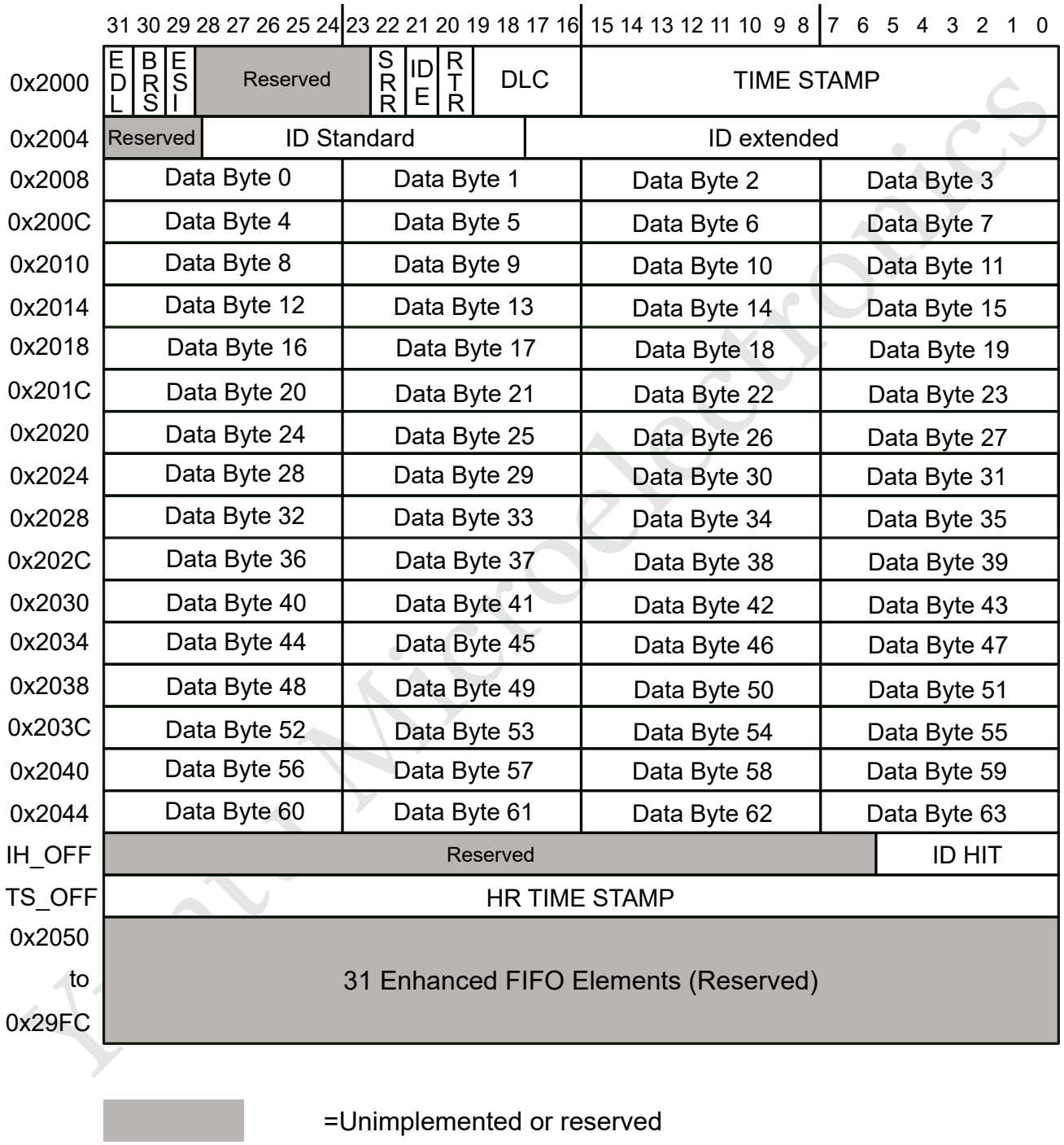


Figure 20.5: Enhanced RX FIFO Structure

NOTE: ID HIT offset and HR TIME STAMP offset are changed dynamically according to Data Length Code (DLC), as shown in below Table.

Table 20.85: CAN ID HIT and HR Time Stamp Offsets

Data Length Code (DLC)	ID HIT Offset (IH_OFF)	HR TIME STAMP Offset (TS_OFF)
0	0x2008	0x200C
1-4	0x200C	0x2010
5-8	0x2010	0x2014
9	0x2014	0x2018
10	0x2018	0x201C
11	0x201C	0x2020
12	0x2020	0x2024
13	0x2028	0x202C
14	0x2038	0x203C
15	0x2048	0x204C

EDL – Extended Data Length

The EDL bit distinguishes between classic CAN format and CAN FD format frames:

- 1 = CAN FD frame format.
- 0 = Classic CAN frame format.

BRS – Bit Rate Switch

The BRS bit indicates whether the bit rate is switched inside a CAN FD format frame:

- 1 = Bit rate is switched in a CAN FD frame.
- 0 = Bit rate is not switched in a CAN FD frame.

ESI – Error State Indicator

The ESI bit indicates if the transmitting node is error active or error passive:

- 1 = Transmitting node is error passive.
- 0 = Transmitting node is error active.

ESI is meaningful only if EDL = 1.

SRR – Substitute Remote Request

SRR is a fixed recessive bit, used only in extended format. Transmitting nodes always send it as recessive and receiving nodes can receive it as either recessive or dominant. If the FlexCAN receives the SRR bit as dominant, then it is interpreted as an arbitration loss.

IDE – ID Extended Bit

The IDE bit indicates whether the frame format is standard or extended:

- 1 = Frame format is extended.
- 0 = Frame format is standard.

RTR – Remote Frame

The RTR bit indicates whether the frame format is a data frame or remote request:

- 1 = Frame is remote request.
- 0 = Frame is a data frame.

Yuntu Microelectronics

DLC – Data Length Code

The 4-bit DLC field defines the number of bytes in the data field of a CAN frame. When RTR = 1, the frame is a remote request and does not include the data field, regardless of the DLC field.

TIME STAMP – 16-bit Time Stamp

The 16-bit TIME STAMP field is a copy of the Free-Running Timer, captured during the CAN frame.

ID – Frame Identifier

In the standard frame format, only the 11 most significant bits are used for frame identification. The 18 least significant bits are ignored. In the extended frame format, all 29 bits are used for frame identification.

DATA BYTE 0 to 63 – Data Field

Up to sixty four bytes can be stored in the data field.

ID HIT – Identifier Acceptance Filter Hit Indicator

The ID HIT field indicates which Enhanced Rx FIFO Filter Element (CAN_ERFFELn) was hit by the received message that is in the output of the Enhanced Rx FIFO. For each filter region, standard ID filter space and extended ID filter space, there is an independent index starting from zero. below Table shows how ID HIT is written by the FlexCAN according to each filter element.

Table 20.86: CAN ID HIT for Enhanced RX FIFO

Enhanced Rx FIFO Filter Element (CAN_ERFFEL)	ID HIT Value	Filter Element Type
CAN_ERFFEL0	0	Extended ID
CAN_ERFFEL1	1	Extended ID
...		Extended ID
CAN_ERFFEL(m-1)	m-1	Extended ID
CAN_ERFFEL(m)	0	Standard ID
CAN_ERFFEL(m+1)	1	Standard ID
...		Standard ID
CAN_ERFFEL(2n-m+1)	2x(n-m)+1	Standard ID

NOTE: m = NEXIF and n = NFE. If NEXIF = 0, then only standard ID filter elements exist. If NEXIF > NFE, then only extended ID filter elements exist.

HR TIME STAMP – High-Resolution Time Stamp

32-bit time base captured during the CAN frame. When CAN_CTRL2.TSTAMPCAP is nonzero, a 32-bit time base is captured from a dedicated on-chip timer which operates in free running mode. The 32-bit time base is captured through the timestamp_base[31:0] input.

20.4 Functional Description

The FlexCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system consists of a set of message buffers (MBs) that store configuration and control data, time stamp, message ID and data (see ”Message Buffer Structure”). The memory corresponding to the first 38 MBs can be configured to support a Legacy Rx FIFO reception scheme with a powerful ID filtering mechanism, capable of checking incoming frames against a table of

Yuntu Microelectronics

IDs (up to 128 extended IDs or 256 standard IDs or 512 8-bit ID slices), with individual mask register for up to 32 ID Filter Table elements.

For Classical CAN frames, simultaneous reception through the Legacy Rx FIFO and mailbox is supported. For CAN FD frames, reception is supported through mailboxes and the Enhanced Rx FIFO. For mailbox reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed in the ID field. A masking scheme makes it possible to match the ID programmed on the MB with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

A message buffer is said to be active at a given time if it can participate in both the matching and arbitration processes. An Rx MB with a 0b0000 code is inactive (see ”CAN message buffer code for RX buffers”). Similarly, a Tx MB with a 0b1000 or 0b1001 code is also inactive (see ”CAN message buffer code for TX buffers”).

The FlexCAN module can also receive and transmit messages in CAN FD format. The MBs are sized to adequately store the quantity of data bytes selected in the MBDSRn fields of the CAN_FDCTRL register. The quantity of FD MBs available for a given quantity of data bytes is described in ”CAN FD Control Register (CAN_FDCTRL)”. See also ”FlexCAN Memory Partition for CAN FD”.

20.4.1 Transmit Process

To transmit a CAN frame, the CPU must prepare a message buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (1001) to the CODE field of the Control and Status (C/S) word to request an abort of the transmission. Wait for the corresponding IFLAG bit to be asserted by polling the CAN_IFLAGn register or by the interrupt request if enabled by the respective IMASK bit. Then, read back the CODE field to check if the transmission was aborted or transmitted (see ”Transmission Abort Mechanism”) and clear the corresponding interrupt flag.
3. Write the ID word, including local priority if enabled by CAN_MCR.LPRIOEN.
4. Write the payload data bytes.
5. Configure the C/S word for the MB as needed:
   - Set IDE to the ID type.
   - For a Remote Transmission Request, set RTR.
   - If CAN FD is enabled (CAN_MCR.FDEN = 1), configure the EDL and BRS bits. **NOTE:** The ESI field does not need to be written, and will automatically be transmitted dominant by error active nodes and recessive by error passive nodes. The exception is if the FlexCAN is operating as a network gateway, in which case, the CPU writes the ESI bit according to the error status of the node that sent the message. For details about the relationship between the written and transmitted values of ESI, see ”CAN written and transmitted values of ESI bit”.
   - Write the Data Length Code to the DLC field.
   - Activate the MB to transmit the CAN frame by setting the CODE field to 0xC.

NOTE: To maximize software performance, configure all the fields in the C/S word in one 32-bit write operation. If the fields are configured in separate writes, the CODE field must be the last write in the C/S word.

When the MB is activated, it participates in the arbitration process and is eventually transmitted according to its priority. When the DLC value stored in the MB selected for transmission is larger than

Yuntu Microelectronics

the respective MB payload size, the FlexCAN adds the necessary number of bytes with constant 0xCC pattern to complete the expected DLC.

At the end of the successful transmission:

- The value of the Free Running Timer is written into the TIME STAMP field.
- The CODE field in the C/S word is updated.
- CAN_CRC and CAN_FDCRC registers are updated.
- A status flag is set in CAN_IFLAGn.
- An interrupt is generated if allowed by the corresponding CAN_IMASKn bit.

The new CODE field after transmission depends on the code that was used to activate the MB (see "CAN message buffer code for TX buffers").

When the Abort feature is enabled (CAN_MCR.AEN is asserted), after the Interrupt Flag is asserted for a mailbox configured as transmit buffer, the mailbox is blocked. Therefore, the CPU is not able to update it until the Interrupt Flag is negated by CPU. This means that the CPU must clear the corresponding CAN_IFLAGn bit before starting to prepare this MB for a new transmission or reception.

NOTE: If backwards compatibility is desired (CAN_MCR.AEN is negated), write the INACTIVE code (0b1000) to the CODE field to inactivate the MB. However, in this case, the pending frame may be transmitted without notification (see "Mailbox Inactivation").

20.4.2 Arbitration Process

The arbitration process scans the mailboxes, searching for the Tx mailbox that holds the message to be sent in the next opportunity. This mailbox is called the arbitration winner.

The scan starts from the lowest number mailbox and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the CAN_CTRL2.TASD field value.
- During the Error Delimiter field of a CAN frame.
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When there is a CPU write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and the CPU writes to the C/S word of any MB.
- When FlexCAN exits Bus Off state.
- Upon leaving Freeze mode or Low Power mode.

If the arbitration process does not manage to evaluate all mailboxes before the CAN bus has reached the first bit of the Intermission field, the temporary arbitration winner is invalidated and the FlexCAN does not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx mailboxes at the end of the scan according to the CAN_CTRL1.LBUF and CAN_MCR.LPRIOEN bit settings.

20.4.2.1 Lowest-Number Mailbox First

If CAN_CTRL1.LBUF is asserted, the first (lowest number) active Tx mailbox found is the arbitration winner. CAN_MCR.LPRIOEN bit has no effect when CAN_CTRL1.LBUF is asserted.

Yuntu Microelectronics

20.4.2.2 Highest-Priority Mailbox First

If CAN_CTRL1.LBUF is negated, then the arbitration process searches the active Tx mailbox with the highest priority, which means that this mailbox's frame would have a higher probability to win the arbitration on CAN bus when multiple external nodes compete for the bus at the same time.

The sequence of bits considered for this arbitration is called the arbitration value of the mailbox. The highest-priority Tx mailbox is the one that has the lowest arbitration value among all Tx mailboxes.

If two or more mailboxes have equivalent arbitration values, the mailbox with the lowest number is the arbitration winner.

The composition of the arbitration value depends on CAN_MCR.LPRIOEN.

20.4.2.2.1 Local Priority Disabled

If CAN_MCR.LPRIOEN is negated, Local Priority is disabled and the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame(see CAN composition of the arbitration value when local priority is disabled).

Table 20.87: CAN Composition of the Arbitration Value when Local Priority is Disabled

Format	Mailbox Arbitration Value(32 bits)				
Standard(IDE=0)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bits)
Extended(IDE=1)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

20.4.2.2.2 Local Priority Enabled

If CAN_MCR.LPRIOEN asserted, Local Priority is enabled. In this case, the mailbox PRIO field is included at the very left of the arbitration value(see CAN composition of the arbitration value when local priority is enabled).

Table 20.88: CAN Composition of the Arbitration Value when Local Priority is Enabled

Format	Mailbox Arbitration Value(35 bits)					
Standard(IDE=0)	PRIO (3 bits)	Standard ID (11 bits)	RTR (1 bit)	IDE (1 bit)	- (18 bits)	- (1 bits)
Extended(IDE=1)	PRIO (3 bits)	Extended ID[28:18] (11 bits)	SRR (1 bit)	IDE (1 bit)	Extended ID[17:0] (18 bits)	RTR (1 bit)

Because the PRIO field is the most significant part of the arbitration value, mailboxes with low PRIO values have higher priority than mailboxes with high PRIO values, regardless of the rest of their arbitration values.

- **NOTE:** The PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

Yuntu Microelectronics

20.4.2.3 Arbitration Completion

After the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called move-out and, after it is done, write access to the C/S word of the corresponding MB is blocked (if CAN_MCR.AEN is asserted). Write access is restored in the following events:

- After the MB is transmitted and the corresponding IFLAG bit is cleared by the CPU.
- The FlexCAN enters Freeze mode or Bus Off.
- The FlexCAN loses the bus arbitration or there is an error during the transmission.

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules.

20.4.2.4 Arbitration Start and Stop Conditions

The arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. CAN_CTRL2.TASD may be changed to optimize the arbitration start point.
- During CAN Bus Off state from TX_ERR_CNT = 124 to 128. CAN_CTRL2.TASD may be changed to optimize the arbitration start point.
- During C/S write by CPU in Bus Idle. The first C/S write starts arbitration process and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, the Tx arbitration process is pending. If there is no arbitration winner after the arbitration process has finished, the TX arbitration machine begins a new arbitration process. If there is a pending arbitration and Bus Idle state starts, an arbitration process is triggered. In this case, the first and second C/S write in Bus Idle do not restart the arbitration process. It is possible that there is not enough time to finish arbitration in Wait For Bus Idle state and the next state is Idle. In this case, the scan is not interrupted, and it is completed during Bus Idle state. During this arbitration, a C/S write does not cause arbitration restart.
- Arbitration winner deactivation during a valid arbitration window.
- Upon exiting Freeze mode (first bit of the Wait For Bus Idle state). If there is a resynchronization during Wait For Bus Idle, the arbitration process is restarted.

Arbitration process stops in the following situations:

- All mailboxes were scanned.
- A Tx active mailbox is found when the lowest buffer feature enabled.
- Arbitration winner inactivation or abort during any arbitration process.
- There was not enough time to finish the Tx arbitration process (for example, when a deactivation was performed near the end of frame). In this case, the arbitration process is pending.
- Error or Overload flag in the bus.
- Low power or Freeze mode request in Idle state.

Arbitration is considered pending as described below:

- It was not possible to finish the arbitration process in time.
- C/S write during arbitration if write is performed in a MB whose number is lower than the Tx arbitration pointer.
- Any C/S write if there is no Tx arbitration process in progress.
- Rx Match has just updated an Rx code to Tx code
- Entering Bus Off state.

Yuntu Microelectronics

C/S write during arbitration has the following effect:

- If the C/S write is performed in the arbitration winner, a new process is restarted immediately.
- If the C/S write is performed in an MB whose number is higher than the Tx arbitration pointer, the ongoing arbitration process will scan this MB as normal.

20.4.3 Receive Process

To be able to receive CAN frames into a mailbox, the CPU must prepare the mailbox for reception by executing the following steps:

1. If the mailbox is active (either Tx or Rx), inactivate the mailbox (see ”Mailbox Inactivation”), preferably with a safe inactivation (see ”Transmission Abort Mechanism”).
2. Write the ID word.
3. Write the EMPTY code (0b0100) to the CODE field of the C/S word to activate the mailbox. No setup is required for the EDL, BRS, and ESI bits; they are overwritten by the respective bit fields in the received message.

After the MB is activated, it will be able to receive frames that match the programmed filter.

At the end of a successful reception, the mailbox is updated by the move-in process (see ”Move-in”) as follows:

1. The received Data field (8 bytes at most for Classical CAN message format and up to 64 bytes for CAN FD message format) is stored.
2. The received ID field is stored.
3. The value of the Free Running Timer at the time of the second bit of frame’s Identifier field is written into the mailbox’s TIME STAMP field.
4. The received SRR, IDE, RTR, EDL, BRS, ESI, and DLC fields are stored.
5. The CODE field in the C/S word is updated(see ”CAN message buffer code for RX buffers”) and ”CAN message buffer code for TX buffers”)).
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended procedure for the CPU to service (read) the frame received in a mailbox is:

1. Read the C/S word of that mailbox.
2. Check the BUSY bit:
   - If the BUSY bit is asserted, the FlexCAN is updating the content of the mailbox. The mailbox does not become locked as a result of reading the C/S word. The CPU should defer reading the contents of the mailbox until a read of the C/S word shows the BUSY bit negated.
   - If the BUSY bit is negated and the CODE field is either FULL or OVERRUN, then the read of the C/S word causes the mailbox to become locked. The CPU can read the mailbox contents. The FlexCAN will not update the content of the mailbox while the mailbox is locked. See ”Mailbox Lock Mechanism”.
3. Read the contents of the mailbox.
4. Acknowledge the proper flag at the CAN_IFLAG registers.
5. Read the Free Running Timer to unlock the mailbox.

The CPU should poll for frame reception by checking the status flag bit for the specific mailbox in the respective CAN_IFLAGn register and not by polling the CODE field of that mailbox. Polling the CODE field does not work because, once a frame is received and the CPU services the mailbox (by reading the C/S word followed by unlocking the mailbox), the CODE field does not return to EMPTY. It remains

Yuntu Microelectronics

FULL, as expained in Table: ”CAN message buffer code for RX buffers”). If the CPU tries to work around this behavior by writing to the C/S word to force an EMPTY code after reading the mailbox without a prior safe inactivation, a newly received frame matching the filter of that mailbox may be lost.

CAUTION: In summary, never do polling by reading directly the C/S word of the mailboxes. Instead, read the CAN_IFLAGn registers.

The received frame's Identifier field is always stored in the matching mailbox. Therefore, the contents of the ID field in a mailbox may change if the match was due to masking.

When CAN_MCR.SRXDIS is asserted, the FlexCAN does not store frames transmitted by itself in any MB, even if it contains a matching Rx mailbox, and no interrupt flag or interrupt signal is generated. Otherwise, when CAN_MCR.SRXDIS is deasserted, the FlexCAN can receive frames transmitted by itself if there is a matching Rx mailbox.

20.4.3.1 Receiving CAN Frames through Enhanced Rx FIFO

To be able to receive CAN frames through the Enhanced Rx FIFO, the CPU must enable and configure the Enhanced Rx FIFO during Freeze mode (see ”Enhanced Rx FIFO”). The system can then use any of the following techniques to read CAN frames that have been received into the Enhanced Rx FIFO:

- Use the Enhanced Rx FIFO Data Available interrupt (CAN_ERFSR.ERFDA) to trigger CPU read of the Enhanced Rx FIFO output starting at address 0x2000. CAN_ERFSR.ERFDA is asserted when there are one or more frames available to be read from the Enhanced Rx FIFO.
- Use the Enhanced Rx FIFO Watermark Indication interrupt (CAN_ERFSR.ERFWMI) to trigger CPU read of the Enhanced Rx FIFO output starting at address 0x2000. CAN_ERFSR.ERFWM is asserted when the number of frames available to be read from the Enhanced Rx FIFO exceeds the watermark level written in CAN_ERFCR.ERFWM.
- Use DMA to read messages from the Enhanced Rx FIFO output starting at address 0x2000.

20.4.3.1.1 CPU Read Triggered by Data Available

The Enhanced Rx FIFO Data Available interrupt (ipi_int_erfda) is asserted when there is at least one message available to be read from the Enhanced Rx FIFO (CAN_ERFSR.ERFDA= 1) and the respective interrupt is enabled (CAN_ERFIER.ERFDAIE = 1).

When ipi_int_erfda is asserted, the oldest frame received in the Enhanced Rx FIFO is available at the Enhanced Rx FIFO output (addresses 0x2000-0x204F). The CPU should use the following procedure to read the available frame(s) from the Enhanced Rx FIFO:

1. Read the C/S word at address 0x2000 to determine the IDE, RTR, and DLC values of the frame.
2. Read the ID standard field and, if IDE= 1, the ID extended field at address 0x2004 to determine the message ID of the frame.
3. Read the Data field starting at address 0x2008 and continuing up to the address determined by the DLC field. For example, if DLC = 10, read data bytes 0-15 from addresses 0x2008-0x2014.
4. Read the ID HIT value from the next address above the address of the last data word (optional: indicates which Enhanced Rx FIFO Filter Element was hit by the received message, see ”ID HIT for Enhanced Rx FIFO”). For example, if DLC = 10, the address of the ID HIT field is 0x2018. See ”ID HIT and HR TIME STAMP Offsets” for the complete mapping of ID HIT address as determined by the DLC value.
5. If the High-Resolution Time Stamp feature is implemented and enabled, read the HR TIME STAMP value from the address above the address of ID HIT (optional). See ”ID HIT and HR TIME STAMP Offsets” for the complete mapping of HR TIME STAMP address as determined by the DLC value.

Yuntu Microelectronics

6. Clear the Enhanced Rx FIFO Data Available interrupt by writing 1 to CAN_ERFSR.ERFDA (mandatory: advances the Enhanced Rx FIFO read pointer to the next Enhanced Rx FIFO entry).

If there are one or more additional frames available to be read from the Enhanced Rx FIFO, ipi_int_erfda is again asserted. Repeat the above procedure to read additional frames until ipi_int_erfda remains de-asserted after clearing CAN_ERFSR.ERFDA.

NOTE: When DMA is not enabled, the Enhanced Rx FIFO read pointer only advances when CAN_ERFSR.ERFDA is cleared by writing 1 to it. When DMA is enabled, the Enhanced Rx FIFO read pointer advances automatically when the DMA controller reads the end address defined by CAN_ERFCR.DMALW.

20.4.3.1.2 CPU Read Triggered by Watermark Indication

The Enhanced Rx FIFO Watermark Indication interrupt (ipi_int_erfwm) is asserted when the number of message available to be read from the Enhanced Rx FIFO exceeds the watermark value in CAN_ERFCR.ERFWM and the respective interrupt is enabled (CAN_ERFIER.ERFWMIIE = 1).

When ipi_int_erfwm is asserted, there are one or more frames available to be ready from the Enhanced Rx FIFO and the oldest frame received in the Enhanced Rx FIFO is available at the Enhanced Rx FIFO output (addresses 0x2000-0x204F). The CPU should use the following procedure to read all available frames from the Enhanced Rx FIFO:

1. Optional: Read CAN_ERFSR.ERFEL to determine the number of frames currently available in the Enhanced Rx FIFO.
2. Read the C/S word at address 0x2000 to determine the IDE, RTR, and DLC values of the frame that is currently at the output of the Enhanced Rx FIFO.
3. Read the ID standard field and, if IDE= 1, the ID extended field at address 0x2004 to determine the message ID of the frame.
4. Read the Data field starting at address 0x2008 and continuing up to the address determined by the DLC field. For example, if DLC = 10, read data bytes 0-15 from addresses 0x2008-0x2014.

NOTE: As alternative to reading only the data bytes that are valid as determined by the DLC value, the interrupt service can always assumed the longest possible message (64 bytes) and read the entire Enhanced Rx FIFO output frame (0x2000-0x204F), then use the DLC value to determine which bytes contain valid data for the CAN frame.

5. Read the ID HIT value from the next address above the address of the last data word (optional: indicates which Enhanced Rx FIFO Filter Element was hit by the received message, see ”ID HIT for Enhanced Rx FIFO”). For example, if DLC = 10, the address of the ID HIT field is 0x2018. See ”ID HIT and HR TIME STAMP Offsets” for the complete mapping of ID HIT address as determined by the DLC value.
6. If the High-Resolution Time Stamp feature is implemented and enabled, read the HR TIME STAMP value from the address above the address of ID HIT (optional). See ”ID HIT and HR TIME STAMP Offsets” for the complete mapping of HR TIME STAMP address as determined by the DLC value.
7. Clear the Enhanced Rx FIFO Data Available bit by writing 1 to CAN_ERFSR.ERFDA (mandatory: advances the Enhanced Rx FIFO read pointer to next Enhanced Rx FIFO entry). In the same write operation, also write 1 to CAN_ERFSR.ERFWMI to clear the Enhanced Rx FIFO Watermark Indication bit so that it can be set again the next time the number of frames in the Enhanced Rx FIFO exceeds the value of CAN_ERFCR.ERFWM.
8. If there are one or more additional frames available to be read from the Enhanced Rx FIFO, CAN_ERFSR.ERFDA is immediately re-asserted. Read CAN_ERFSR.ERFDA to determine if there is another message available to be read from the Enhanced Rx FIFO.

Yuntu Microelectronics

To continue reading all available frames in the Enhanced Rx FIFO, repeat the above procedure to read additional frames until CAN_ERFSR.ERFDA remains de-asserted after writing 1 to it.

NOTE: When DMA is not enabled, the Enhanced Rx FIFO read pointer only advances when CAN_ERFSR.ERFDA is cleared by writing 1 to it. When DMA is enabled, the Enhanced Rx FIFO read pointer advances automatically when the DMA controller reads the end address defined by CAN_ERFCR.DMALW.

20.4.3.1.3 Using DMA to Service the Enhanced Rx FIFO

When CAN_MCR.DMA is asserted, the FlexCAN generates a DMA request when the number of frames stored in the Enhanced Rx FIFO is greater than the watermark defined in CAN_ERFCR.ERFWM. The DMA controller can then read the frames stored in the Enhanced Rx FIFO, accessing each frame as a message buffer starting at address 0x2000. To read all messages in the Enhanced Rx FIFO, the DMA controller must perform CAN_ERFCR.ERFWM+1 read loops, each ending at the address defined by CAN_ERFCR.DMALW.

Under DMA operation, the FlexCAN hardware automatically clears CAN_ERFSR.ERFDA and CAN_ERFSR.ERFWMI. The Enhanced Rx FIFO read pointer is automatically advanced each time the DMA controller reads the end address defined by CAN_ERFCR.DMALW.

For additional details, see section ”Enhanced Rx FIFO Under DMA Operation” to configure the FlexCAN to use Enhanced Rx FIFO with DMA.

To be able to receive CAN frames through the Legacy Rx FIFO, the CPU must enable and configure the Legacy Rx FIFO during Freeze mode (see ”Legacy Rx FIFO”). Upon receiving the Frames Available in Rx FIFO interrupt (see the description of the BUF5I bit in ”CAN IFLAG1 Register Interrupt Flags 1 Register (CAN_IFLAG1)”), the CPU should service the received frame using the following procedure:

1. Read the C/S word (optional: needed only if a mask was used for the IDE and RTR bits).
2. Read the ID field (optional: needed only if a mask was used).
3. Read the Data field.
4. Read the CAN_RXFIR register (optional).
5. Clear the Frames Available in Legacy Rx FIFO interrupt by writing 1 to CAN_IFLAG1.BUF5I (mandatory: releases the MB and allows the CPU to read the next Legacy Rx FIFO entry).

When CAN_MCR.DMA is asserted, upon receiving a frame in the Legacy Rx FIFO, CAN_IFLAG1.BUF5I generates a DMA request and does not generate a CPU interrupt (see ”Legacy Rx FIFO under DMA operation”). The CAN_IMASK1 bits in the Legacy Rx FIFO region are not used.

The DMA controller must service the received frame using the following procedure:

1. Read the C/S word (read 0x80 address, optional).
2. Read the ID field (read 0x84 address, optional).
3. Read all Data Bytes (start read at 0x88 address, optional).
4. Read the last Data Bytes (read 0x8C address is mandatory).

The DMA read must end by reading address 0x8C, which clears CAN_IFLAG1.BUF5I and updates both the Legacy Rx FIFO output with the next message (if the Legacy Rx FIFO is not empty) and the CAN_RXFIR register with the attributes of the new message. If there are more messages stored in the Legacy Rx FIFO, CAN_IFLAG1.BUF5I is re-asserted and another DMA request is issued. Otherwise, CAN_IFLAG1.BUF5I remains negated.

Yuntu Microelectronics

20.4.4 Matching Process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. If the Legacy Rx FIFO is enabled, the priority of scanning can be selected between mailboxes and FIFO filters. The matching starts from the lowest number MB toward the higher ones. If no match is found within the first structure then the other is scanned subsequently. In the event that the FIFO is full, the matching algorithm always looks for a matching MB outside the FIFO region.

For details about the Legacy Rx FIFO, see ”Legacy Rx FIFO”.

For details about the Enhanced Rx FIFO, see ”Enhanced Rx FIFO”.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- If the received frame is a remote frame, the start point is the CRC field of the frame.
- If the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame.
- If the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame.

If a matching ID is found in the FIFO table or in one of the mailboxes, the contents of the Rx SMB are transferred to the FIFO or to the matched mailbox by the move-in process. If any CAN protocol error is detected, then no match results are transferred to the FIFO or to the matched mailbox at the end of reception.

The matching process scans all matching elements of both Rx FIFO (if enabled) and the active Rx mailboxes (CODE is EMPTY, FULL, OVERRUN, or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The Rx SMB has the same structure as a mailbox. The reception structures (Rx FIFO or mailboxes) associated with the matching elements that had a successful comparison are the matched structures. The matching winner is selected at the end of the scan among those matched structures and depends on conditions described below.

Table 20.89: CAN Matching Architecture

Structure	SMB.RTR	CTRL2.RRS	CTRL2.EACEN	MB.IDE	MB.RTR	MB.ID[(1)]	MB.CODE
Mailbox	0	–	0	cmp[(2)]	nocmp[(3)]	cmp_msk[(4)]	EMPTY or FULL or OVERRUN
Mailbox	0	–	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
Mailbox	1	0	–	cmp	no_cmp	cmp	RANSWER
Mailbox	1	1	0	cmp	no_cmp	cmp_msk	EMPTY or FULL or OVERRUN

Yuntu Microelectronics

Table 20.89 continued from previous page

Structure	SMB.RTR	CTRL2.RRS	CTRL2.EACEN	MB.IDE	MB.RTR	MB.ID[1]	MB.CODE
Mailbox	1	1	1	cmp_msk	cmp_msk	cmp_msk	EMPTY or FULL or OVERRUN
FIFO[5]	–	–	–	cmp_msk	cmp_msk	cmp_msk	–

1. For mailbox structure, if SMB[IDE] is asserted, the ID is 29 bits(ID Standard ID Extended). If SMB[IDE] is negated, the ID is only 11 bits(ID Standard). For Rx FIFO structure, the ID depends on CAN_MCR[IDA].
2. cmp: Compares the Rx SMB contents with the MB contents regardless the masks.
3. no_cmp: The Rx SMB contents are not compared with the MB contents.
4. cmp_msk: Compares the Rx SMB contents with MB contents taking into account the masks.
5. SMB[IDE] and SMB[RTR] are not taken into account when CAN_MCR[IDAM] is format C.

NOTE: For details about the matching process for the Enhanced Rx FIFO, see "Enhanced Rx FIFO Matching Process".

A reception structure is free-to-receive when any of the following conditions is satisfied:

- The CODE field of the mailbox is EMPTY.
- The CODE field of the mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the CPU and unlocked as described in "Mailbox Lock Mechanism").
- The CODE field of the mailbox is either FULL or OVERRUN and an inactivation is performed. See "Mailbox Inactivation".
- The Legacy Rx FIFO or Enhanced Rx FIFO is not full.

The scan order for mailboxes and the legacy Rx FIFO is from the matching element with lowest number to the higher ones.

The matching winner search for mailboxes is affected by the CAN_MCR.IRMQ bit. If IRMQ is negated, the matching winner is the first matched mailbox regardless if it is free-to receive or not. If IRMQ is asserted, the matching winner is selected according to the priority below:

1. The first free-to-receive matched mailbox.
2. The last non free-to-receive matched mailbox.

It is possible to select the priority of scan between mailboxes and the legacy Rx FIFO or Enhanced Rx FIFO by using the CAN_CTRL2.MRP bit.

If the selected priority is Rx FIFO first:

- If the Rx FIFO is a matched structure and is free-to-receive, then the Rx FIFO is the matching winner regardless of the scan for mailboxes.
- Otherwise (the Rx FIFO is not a matched structure or is not free-to-receive), then the matching winner is searched for among mailboxes as described above.

If the selected priority is mailboxes first:

- If a free-to-receive matched mailbox is found, it is the matching winner regardless of the scan for the Rx FIFO.
- If no matched mailbox is found, then the matching winner is searched for in the scan for the Rx FIFO.

Yuntu Microelectronics

If both conditions above are not satisfied and a non-free-to-receive matched mailbox is found, then the matching winner determination is conditioned by the CAN_MCR.IRMQ bit:

- If CAN_MCR.IRMQ is negated, the matching winner is the first matched mailbox.
- If CAN_MCR.IRMQ is asserted, the matching winner is the last non-free-to receive matched mailbox.

See below Table for a summary of matching possibilities.

Table 20.90: CAN Matching Possibilities and Resulting Reception Structures

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception structure	Description
No FIFO, only MB, match is always MB first:						
0	0	X[(1)]	None[(2)]	–[(3)]	None	Frame lost by no match
0	0	X	Free[(4)]	–	First MB	
0	1	X	None	–	None	Frame lost by no match
0	1	X	Free	–	First MB	
0	1	X	Not Free	–	Last MB	Overrun
FIFO enabled, no match in FIFO is as if FIFO does not exist:						
1	0	X	None	None[(5)]	None	Frame lost by no match
1	0	X	Free	None	First MB	
1	1	X	None	None	None	Frame lost by no match
1	1	X	Free	None	First MB	
1	1	X	Not Free	None	Last MB	Overrun
FIFO enabled, Queue disabled:						
1	0	0	X	Not Full[(6)]	FIFO	
1	0	0	None	Full[(7)]	None	Frame lost by FIFO full (FIFO Overflow)
1	0	0	Free	Full	First MB	
1	0	0	Not Free	Full	First MB	
1	0	1	None	Not Full	FIFO	
1	0	1	None	Full	None	Frame lost by FIFO full (FIFO Overflow)
1	0	1	Free	X	First MB	
1	0	1	Not Free	X	First MB	Overrun
FIFO enabled, Queue enabled:						
1	1	0	X	Not Full	FIFO	
1	1	0	None	Full	None	Frame lost by FIFO full (FIFO Overflow)
1	1	0	Free	Full	First MB	
1	1	0	Not Free	Full	Last MB	Overrun
1	1	1	None	Not Full	FIFO	
1	1	1	Free	X	First MB	

Yuntu Microelectronics

Table 20.90 continued from previous page

RFEN	IRMQ	MRP	Matched in MB	Matched in FIFO	Reception structure	Description
1	1	1	Not Free	Not Full	FIFO	
1	1	1	Not Free	Full	Last MB	Overrun

1. This is a ”don’t care” condition.
2. Matched in MB ”None” means that the frame has not matched any MB (free-to-receive or non-freetoreceive).
3. This is a forbidden condition.
4. Matched in MB ”Free” means that the frame matched at least one MB free-to-receive regardless of whether it has matched MBs non-free-to-receive.
5. Matched in FIFO ”None” means that the frame has not matched any filter for the Rx FIFO. It is as if the Rx FIFO does not exist (CAN_CTRL2.RFEN = 0).
6. Matched in FIFO ”Not Full” means that the frame has matched a FIFO filter and has empty slots to receive it.
7. Matched in FIFO ”Full” means that the frame has matched a FIFO filter but the FIFO couldn’t store it because the FIFO has no empty slots to receive it.

If a non-safe mailbox inactivation (see ”Mailbox Inactivation”) occurs during the matching process and the mailbox inactivated is the temporary matching winner, then the temporary matching winner is invalidated. The matching elements scan is not stopped nor restarted; it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that the Rx FIFO is disabled, IRMQ is asserted, there are two MBs with the same ID, and the FlexCAN starts receiving messages with that ID. If the two MBs are the second and the fifth in the array, then when the first message arrives, the matching algorithm finds the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm finds MB number 2 again, but it is not “free-to-receive”, so it keeps looking, finds MB number 5 and stores the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are “free-to-receive”, so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue (in addition to the full-featured FIFO) to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages are queued into the MBs. The CPU can examine the TIME STAMP field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID Acceptance Masks. The FlexCAN supports individual masking per MB (see ”Rx Individual Mask Registers (CAN_RXIMRn)”). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is a “don’t care”. The Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed when the FlexCAN is in Freeze mode; otherwise, they are blocked by hardware.

The FlexCAN also supports an alternate masking scheme with only four mask registers (CAN_RXFGMASK, CAN_RXMGMASK, CAN_RX14MASK, and CAN_RX15MASK) for backward compatibility with legacy applications. This alternate masking scheme is enabled when CAN_MCR.IRMQ is negated.

Yuntu Microelectronics

20.4.5 Receive Process under Pretended Networking Mode

Pretended Networking mode adds specific Wake Up functionality in low power modes (Doze and Deepsleep mode). When Pretended Network mode (PN) is enabled by asserting CAN_MCR.PNET_EN, the FlexCAN continues processing Rx CAN messages under low power mode, and is able to detect specific Wake Up messages by filtering them against ID and payload target values using a pre-selected matching criteria. Wake Up functionality is not available for messages in CAN FD format. When the FlexCAN is in Pretended Networking mode, CAN FD format messages are ignored.

PN registers are located in the 0x0B00–0x0B7C address range and can be written only in Freeze mode. These registers are used for writing PN configuration (both control and target values) prior to entering Pretended Networking mode, and for reading Wake Up flags and the received message ID and data when returning back to normal mode after Wake Up. The CPU must wait for CAN_MCR.LPMACK to be negated before performing any access to FlexCAN PN registers.

PN control registers are described in "CAN CTRL1 PN Register (CAN_CTRL1_PN)" and "CAN CTRL2 PN Register (CAN_CTRL2_PN)". The control bit fields that configure the filtering criteria are:

- PLFS: Payload filtering selection.
- IDFS: ID filtering selection.
- FCS: Filtering combination selection.

PN target values are:

- FLT_IDE: IDE target value used to filter the incoming message by its format (standard or extended).
- FLT_RTR: RTR target value used to filter the incoming message by its type (data or remote frame).
- FLT_DLC_HI and FLT_DLC_LO: Target DLC range used to filter the size of payload part of an incoming message.
- FLT_ID1: ID target value used to filter the incoming message ID (equal to, smaller than or equal, greater than or equal, or the lower limit value in an ID range).
- FLT_ID2: ID target value used as the upper limit in an ID range.
- PL1: Payload target value used to filter the incoming message payload (equal to, smaller than or equal, greater than or equal, or the lower limit value in a payload range).
- PL2: Payload target value used as the upper limit in a payload range.

The IDE, RTR, ID, and payload filters have their respective masks. These masks determine which bits are taken into account in equality comparisons (1 in certain mask positions) and which ones are don't care (0 in other mask positions). ID and payload masks are used only for exact ID and/or exact payload comparisons.

The ID of Rx incoming messages can be filtered based on the following criteria:

- A match with the exact ID value by detecting the equality between the ID field of the incoming message and the content of target CAN_FLT_ID1 register. The ID mask is used.
- A match with the maximum range of ID; in other words, any message with ID value smaller than or equal to the content of target CAN_FLT_ID1 register is accepted. The ID mask is not used.
- A match with the minimum range of ID; in other words, any message with ID value greater than or equal to the content of target CAN_FLT_ID1 register is accepted. The ID mask is not used.
- A match inside a range of IDs; in other words, any message with an ID value that is greater than or equal to the content of target CAN_FLT_ID1 register and smaller than or equal to the content of target CAN_FLT_ID2_IDMASK register is accepted. The ID mask is not used.

The above criteria for ID filtering must be coherent with FLT_IDE and FLT_RTR target values in

Yuntu Microelectronics

CAN_FLT_ID1 register. Only Rx frames that match the respective IDE and RTR bits to the contents of FLT_IDE and FLT_RTR bit fields are compared. When range of IDs is selected (CAN_CTRL1_PN.IDFS = 11), both FLT_ID1 and FLT_ID2 are referred to the same FLT_IDE and FLT_RTR bits in the CAN_FLT_ID1 register.

The ID mask is applied only to the exact ID comparison filtering option (CAN_CTRL1_PN.IDFS = 00) to determine which bits are taken into account in the comparison. For the exact match option, the mask can select any bit within the ID field. For maximum range, minimum range, and inside range comparisons, the ID mask is not considered.

The IDE and RTR masks are applied in both exact and range ID comparison filtering options to determine which bits are taken into account in comparison.

Similarly to the ID criteria, 64-bit data or payloads (PL) of Rx incoming messages can be filtered based on the following criteria:

- A match with the exact payload value by detecting the equality between the payload field of the incoming message and the content of PL1 register. The payload mask is used.
- A match with the maximum range of payload; in other words, any message with payload value smaller than or equal to the content of PL1 register is accepted. The payload mask is not used.
- A match with the minimum range of payload; in other words,. any message with payload value greater than or equal to the content of PL1 register is accepted. The payload mask is not used.
- A match inside a range of payloads; in other words, any message with a payload value that is greater than or equal to the content of PL1 register and smaller than or equal to the content of PL2 register is accepted. The payload mask is not used.

The above criteria for payload filtering must be coherent with FLT_DLC upper and lower limit values in CAN_FLT_DLC register. The payload of a Rx incoming message is filtered in accordance to the selected criteria only if the DLC value of the Rx incoming message is inside a DLC range:

- Greater than or equal to the FLT_DLC_LO (lower limit) and
- Lower than or equal to the FLT_DLC_HI (upper limit)

Conversely, a DLC value out of the specified range results in mismatch. By making FLT_DLC_LO = FLT_DLC_HI, only payloads of the specified quantity of bytes will be filtered. DLC is not maskable.

When the inside range of payloads option is selected (CAN_CTRL1_PN.PLFS = 11), both PL1 and PL2 are considered with the 8-byte data length. All the data bytes excluded by the DLC of the received message are considered with value zero.

Payload mask is only used in the exact match option (CAN_CTRL1_PN.PLFS = 00) to select which bits or bytes in the 8-byte data field of both Rx incoming message and the contents of PL1 register are selected for matching. Mask length must be in accordance with the expected range of DLC values. For maximum range, minimum range, and inside range comparisons, the payload mask is not considered.

When a remote frame is received by the FlexCAN and CAN_CTRL1_PN.FCS is configured to select the payload comparison, the payload filtering is not considered and the comparison results in a mismatch.

Rx incoming messages can also be filtered based upon the quantity and rate of message reception, specifically:

- Several messages that match the filtering criteria for ID or payload for a predefined number of times. This number can be configured in the 1 to 255 range. See ”CAN CTRL1 PN Register (CAN_CTRL1_PN)”.
- No message matching the filtering criteria for ID or payload up to a timeout trigger. That is, non-reception of a matching message for a defined quantity of time. ”CAN CTRL2 PN Register

Yuntu Microelectronics

(CAN_CTRL2_PN)".

The FlexCAN can generate a Wake Up timeout event from an internal timer with associated comparator circuitry capable of generating a timeout flag when the counting reaches the pre-defined timeout value, as specified in CAN_CTRL2_PN.MATCHTO.

The above filtering criteria can be used together as follows:

- Message ID filtering only.
- Message ID filtering and Payload filtering.
- Message ID filtering only occurring N times.
- Message ID filtering and Payload filtering occurring N times.

The timeout counter runs concurrently with the reception filtering process. Both engines, timeout counter and message filtering, are independent. If an incoming message matches the selected filter criteria, the timeout counter keeps counting until the CPU wakes up.

Conversely, if the timeout counter reaches the target value, then the message filtering process continue to filter incoming messages until the CPU wakes up. The CAN_WU_MTC.MCOUNTER field reports the number of matched messages that occurred under Pretended Networking mode up to the moment the CPU wakes up.

Under Pretended Networking mode, a Wake Up event that occurs sets the respective Wake Up flag (see "Pretended Networking Wake Up Match Register (CAN_WU_MTC)" :

- In case of a successful match in accordance with the selected filtering criteria, CAN_WU_MTC.WUMF.
- In case of a timeout trigger, CAN_WU_MTC.WTOF.

Either of these flags generates an interrupt to the CPU if the respective mask bit is asserted (WUMF_MSK or WTOF_MSK in the CAN_CTRL1_PN register).

There are four Wake Up Message Buffers (WMBs) used to store incoming messages in Pretended Networking mode. Up to four messages can be stored (see "Wake Up Message Buffer Register for ID (CAN_WMBn_ID)").

When the CAN_CTRL1_PN.NMATCH value is 1, only one message is received if the filtering criteria are matched, and this message is stored in CAN_WMB0.

If the NMATCH value is between 2 and 4, CAN_WMB1, CAN_WMB2, and CAN_WMB3 are used to store the second, third, and fourth matching messages, respectively.

If NMATCH is greater than 4, the last four matching messages are stored in the WMBs. The WMB index indicates the arrival order. The last message is stored in CAN_WMB3.

Only the valid data bytes of the incoming match message is stored in the data field of WMBs. The non-valid data bytes are read as zero. In case the of DLC = 0 and RTR = 1, the data field is filled with zeros. In any of the above cases, the Wake Up interrupt is generated

when the filtering criteria is completed and CAN_CTRL1_PN.WUMF_MSK is asserted. When a non-match Wake Up event occurs (timeout or external) and CAN_WU_MTC.MCOUNTER is equal to or greater than 4, the message stored in WMB0 does not have a valid content. The CAN_WMB0 is used as the buffer for the current message on the CAN bus. Messages received during Pretended Networking mode do not have timestamps and respective field in the WMB structure must be ignored.

Under low-power mode (Doze or Deepsleep), all processes are shut down except for the PN functionality in the PE submodule, which is kept clocked by the Oscillator clock (see "Clock Domains and

Yuntu Microelectronics

Restrictions"). The FlexCAN continues to receive Rx incoming messages, but only compares them to the predefined target values and according to the selected filtering criteria. The matching, arbitration, move-in, and move-out processes, normally available in Normal mode, are not performed under Pretended Networking mode.

Under Pretended Networking, the FlexCAN reacts to messages on the CAN bus in the same manner as in Normal mode (generates acknowledge bits, detects and counts errors, and so on).

20.4.6 Move Process

There are two types of move process: move-in and move-out.

20.4.6.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to an Rx mailbox or FIFO that has matched it. If the move destination is the Legacy Rx FIFO, attributes of the message are also copied to the CAN_RXFIR FIFO. Each Rx SMB has its own move-in process, but only one is performed at a given time as described below. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see "Matching Process") and all of the following conditions are true:

- The CAN bus has reached or let past either:
  - The second bit of the Intermission field next to the frame that carried the message that is in the Rx SMB, or
  - The first bit of an overload frame next to the frame that carried the message that is in the Rx SMB
- There is no ongoing matching process.
- The destination mailbox is not locked by the CPU.
- There is no ongoing move-in process from another Rx SMB. If more than one move-in processes are to be started at the same time, both are performed and the newest substitutes for the oldest.

The term pending move-in is used throughout the documentation and stands for a move-tobe that still does not satisfy all of the aforementioned conditions.

The move-in is canceled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- The destination mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished.
- There is a previous pending move-in to the same destination mailbox.
- The Rx SMB is receiving a frame transmitted by the FlexCAN itself and self-reception is disabled (CAN_MCR.SRXDIS bit is asserted)
- Any CAN protocol error is detected.

Note that the pending move-in is not canceled if the FlexCAN enters Freeze or low-power mode. The pending move-in stays on hold, waiting for exit from Freeze or low-power mode and to be unlocked. If an MB is unlocked during Freeze mode, the move-in happens immediately.

The move-in process is the execution by the FlexCAN of the following steps:

1. Push IDHIT into the RXFIR FIFO if the message is destined to the Legacy Rx FIFO.
2. Read all data words from the Rx SMB according to the selected payload size for the Rx storage element.

Yuntu Microelectronics

3. Write all data words to the Rx mailbox according to the selected payload size for the Rx storage element. If the data size of the storage element is smaller than the original payload size described in the message's DLC field, the payload is truncated and the high order bytes that do not fit the destination size are lost.
4. Read the C/S and ID words from the Rx SMB.
5. Write C/S and ID words to the Rx mailbox, and update the CODE field.

The move-in process is not atomic, in such a way that it is immediately canceled by the inactivation of the destination mailbox (see "Mailbox Inactivation"). In this case, the mailbox may be left partially updated, thus incoherent. The exception is if the move-in destination is a Legacy or Enhanced Rx FIFO message buffer, in which case the process cannot be canceled.

The BUSY bit (least significant bit of the CODE field) of the destination message buffer is asserted while the move-in is being performed to alert the CPU that the message buffer content is temporarily incoherent.

20.4.6.2 Move-out

The move-out process is the copy of the content from a Tx mailbox to the Tx SMB when a message for transmission is available (see "Arbitration Process"). The moveout occurs in the following conditions:

- The first bit of Intermission field.
- During Bus Off state when TX Error Counter is in the 124 to 128 range.
- During Bus Idle state.
- During Wait For Bus Idle state.

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently out of Bus Idle state. In Bus Idle, the move-out has the lowest priority to the concurrent memory accesses.

20.4.7 Data Coherence

In order to maintain data coherency and FlexCAN proper operation, the CPU must obey the rules described in "Transmit Process" and "Receive Process".

20.4.7.1 Transmission Abort Mechanism

The abort mechanism provides a safe way to request the abortion of a pending transmission. A feedback mechanism is provided to inform the CPU if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

Two primary conditions must be fulfilled in order to abort a transmission:

- CAN_MCR.AEN must be asserted.
- The first CPU action must be the writing of abort code (1001) into the CODE field of the C/S word.

Active MBs configured for transmission must be aborted first before they can be updated. If the abort code is written to a mailbox that is currently being transmitted or to a mailbox that was already loaded into the Tx SMB for transmission, the write operation is blocked and the transmission is not disturbed. However, the abort request is captured and kept pending until one of the following conditions is satisfied:

- The FlexCAN loses the bus arbitration.
- There is an error during the transmission.
- The FlexCAN is put into Freeze mode.

Yuntu Microelectronics

- The FlexCAN enters the Bus Off state.
- There is an overload frame.

If none of the conditions above are reached, the MB is transmitted correctly, the interrupt flag is set in the respective CAN_IFLAGn register, and an interrupt to the CPU is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. On the other hand, if one of the above conditions is reached, the frame is not transmitted; therefore, the abort code is written into the CODE field, the interrupt flag is set in the respective CAN_IFLAGn register, and an interrupt is (optionally) generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, then the write operation is not blocked; therefore, the MB is updated and the interrupt flag is set. In this way, the CPU only needs to read the abort code to make sure the active MB was safely inactivated. Although the AEN bit is asserted and the CPU wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One mailbox is only aborted when the abort request is captured and kept pending until one of the previous conditions is satisfied.

The abort procedure can be summarized as follows:

- CPU checks the corresponding IFLAG and clears it, if asserted.
- CPU writes 1001 into the CODE field of the C/S word.
- CPU waits for the corresponding IFLAG indicating that the frame was either transmitted or aborted.
- CPU reads the CODE field to check if the frame was either transmitted (CODE = 1000) or aborted (CODE = 1001).
- It is necessary to clear the corresponding IFLAG in order to allow the MB to be reconfigured.
- It is necessary to reconfigure the EDL, BRS, and ESI fields of the aborted MB before transmitting it again.

20.4.7.2 Mailbox Inactivation

Inactivation is a mechanism provided to protect the mailbox against updates by the FlexCAN internal processes, thus allowing the CPU to rely on mailbox data coherence after having updated it, even in Normal mode.

Inactivation of transmission mailboxes must be performed only when MCR.AEN bit is deasserted.

If a mailbox is inactivated, it participates in neither the arbitration process nor the matching process until it is reactivated. See ”Transmit Process” and ”Receive Process” for more detailed instructions on how to inactivate and reactivate a mailbox.

To inactivate a mailbox, the CPU must update its CODE field to INACTIVE (either 0000 or 1000).

Because the CPU is not able to synchronize the CODE field update with the FlexCAN internal processes, an inactivation can have the following consequences:

- A frame on the bus that matches the filtering of the inactivated Rx mailbox may be lost without notice, even if there are other mailboxes with the same filter.
- A frame containing the message within the inactivated Tx mailbox may be transmitted without setting the respective IFLAG.

To perform a safe inactivation and avoid the above consequences for Tx mailboxes, the CPU must use the Transmission Abort mechanism (see ”Transmission Abort Mechanism”).

The inactivation automatically unlocks the mailbox (see ”Mailbox Lock Mechanism”).

Yuntu Microelectronics

NOTE: Message buffers that are part of the Legacy Rx FIFO or Enhanced Rx FIFO cannot be inactivated. There is no write protection on the Legacy Rx FIFO region by FlexCAN. The CPU must maintain data coherency in the Legacy Rx FIFO region when RFEN is asserted.

20.4.7.3 Mailbox Lock Mechanism

Other than mailbox inactivation, the FlexCAN has another data coherence mechanism for the receive process. When the CPU reads the C/S word of an Rx MB with codes FULL or OVERRUN, the FlexCAN assumes that the CPU wants to read the whole MB in an atomic operation and therefore it sets an internal lock flag for that MB. The lock is released when the CPU reads the Free Running Timer (global unlock operation), or when it reads the C/S word of another MB regardless of its code. A CPU write into the C/S word also unlocks the MB, but this procedure is not recommended for normal unlock use because it cancels a pending move and potentially may lose a received message. The MB locking prevents a new frame from being written into the MB while the CPU is reading it.

NOTE: The locking mechanism applies only to Rx MBs that are not part of the Legacy Rx FIFO and have a code other than INACTIVE (0000) or EMPTY (0100). Tx MBs cannot be locked.

NOTE: In previous FlexCAN versions, reading the C/S word locked the MB even if it was EMPTY. This behavior is maintained when the IRMQ bit is negated.

Suppose, for example, that the Legacy Rx FIFO is disabled and the second and the fifth MBs of the array are programmed with the same ID, and the FlexCAN has already received and stored messages into these two MBs. Suppose now that the CPU decides to read MB number 5 and at the same time another message with the same ID is arriving. When the CPU reads the C/S word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no free-to-receive MBs, so it decides to override MB number 5. However, this MB is locked, so the new message cannot be written there. It will remain in the Rx SMB waiting for the MB to be unlocked, and only then will be written to the MB.

If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the message in the Rx SMB and there is no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved in from the Rx SMB to the MB, the BUSY bit on the CODE field is asserted. If the CPU reads the C/S word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

NOTE: If the BUSY bit is asserted or if the MB is empty, then reading the C/S word does not lock the MB.

Inactivation takes precedence over locking. If the CPU inactivates a locked Rx MB, then its lock status is negated and the MB is marked as invalid for the current matching round. Any pending message on the Rx SMB will not be transferred to the MB. An MB is unlocked when the CPU reads the Free Running Timer Register, or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during a low power mode (see ”Modes of operation”), and it takes place only when the module returns to Normal or Freeze mode.

20.4.8 Enhanced Rx FIFO

As an alternative to the 6-message Legacy Rx FIFO, the FlexCAN supports an Enhanced Rx FIFO, which can store up to 32 CAN FD messages. The region 0x2000–0x204F contains the output of the Enhanced Rx

Yuntu Microelectronics

FIFO, which should be read by the CPU. The Enhanced Rx FIFO is enabled by setting CAN_ERFCR.ERFEN.

The Legacy Rx FIFO and Enhanced Rx FIFO cannot be enabled at the same time.

The Enhanced Rx FIFO has a watermark, which is configured by setting CAN_ERFCR.ERFWM. If CAN_ERFCR.ERFWM= 1, the CPU is notified only if a minimum number of messages is stored in the FIFO. When the number of stored messages is greater than the value in CAN_ERFCR.ERFWM, CAN_ERFSR.ERFWMI is set by the hardware. An interrupt can be triggered by setting CAN_ERFIER.ERFWMIIE. A DMA transfer can be triggered by setting CAN_MCR.DMA.

For the Enhanced Rx FIFO to receive, the CPU must execute the following configuration procedure:

1. Enter Freeze mode.
2. Enable the Enhanced Rx FIFO (if it is not already enabled) by setting ERFCR.ERFEN.

   NOTE: CAN_MCR.RFEN must be 0 when enabling the Enhanced Rx FIFO.
3. Write 1 to CAN_ERFSR.ERFCLR to reset Enhanced Rx FIFO engine.
4. Clear CAN_ERFSR.ERFUFW, CAN_ERFSR.ERFOVF, CAN_ERFSR.ERFWMI, and CAN_ERFSR.ERFDA, if they are set.
5. Write to CAN_ERFCR.NFE to configure the total number of Enhanced Rx FIFO filter elements to be used in Enhanced Rx FIFO reception.
6. Write to CAN_ERFCR.NEXIF to configure the number of extended ID and standard ID filter elements to be used in Enhanced Rx FIFO reception. CAN_ERFCR.NEXIF must be less than or equal to EFRCR.NFE + 1.
7. Configure the Enhanced Rx FIFO watermark by writing CAN_ERFCR.ERFWM.
8. If interrupts are to be used, set the interrupt enables in the ERIER register.
9. If DMA is to be used, set CAN_MCR.DMA to enable DMA operation and write CAN_ERFCR.DMALW to configure the number of words to transfer for each Enhanced Rx FIFO data element.
10. Configure the filter elements by writing to the CAN_ERFFELn registers.

    NOTE: The CAN_ERFFELn registers are in RAM and need to be explicitly initialized before any reception.
11. Exit Freeze mode.

The above procedure must be done if the CPU needs to change any of the configuration of the Enhanced Rx FIFO.

20.4.8.1 Enhanced Rx FIFO Filter Elements

There are two types of Enhanced Rx FIFO filter elements that can be stored in the CAN_ERFFELn registers: extended ID filter element and standard ID filter element. Each extended ID filter element is stored in two CAN_ERFFELn registers; each standard-ID filter element is stored in one CAN_ERFFELn register. The total number of Enhanced RX FIFO filter elements is defined by CAN_ERFCR.NFE.

The filter memory space can be split into two regions: one for extended ID filter elements and one for standard ID filter elements, as controlled by CAN_ERFCR.NEXIF. Figure ”Number of enhanced RX FIFO

Yuntu Microelectronics

filter elements" shows how the Enhanced Rx FIFO filter elements are defined for the following example configuration:

- Configuration parameter ERX_FIFO_FLT_ELEM = 128
- CAN_ERFCR.NEXIF = 3 (There are 3 extended ID filter elements, each occupying two CAN_ERFFELn registers.)
- CAN_ERFCR.NFE = 63 (There are 2 x (63 - 3 + 1) = 122 standard ID filter elements, each occupying one CAN_ERFFELn register.)

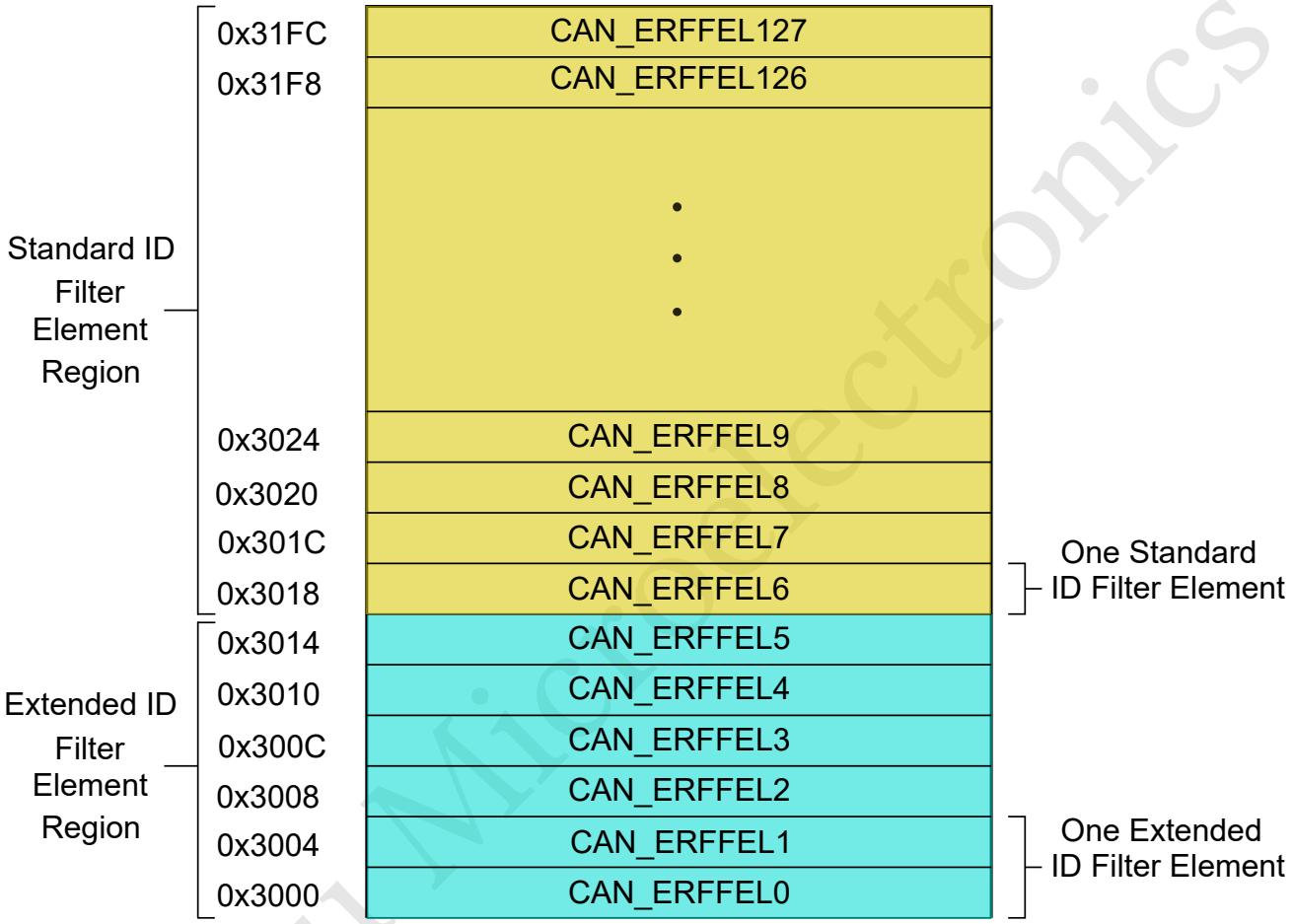


Figure 20.6: Number of Enhanced RX FIFO Filter Elements

20.4.8.2 Enhanced Rx FIFO Matching Process

When CAN_ERFCR.ERFEN = 1, the FlexCAN scans the CAN_ERFFELn memory region. If at least one filter element satisfies the matching criteria, the CAN message content is transferred to the Enhanced RX FIFO memory.

NOTE: If multiple filters match the incoming message ID, then the first matching filter found by the matching process is indicated in ID HIT.

Each CAN_ERFFELn register can store one standard filter element. The matching criteria are defined by CAN_ERFFELn.FSCH as follows:

- If FSCH = 00, the filter scheme is based on mask and filter. A CAN message matches a standard ID filter element only if the following criteria are met:

Yuntu Microelectronics

– CAN message is base frame format (IDE = 0)
– (ID[n] = STD ID Filter [n]) or (STD ID Mask [n] =0) for each bit n from 0 to 10
– (RTR = RTR Filter) or (RTR Mask = 0)

RTR and ID are the Remote Transmit Request bit and the ID from a CAN message. If FSCH = 00, the filters and masks are defined as shown in below Figure.

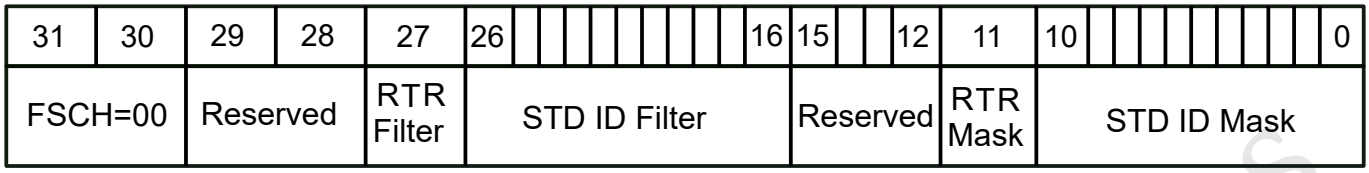


Figure 20.7: Standard ID Filter Element with Filter Mask Scheme FSCH 00

- If FSCH = 01, the filter scheme is based on range. A CAN message matches a standard ID filter element only if the following criteria are met:
  – CAN message is base frame format (IDE = 0)
  – ID ≥ STD ID Filter1
  – ID ≤ STD ID Filter2
  – (RTR = RTR Filter) or (RTR Mask = 0)

  RTR and ID are the Remote Transmit Request bit and ID from a CAN message. If FSCH = 01, the filters and masks are defined as shown in below Figure.

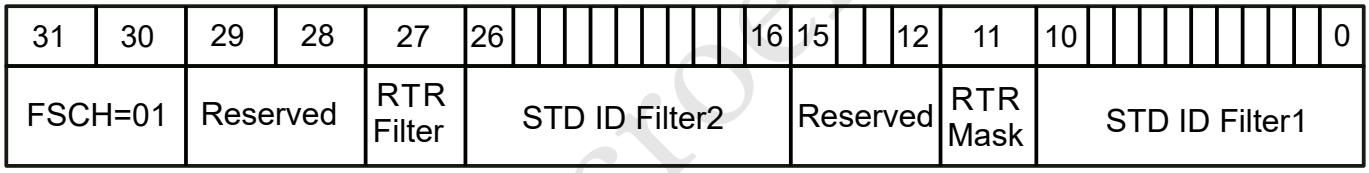


Figure 20.8: Standard ID Filter Element with Range Scheme FSCH 01

- If FSCH = 10, the filter scheme is based on two filters without masks. A CAN message matches a standard ID filter element only if the following criteria are met:
  – CAN message is base frame format (IDE = 0)
  – (ID[n] = STD ID Filter1[n]) or (ID[n] = STD ID Filter2[n]) for each bit n from 0 to 10
  – (RTR = RTR Filter1) or (RTR = RTR Filter2)

  RTR and ID are the Remote Transmit Request bit and ID from a CAN message. If FSCH = 10, the filters and masks are defined as shown in below Figure.

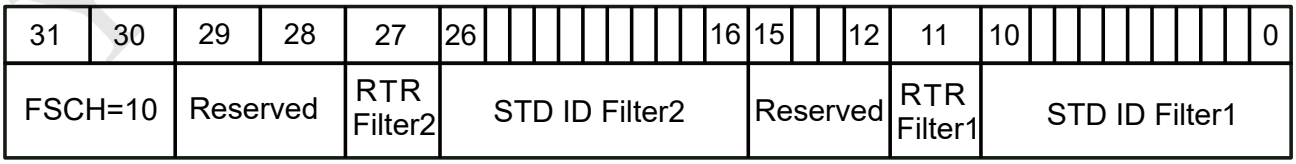


Figure 20.9: Standard ID Filter Element with Two Filter Scheme FSCH 10

Each pair of CAN_ERFFELn registers can store one extended filter element. The matching criteria are defined by CAN_ERFFELn.FSCH as follows:

Yuntu Microelectronics

- If FSCH = 00, the filter scheme is based on mask and filter. A CAN message matches an extended ID filter element only if the following criteria are met:
  - CAN message is extended frame format (IDE = 1)
  - (ID[n] = EXT ID Filter [n]) or (EXT ID Mask [n] =0) for each bit n from 0 to 28
  - (RTR = RTR Filter) or (RTR Mask = 0)

  If FSCH = 00, the filters and masks are defined as shown in below Figure.

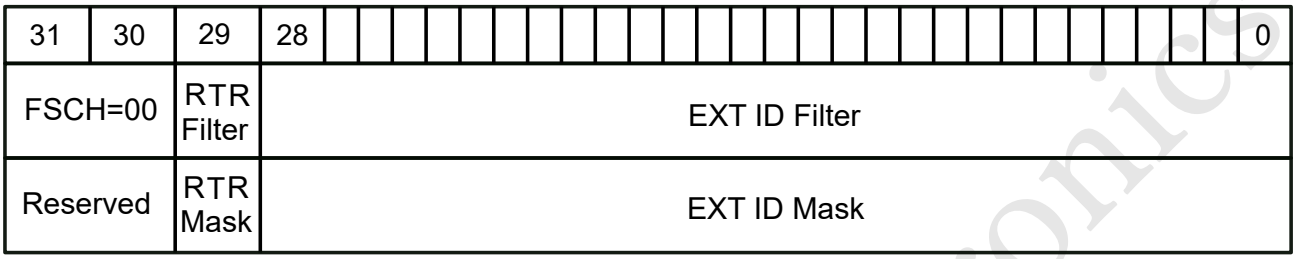


Figure 20.10: Extended ID Filter Element with Filter Mask Scheme FSCH 00

- If FSCH = 01, the filter scheme is based on range. A CAN message matches an extended ID filter element only if the following criteria are met:
  - CAN message is extended frame format (IDE = 1)
  - ID ≥ STD ID Filter1
  - ID ≤ STD ID Filter2
  - (RTR = RTR Filter) or (RTR Mask = 0)

  If FSCH = 01, the filters and masks are defined as shown in below Figure.

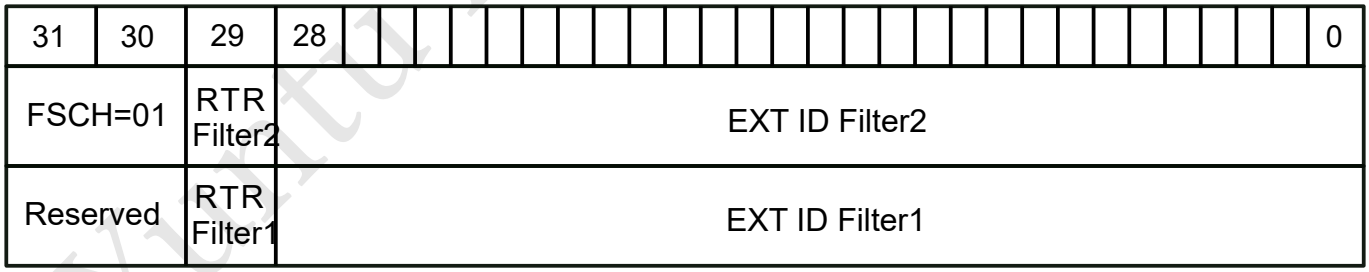


Figure 20.11: Extended ID Filter Element with Filter Mask Scheme FSCH 01

- If FSCH = 10, the filter scheme is based on two filters without masks. A CAN message matches an extended ID filter element only if the following criteria are met:
  - CAN message is extended frame format (IDE = 1)
  - (ID[n] = EXT ID Filter1[n]) or (ID[n] = EXT ID Filter2[n]) for each bit n from 0 to 28
  - (RTR = RTR Filter1) or (RTR = RTR Filter2)

  If FSCH = 10, the filters and masks are defined as shown in below Figure.

Yuntu Microelectronics

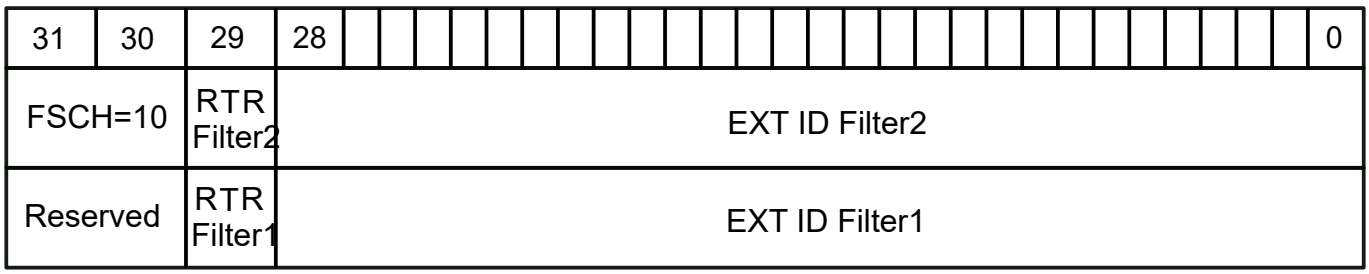


Figure 20.12: Extended ID Filter Element with Two Filter Scheme FSCH 10

20.4.8.3 Enhanced Rx FIFO Under DMA Operation

The DMA feature is enabled by asserting both CAN_ERFCR.ERFEN and CAN_MCR.DMA.

The DMA controller can read the received message by reading a message buffer structure at the Enhanced Rx FIFO output port at the address range defined in Enhanced Rx FIFO structure.

The CPU should not access the Enhanced Rx FIFO output port address range during DMA operation, so that the FIFO engine operates properly. Before setting CAN_MCR.DMA, the CPU must service Enhanced Rx FIFO status bits. Otherwise, these bits may show that the FIFO has data to be serviced, and mistakenly generate a DMA request.

Before clearing CAN_MCR.DMA, the CPU must first clear the ERFUFW, ERFOVF, ERMWMI, and ERFDA fields in the CAN_ERFSR register, then clear the Enhanced Rx FIFO engine by writing 1 to CAN_ERFSR.ERFCLR.

CAN_ERFSR.ERFDA is set by the hardware when there is at least one frame available to be read from the Enhanced Rx FIFO, and CAN_ERFSR.ERFWMI is set by the hardware when the number of frames stored in the FIFO is greater than the watermark defined in CAN_ERFCR.ERFWM. A DMA request is generated when CAN_ERFSR.ERFWMI is set. Upon receiving the request, the DMA controller can read the message in the Enhanced Rx FIFO output. The DMA controller must perform the number of read loops in the FIFO output defined by CAN_ERFCR.ERFWM. Also, each message reading process must end at the address defined in CAN_ERFCR.DMALW.

The following rules must be followed for Enhanced Rx FIFO DMA operation:

- Because a DMA transfer cannot be dynamically changed, CAN_ERFCR.DMALW should be programmed so that the Enhanced Rx FIFO element can store the largest CAN message present on the CAN bus.
- Data bytes are valid according to the DLC field. See ”CAN data bytes validity”.

Each time one message is read from Enhanced Rx FIFO by the DMA controller, CAN_ERFSR.ERFDA is cleared by the FlexCAN and set again if there is at least one message stored in the FIFO. CAN_ERFSR.ERFWMI is cleared by the hardware when the messages are read in the FIFO output, and set again if the number of messages in the FIFO is greater than the value defined by CAN_ERFCR.ERFWM.

20.4.8.4 Enhanced Rx FIFO Clear Operation

When CAN_ERFCR.ERFEN is set, the CPU can clear the Enhanced Rx FIFO by writing 1 to CAN_ERFSR.ERFCLR. The clear operation resets the internal FIFO pointers, although the FIFO content stored in RAM is not changed. This operation can only be performed in Freeze mode and is blocked by hardware in other modes. The clear operation does not clear the ERFUFW, ERFOVF, ERFDA, and ERFWMI fields in CAN_ERFSR. Consequently, the CPU must service all these fields before executing the clear FIFO task.

Yuntu Microelectronics

20.4.9 Legacy Rx FIFO

The Legacy Rx FIFO is receive-only and is enabled by asserting the CAN_MCR.RFEN bit.

The reset value of this bit is zero to maintain software backward compatibility with previous versions of the FlexCAN module that did not have the Legacy Rx FIFO feature.

Caution: The Legacy Rx FIFO must not be enabled when the CAN FD feature is enabled.

The Legacy Rx FIFO is 6 messages deep. The memory region occupied by the Legacy Rx FIFO structure (both message buffers and Legacy Rx FIFO engine) is described in "Rx FIFO Structure". The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a message buffer structure at the output of the Legacy Rx FIFO.

CAN_IFLAG1.BUF5I (Frames available in Legacy Rx FIFO) is asserted when there is at least one frame available to be read from the Legacy Rx FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, the CPU can read the message (accessing the output of the Legacy Rx FIFO as a message buffer) and the CAN_RXFIR register, then clear the interrupt. If there are more messages in the Legacy Rx FIFO, the act of clearing the interrupt updates the output of the FIFO with the next message and updates CAN_RXFIR with the attributes of that message, reissuing the interrupt to the CPU. Otherwise, the flag remains negated. The output of the Legacy Rx FIFO is only valid while CAN_IFLAG1.BUF5I is asserted.

CAN_IFLAG1.BUF6I (Legacy Rx FIFO Warning) is asserted when the number of unread messages in the Legacy Rx FIFO is increased to 5 from 4 due to the reception of a new message, meaning that the Legacy Rx FIFO is almost full. The flag remains asserted until the CPU clears it.

CAN_IFLAG1.BUF7I (Legacy Rx FIFO Overflow) is asserted when an incoming message was lost because the Legacy Rx FIFO is full. CAN_IFLAG1.BUF7I is not asserted when the Legacy Rx FIFO is full and the message was captured by a mailbox. BUF7I remains asserted until the CPU clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing workload. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to the CAN_CTRL2.RFFN setting, that can be configured to one of the following formats:

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

NOTE: A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

See also "Legacy Rx FIFO Structure".

Every frame available in the Legacy Rx FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can read in the IDHIT field from C/S word, as shown in the Legacy Rx FIFO structure description. Another way the CPU can obtain this information is by accessing the CAN_RXFIR register. The CAN_RXFIR.IDHIT field refers to the message at the output of the Legacy Rx FIFO and is valid while the CAN_IFLAG1.BUF5I flag is asserted. The CAN_RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the Legacy Rx FIFO.

Up to 32 elements of the filter table are individually affected by the individual Mask Registers

Yuntu Microelectronics

(CAN_RXIMRn), according to the setting of CAN_CTRL2.RFFN, allowing very powerful filtering criteria to be defined. If CAN_MCR.IRMQ is negated, the Legacy Rx FIFO filter table is affected by CAN_RXFGMASK.

20.4.9.1 Rx FIFO Under DMA Operation

The Legacy Rx FIFO can support DMA. The Legacy Rx FIFO with DMA feature is enabled by asserting both CAN_MCR.RFEN and CAN_MCR.DMA. The reset value of CAN_MCR.DMA is zero to maintain backward compatibility with previous versions of the FlexCAN that did not have the DMA feature.

The DMA controller can read the received message by reading a message buffer structure at the Legacy Rx FIFO output port at the 0x80–0x8C address range.

When CAN_MCR.DMA is asserted, the CPU must not access the Legacy Rx FIFO output port address range. Before asserting CAN_MCR.DMA, the CPU must service the IFLAGs asserted in the Legacy Rx FIFO region. Otherwise, these IFLAGs may show that the Legacy Rx FIFO has data to be serviced, and mistakenly generate a DMA request. Before negating CAN_MCR.DMA, the CPU must perform a clear Legacy Rx FIFO operation.

CAN_IFLAG1.BUF5I (Frames available in Legacy Rx FIFO) is asserted when there is at least one frame available to be read from the Legacy Rx FIFO; consequently, a DMA request is generated simultaneously. Upon receiving the request, the DMA controller can read the message (accessing the output of the Legacy Rx FIFO as a message buffer). The DMA reading process must end by reading address 0x8C, which clears CAN_IFLAG1.BUF5I and updates both the Legacy Rx FIFO output with the next message (if the Legacy Rx FIFO is not empty) and the CAN_RXFIR register with the attributes of the new message. If there are more messages stored in the Legacy Rx FIFO, CAN_IFLAG1.BUF5I is re-asserted and another DMA request is issued. Otherwise, CAN_IFLAG1.BUF5I remains negated.

NOTE: CAN_RXFIR register contents cannot be read after DMA completes the Legacy Rx FIFO read. The IDHIT information is also available in the C/S word at address 0x080 (see ”Legacy Rx FIFO Structure”).

The CAN_IFLAG1.BUF6I and CAN_IFLAG1.BUF7I flags are not used when the DMA feature is enabled.

When the FlexCAN is working with DMA, the CPU does not receive any Legacy Rx FIFO interrupt and must not clear the related IFLAGs. In addition, the related IMASKs are not used to mask the generation of DMA requests.

20.4.9.2 Clear Legacy Rx FIFO Operation

When CAN_MCR.RFEN is asserted, the clear Legacy Rx FIFO operation is a feature used to empty Legacy Rx FIFO contents. With CAN_MCR.RFEN asserted, the clear Legacy Rx FIFO occurs when the CPU writes 1 to CAN_IFLAG1.BUF0I. This operation can only be performed in Freeze mode and is blocked by hardware in other modes. This operation does not clear the Legacy Rx FIFO IFLAGs; consequently, the CPU must service all Legacy Rx FIFO IFLAGs before executing the clear Legacy Rx FIFO task.

When the Legacy Rx FIFO is working with DMA, the clear Legacy Rx FIFO operation clears the CAN_IFLAG1.BUF5I flag and the DMA request is canceled.

Caution: The clear Legacy Rx FIFO operation does not clear IFLAGs, except when CAN_MCR.DMA is asserted. When CAN_MCR.DMA is asserted, only the CAN_IFLAG1.BUF5I flag is cleared.

20.4.10 DMA Operation

DMA operation is available to service either the Enhanced Rx FIFO or Legacy Rx FIFO. To enable DMA, set the following bits while the FlexCAN is in Freeze mode:

Yuntu Microelectronics

- CAN_MCR.DMA = 1 to enable DMA.
- Either CAN_ERFCR.ERFEN to enable the Enhanced Rx FIFO or CAN_MCR.RFEN to enable the Legacy Rx FIFO.

A DMA request is then generated on ipd_req when the respective Rx FIFO has data available:

- For the Enhanced Rx FIFO, a DMA request is generated when the number of entries available in the Enhanced Rx FIFO exceeds the watermark set in CAN_ERFCR.ERFWM.
- For the Legacy Rx FIFO, a DMA request is generated when there are one or more entries available to be read from the Legacy Rx FIFO.

The DMA controller must then service the DMA request by following the DMA service protocol for the respective FIFO. For details, refer to:

- See "Enhanced Rx FIFO Under DMA Operation" and "Using DMA to Service the Enhanced Rx FIFO" for the Enhanced Rx FIFO.
- See "Legacy Rx FIFO Under DMA Operation" and "Using DMA to Service the Legacy Rx FIFO" for the Legacy Rx FIFO.

20.4.11 CAN Protocol Related Features

This section describes the CAN protocol related features.

20.4.11.1 CAN FD Frames

The ISO 11898-1 standard specifies the Classical Frame format compliant to ISO 11898-1 (2003) and introduces the CAN Flexible Data Rate Frame format. The Classical Frame format allows bit rates up to 1 Mbit/s and payloads up to 8 bytes per frame. The Flexible Data Rate Frame format allows bit rates higher than 1 Mbit/s and payloads longer than 8 bytes per frame. The FlexCAN can receive and transmit CAN FD messages interleaved with Classical CAN messages.

There are three additional control bits in the CAN FD frame:

- The Extended Data Length (EDL) bit enables a longer data payload with different data length coding.
- The Bit Rate Switch (BRS) bit decides whether the bit rate is switched inside a CAN FD format frame.
- The Error State Indicator (ESI) flag is transmitted dominant by error active nodes, and recessive by error passive nodes.

There are no remote frames in the CAN FD format. A message configured to transmit a remote frame is always sent out in the Classical CAN format. When a FD frame is received and matches a mailbox, the RTR bit in the receiving message buffer is negated. The RTR bit must be considered in Classical CAN frames only.

CAN FD messages may be formatted as long frames, in which the data field exceeds 8 bytes and may range from 12 up to 64 bytes. CAN FD messages can also be configured to support bit rate switching, where the control field, the data field, and the CRC field of a CAN frame are transmitted with a higher bit rate than the beginning and the end of the frame. Messages in Classical CAN format are limited to transport a maximum payload of 8 bytes at nominal rate. Figure: "CAN Message Formats" illustrates the message formats for Classical and FD frames with either standard or extended ID.

The ability to receive and transmit CAN FD messages is enabled by the CAN_MCR.FDEN bit. Either a recessive R0 bit in CAN frames with 11-bit identifiers or a recessive R1 bit in CAN frames with 29-bit identifiers are decoded as an EDL bit (not a reserved one). A CAN FD frame is recognized by a recessive

Yuntu Microelectronics

EDL bit, while a Classical CAN frame is recognized by a dominant EDL bit. The BRS bit specifies whether this frame switches the bit rate in its data phase. A long frame is decoded according to the DLC field value.

CAN FD messages can be transmitted with two different bit rates. The first part of a CAN FD frame, from the Start Of Frame (SOF) bit until the Bit Rate Switch (BRS) bit, also called the arbitration phase, is transmitted with the nominal bit rate based on a set of nominal CAN bit timing configuration values. The second part, from the BRS bit until the CRC Delimiter bit, also named the data phase, is transmitted with the data bit rate defined by a second set of CAN data bit timing configuration values. Finally, from the CRC Delimiter until the Intermission bits, the transmission resumes to nominal bit rate. In CAN FD frames with bit rate switching, the bit timing is changed inside the frame at the sample point of the BRS bit if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the nominal CAN bit timing is used as defined by the CAN_CBT register (also by CAN_CTRL1 register for backward compatibility). Upon detecting a recessive BRS bit, the CAN data bit timing is used as defined by the CAN_FDCBT register.

Yuntu Microelectronics

NOTE: If the length of the time quantum in the nominal bit timing and the length of the time quantum in the data bit timing are not identical, a quantization error of up to one time quantum of the arbitration phase may be present as a phase error. This situation can occur after the switch from arbitration to data phase and lasts until the next synchronization event. Thus, the length of the time quantum should be the same in nominal and data bit timing in order to minimize the chance of error frames on the CAN bus, and to optimize the clock tolerance in networks that use CAN FD frames.

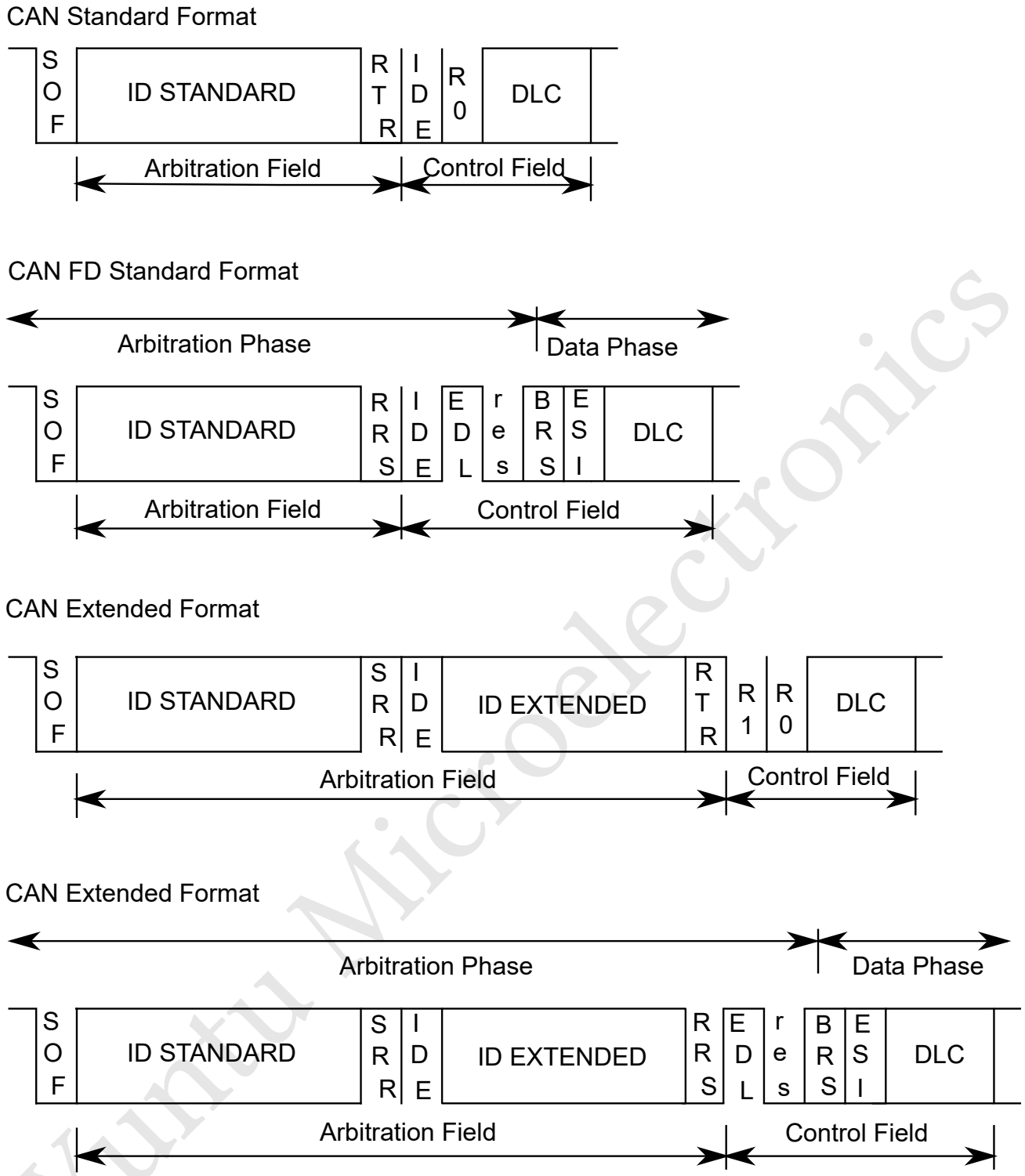


Figure 20.13: CAN Message Formats

CAN_FDCTRL.FDRATE enables the transmission of all frames with bit rate switching if the BRS bit in the selected Tx MB is set. If FDRATE is negated, the transmission is performed at nominal rate regardless of the BRS bit value. The CAN_FDCTRL.FDRATE bit can be written any time but takes effect only for the next message transmitted or received.

The nominal bit timing is resumed at either the sample point of the CRC Delimiter bit or when an error is detected, whichever occurs first. Figure: ”Bit rate switching mechanism for can fd messages” describes

Yuntu Microelectronics

the mechanism for entering and leaving the data phase when BRS bit is recessive.

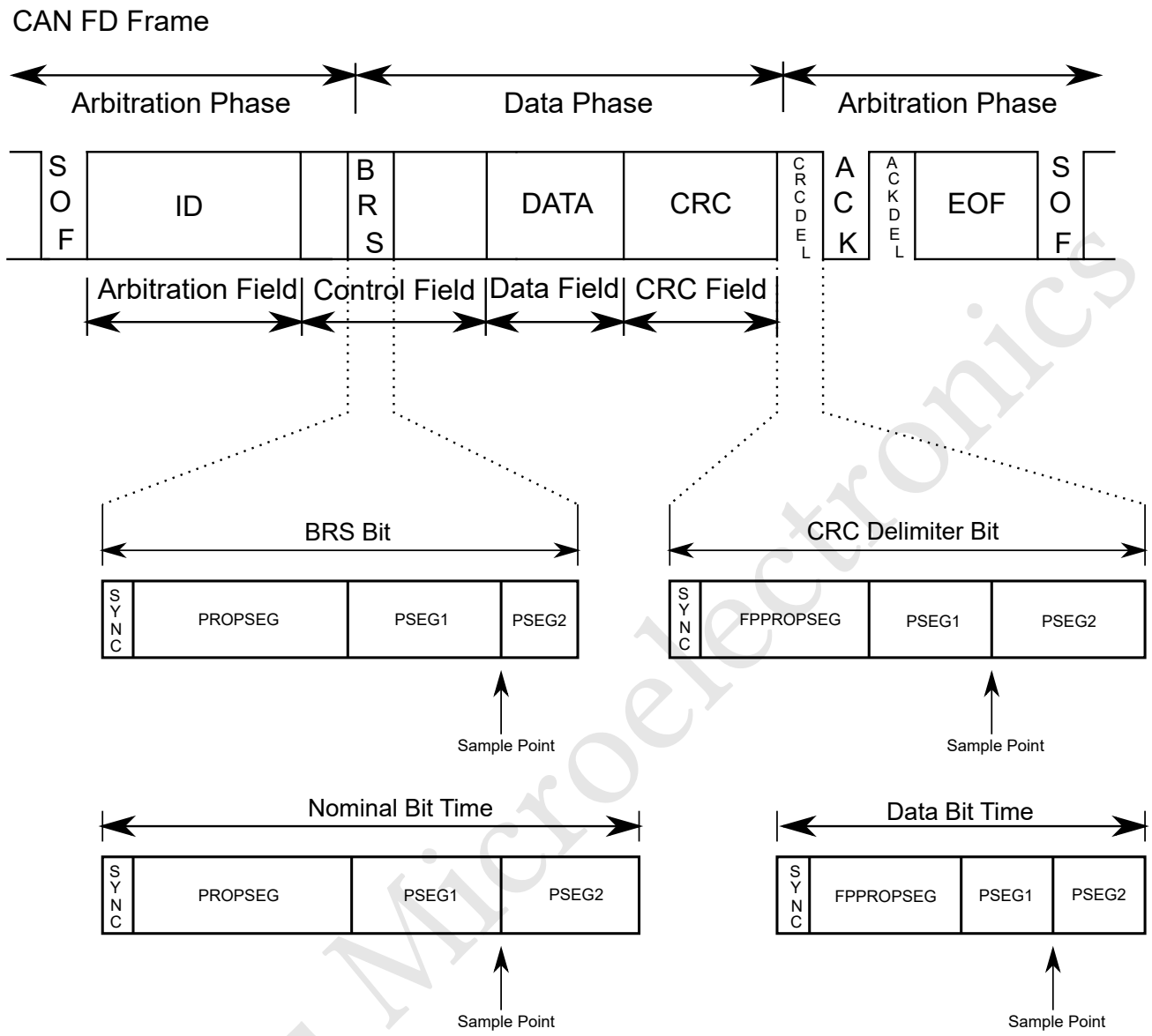


Figure 20.14: Bit Rate Switching Mechanism for CAN FD Messages

NOTE: In Classical CAN frames, the CRC delimiter is one single recessive bit. In CAN FD frames, the CRC delimiter may consist of one or two recessive bits. The FlexCAN sends only one recessive bit as the CRC delimiter, but it accepts two recessive bits before the edge from recessive to dominant that starts the acknowledge slot. As a receiver, the FlexCAN sends its acknowledge bit after the first CRC delimiter bit. In CAN FD frames, the FlexCAN accepts a two-bit dominant ACK slot as a valid ACK to compensate for phase shifts between the receivers.

The maximum configurable bit rate in the CAN FD data phase depends on the clock frequency of the PE submodule. For example, with a PE clock frequency of 40 MHz and the shortest configurable bit time of 5 time quanta, the bit rate in the data phase is 8 Mbit/s.

The value of the ESI bit is determined either by the transmitter's error state at the start of the transmission, if the frame is originated in the FlexCAN node, or by the original transmitting node if the FlexCAN is acting as a gateway for the message. If the transmitter is error passive, ESI is transmitted

Yuntu Microelectronics

recessive; otherwise, it is transmitted dominant(see ”CAN written and transmitted values of ESI bit”).

Table 20.91: CAN Written and Transmitted Values of ESI Bit

FlexCAN Fault Confinement Status at Start of Frame	ESI Bit of Tx MB	Transmitted ESI
Error active	0	0 (Error active)
Error passive	0	1 (Error passive)
Error active	1	1 (Error passive)
Error passive	1	1 (Error passive)

There are different CRC polynomials for different CAN frame formats:

- The first polynomial, CRC_15, is used for all frames in Classical CAN format.
- The second, CRC_17, is used for frames in CAN FD format with a data field up to sixteen bytes long.
- The third, CRC_21, is used for frames in CAN FD format with a data field longer than sixteen bytes.

Each polynomial results in a Hamming distance of 6. At the start of the frame, all three CRC polynomials are calculated concurrently. The CRC sequence to be transmitted is selected by the values of the EDL bit and the DLC bit field. When receiving a message, the FlexCAN decodes EDL and DLC to select the proper CRC polynomial to check for a CRC error.

In CAN FD format frames, stuff bits are included in the bit stream for CRC calculation. In Classical CAN format frames, stuff bits are not included. After the transmission of the last bit relevant to the CRC calculation, the CAN_FDCRC register stores the calculated CRC for the transmitted message, with the adequate length for the type of message, for both CAN FD and non-FD messages. The CAN_CRCR register reports a valid CRC for Classical CAN messages only.

In CAN FD format frames, the CAN bit stuffing method is changed for the CRC sequence so that the stuff bits are inserted at fixed positions. When the FlexCAN is transmitting a CAN FD frame, a fixed stuff bit is inserted just before the first bit of the CRC sequence, even if the last bits of the preceding field do not fulfill the CAN stuff condition. Additional stuff bits are inserted after each fourth bit of the CRC sequence. The value of any fixed stuff bit is the inverse value of its preceding bit. When the FlexCAN is receiving a CAN FD frame, it discards the fixed stuff bits from the bit stream for the CRC check. A Stuff Error is detected if the fixed stuff bit has the same value as its preceding bit.

The FlexCAN detects errors in CAN FD frames the same way as in Classical CAN frames. The error counters RXERRCNT and TXERRCNT in the CAN_ECR register accumulate the counts of Rx and Tx errors, respectively, for both FD and non-FD frames indiscriminately. There are two extra error counters (RXERRCNT_FAST and TXERRCNT_FAST) that accumulate Rx and Tx errors occurring in the data phase of CAN FD frames with the BRS bit set only. The rules for updating the error counters are the same for both CAN FD and non-FD frames (see ”Error Counter (CAN_ECR)”).

Error flags BITERR1, BITERR0, ACKERR, CRCERR, FRMERR, and STFERR in the CAN_ESR1 register report errors in both CAN FD and non-FD frames. They also generate the ERRINT interrupt if CAN_CTRL1.ERRMSK is asserted. The CAN_ESR1 register has additional error flags (BITERR1_FAST, BITERR0_FAST, CRCERR_FAST, FRMERR_FAST, and STFERR_FAST) to individually indicate the occurrence of errors in the data phase of CAN FD frames with the BRS bit set. There is no ACKERR detected in the data phase of a CAN FD frame. Fault confinement status reported in CAN_ESR1.FLTCONF is the same for both CAN FD and Classical CAN

Yuntu Microelectronics

frames, and is based on RXERRCNT and TXERRCNT error counters only. Information contained in RXERRCNT_FAST and TXERRCNT_FAST counters may be considered as status to help detect the error nature related to the bit rate value.

When the FlexCAN is in the data phase, either transmitting or receiving a CAN FD message, and detects an error, it immediately switches back to the arbitration phase and to the nominal rate to start an Error Flag.

Resynchronization and Hard Synchronization occur in CAN FD frames in the same way as in Classical CAN frames. A Hard Synchronization is also performed at the recessive-todominant edge from EDL to R0 in CAN FD format frames. The FlexCAN does not resynchronize while transmitting in the CAN FD data phase.

20.4.11.2 Transceiver Delay Compensation

The CAN FD protocol allows the transmission and reception of data at a higher bit rate than the nominal rate used in the arbitration phase when the message's BRS bit is set. This feature enables the use of rates up to 8 Mbps.

During the data phase of a CAN FD frame, the transmitter detects a bit error if it cannot receive its own latest transmitted bit at the sample point of that bit. When bit rate switching is enabled (BRS bit is asserted), the length of the CAN bit time in the data phase can become shorter than the transceiver's loop delay, thus impeding the correct comparison between the transmitted bit and the received bit within the current CAN bit time interval.

The FlexCAN supports an optional Transceiver Delay Compensation (TDC) mechanism that defines a secondary sample point where the transmitted bit is correctly compared with the received bit in order to check for bit errors.

The TDC mechanism is enabled by the CAN_FDCTRL.TDCEN bit and is effective only during the data phase of CAN FD frames having the BRS bit set. It has no effect either on non-FD frames, or on FD frames transmitted at normal bit rate. The TDC is active from the sample point of the BRS bit until the sample point of the CRC Delimiter bit, provided the respective message under transmission has the BRS bit set. When it is active, a comparison is done between the real received bit and the delayed transmitted bit, where the delay is calculated based on the measured transceiver loop delay.

NOTE: The actual value of the CRC Delimiter bit is disregarded by transmitters using the Transceiver Delay Compensation mechanism. A global error at the end of the CRC Field will cause the receivers to send error frames that the transmitter will detect during Acknowledge or End of Frame.

For every transmitted FD frame having the BRS bit set, the delay measurement is triggered by the transition from the recessive EDL bit to the dominant R0 bit (as shown in Figure: "Transceiver loop delay measurement".

The loop delay is measured in PE clock periods (CANCLK, see "Protocol Timing"), from the transmitted EDL-R0 edge to the received EDL-R0 edge. The position of the secondary sample point is defined by the measured loop delay time added to an offset value specified in CAN_FDCTRL.TDCOFF. The CAN_FDCTRL.TDCVAL bit field stores the result of this calculation. The TDCVAL value saturates at its maximum value of 15 CANCLK when the delay measurement is too long.

Yuntu Microelectronics

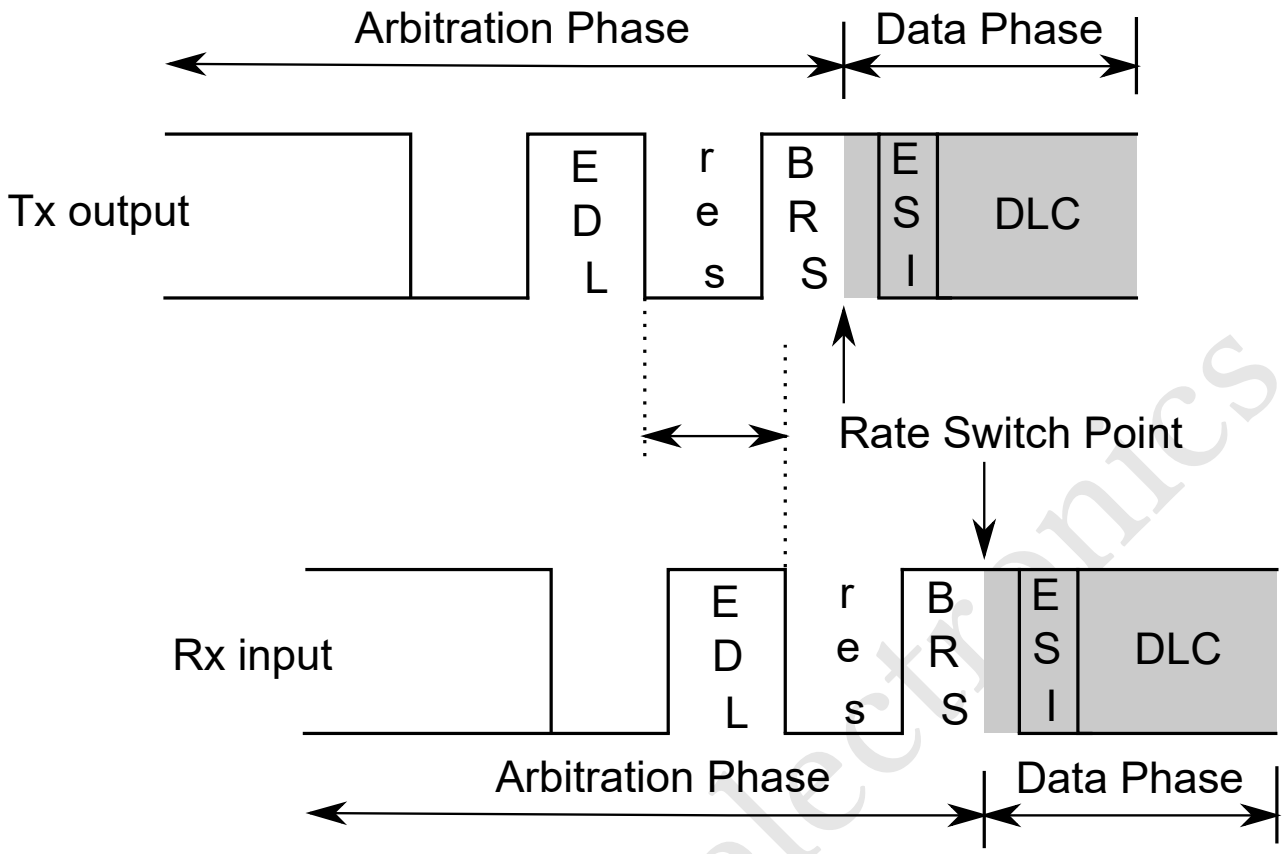


Figure 20.15: Transceiver Loop Delay Measurement

The measured loop delay is not enough to be used to define the secondary sample point because it relates to the CAN bit edges. The transceiver delay compensation offset TDCOFF or ETDCOFF is used to shift the secondary sample point from the edge to an intermediate point inside the bit time (for example, half of the bit time in the data phase), far away from its edges. Therefore, the TDCOFF or ETDCOFF value cannot be larger than the CAN bit duration in the data phase.

If the secondary sample point is set very near the CAN bit edge (SYNC field), then problems may occur during the bit sampling in the data phase. For the TDC to work reliably, the offset has to use optimal settings. To be sure the bit sampling is performed in the best region, the TDC offset should be configured as shown in the following equation:

Offset = (FPSEG1 + FPROPSEG + 2) x (FPRESDIV + 1)

Offset = (DTSEG1 + 2) x (EDPRESDIV + 1), if ETDCEN = 1

Below Figure shows the secondary sample point (SSP) position when these settings are used.

Yuntu Microelectronics

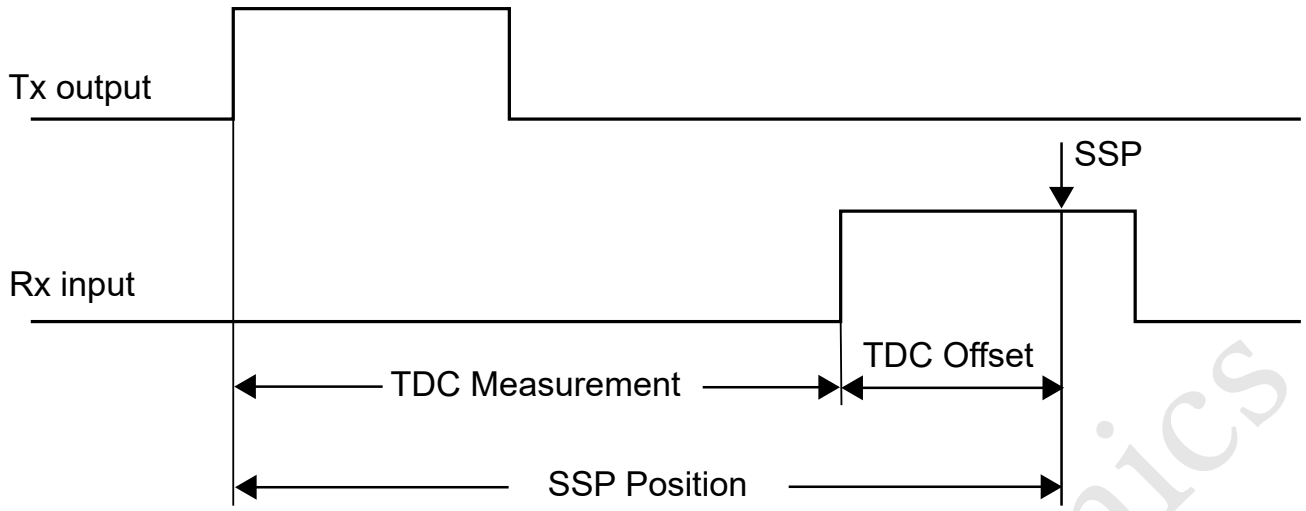


Figure 20.16: SSP Position with Optional Values

Alternatively, if CAN_CTRL2.BTE and CAN_ETDC.ETDCEN are set, the CAN_ETDC.TDMDIS bit can be set to disable the transceiver delay measurement. In this case, the SSP position is defined only by the CAN_ETDC.ETDCOFF field. Figure 64 shows the SSP position when the transceiver delay measurement is disabled.

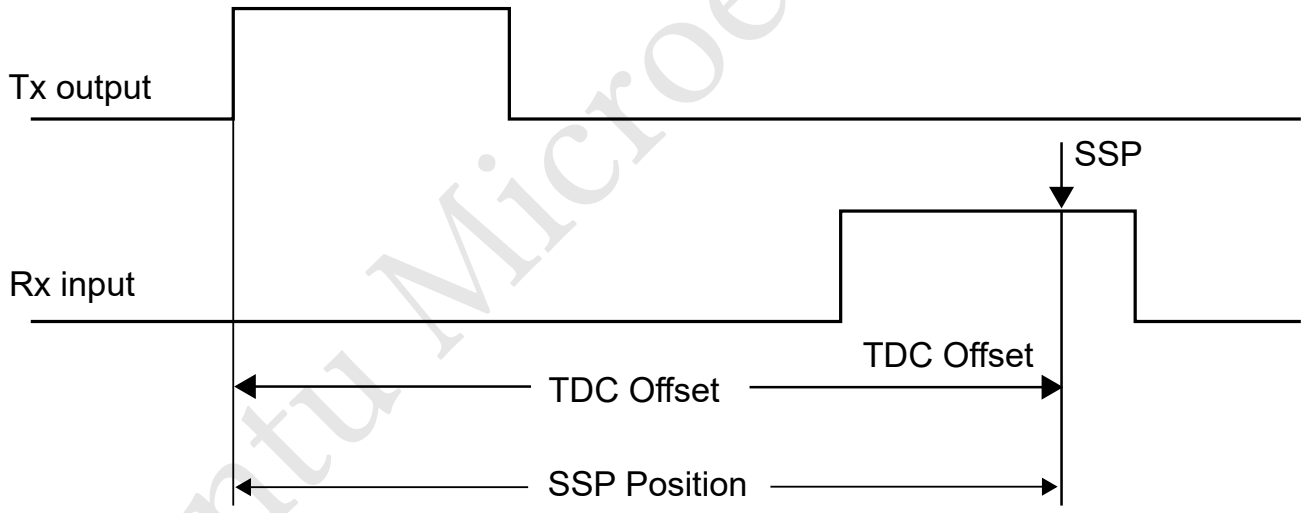


Figure 20.17: SSP Position with TDC Measurement Disabled by ETDCEN and TDMDIS

During the data phase of CAN FD frames with bit rate switching enabled, at the onset of every Tx CAN bit, the transmitted Tx bit value is temporarily stored in a buffer and a time countdown based on TDCVAL is started which ends with the comparison of the received Rx bit (delayed by the external loop delay plus the specified offset) with the stored Tx bit. If a bit error is detected at the secondary sample point, the FlexCAN issues an error flag to the CAN bus at the next sample point.

During the arbitration phase, the delay compensation is always disabled. The maximum delay that can be compensated by the FlexCAN's transceiver delay compensation during the data phase is 3 CAN bit times - 2 Tq. Beyond this limit, the CAN_FDCTRL.TDCFAIL flag is set to indicate when the Transceiver Delay Compensation mechanism is out of range and unable to compensate the transceiver loop delay.

Yuntu Microelectronics

20.4.11.3 Remote Frame Transmission

The CPU can program a mailbox to be a remote request frame by configuring the mailbox as transmit with the RTR bit set to '1'. After the remote request frame is transmitted successfully, the mailbox becomes a receive message buffer, with the same ID as before.

The FlexCAN receives the resulting answer frame into the same message buffer that transmitted the remote frame. Filtering for that message buffer must not consider the RTR bit when comparing the incoming answer frame. The answer frame will have RTR = 0, whereas the remote frame is configured with RTR = 1. To exclude the RTR bit when filtering the incoming answer frame, make sure that either or both of the following bits are 0:

- The RTR bit of the respective mask register, either individual mask register RXIMRn[31] or global mask register RXMGMASK[31], depending on the value of CAN_MCR[IRMQ]. RXIMRn[31] disables RTR comparison for message buffer n (effective when CAN_MCR[IRMQ] = 1). RXMGMASK[31] disables RTR comparison for all message buffers (effective when CAN_MCR[IRMQ] = 0).
- CAN_CTRL2.EACEN = 0 globally disables comparing the RTR bit for incoming frames, regardless of a message buffer's RTR mask bit.

20.4.11.4 Remote Frame Reception

When a remote request frame is received by the FlexCAN, it can be treated in one of four ways, depending on the Remote Request Storing (CAN_CTRL2[RRS]), Legacy Rx FIFO Enable (CAN_MCR.RFEN), and Enhanced Rx FIFO Enable (CAN_MCR.ERFEN) bits:

- If RRS is negated, the frame's ID is compared to the IDs of the transmit message buffers with the CODE field 1010. If there is a matching ID, then this mailbox frame is transmitted. Note that if the matching mailbox has the RTR bit set, then the FlexCAN transmits a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame is received and matches a mailbox, this message buffer immediately enters the internal arbitration process, but is considered as a normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.
- If RRS is asserted, the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0100, 0010, or 0110. If there is a matching ID, then this mailbox stores the remote frame in the same fashion of a data frame. No automatic remote response frame is generated. The mask registers are used in the matching process.
- If RFEN is asserted, the FlexCAN does not generate an automatic response for remote request frames that match the Legacy Rx FIFO filtering criteria. If the remote frame matches one of the target IDs, it is stored in the Legacy Rx FIFO and presented to the CPU. Note that for filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID). Remote request frames are considered as normal frames, and generate a Legacy Rx FIFO overflow when a successful reception occurs and the Legacy Rx FIFO is already full.
- If ERFEN is set, the FlexCAN does not generate an automatic response for remote request frames that match the Enhanced Rx FIFO filtering criteria. Remote Request Frames are considered normal frames and generate an Enhanced Rx FIFO overflow when a successful reception occurs and the Enhanced Rx FIFO is already full.

NOTE: There is no remote frame in the CAN FD format. The RTR bit is replaced by a fixed dominant

Yuntu Microelectronics

RRS bit. The FlexCAN receives and transmits remote frames in the Classical CAN format.

20.4.11.5 Overload Frames

The FlexCAN does transmit overload frames when the following conditions are detected on the CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission.
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames).
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter.

20.4.11.6 Message Buffer Time Stamp

The value of the Free Running Timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of move-in in the TIME STAMP field of a message buffer, providing network behavior with respect to time.

When CAN_CTRL2.TIMER_SRC is asserted, the Free Running Timer is continuously clocked by an external time tick.

When CAN_CTRL2.TIMER_SRC is negated, the Free Running Timer is clocked by the FlexCAN bit-clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted.

When there is no message on the bus, it counts using the previously programmed baud rate.

The Free Running Timer is not incremented during Module Disable, Doze, Deepsleep, and Freeze modes. It can be reset upon a specific frame reception, enabling network time synchronization. See the TSYN description in ”Control 1 Register (CAN_CTRL1)”.

Alternatively, if the high-resolution time stamp feature is implemented (TIME_STAMP_EN = 1), the time stamp of the message buffer can capture either the upper or lower 16 bits of the 32-bit time base (timestamp_base[31:0]), as controlled by CAN_CTRL2.MBTSBASE.

20.4.11.7 High-Resolution Time Stamp

The high-resolution time stamp feature, implemented when configuration parameter TIME_STAMP_EN = 1, uses a dedicated timer with a 32-bit counter operating in freerunning mode.

The high-resolution time stamp is enabled by setting CAN_CTRL2.TSTAMPCAP to a nonzero value. When enabled, the dedicated 32-bit counter value at the timestamp_base[31:0] input is captured during a valid CAN frame and stored in the CAN_HR_TIME_STAMPn register.

Each CAN_HR_TIME_STAMPn register corresponds to a specific message buffer. For example, CAN_HR_TIME_STAMP0 stores the 32-bit time stamp associated with message buffer 0; HR_TIME_STAMP1 stores the 32-bit time stamp associated with message buffer 1; and so on.

The counter value is captured according to the CAN_CTRL2.TSTAMPCAP value. For classical CAN frames, the capture point can be the Start of Frame bit or the point in time at which a CAN frame is considered valid, which is the seventh bit of End of Frame for transmission and the sixth bit of End of Frame for reception. For CAN FD frames, the capture points can be the Start of Frame, the point in time at which a CAN FD frame is considered valid, or the res bit of a CAN FD frame.

The 16-bit TIME STAMP field of the message buffer can be configured to capture the lower or higher 16 bits of the high-resolution timer, as controlled by CAN_CTRL2.MBTSBASE.

Yuntu Microelectronics

20.4.11.8 Protocol Timing

Figure: ”CAN engine clocking scheme” shows the structure of the clock generation circuitry that feeds the PE submodule. The clock source bit (CAN_CTRL1.CLKSRC) defines whether the internal clock is connected to the output of a crystal oscillator (Oscillator Clock) or to the Peripheral Clock.

To ensure reliable operation, the clock source should be selected while the FlexCAN is in Module Disable mode (CAN_MCR.MDIS bit is asserted).

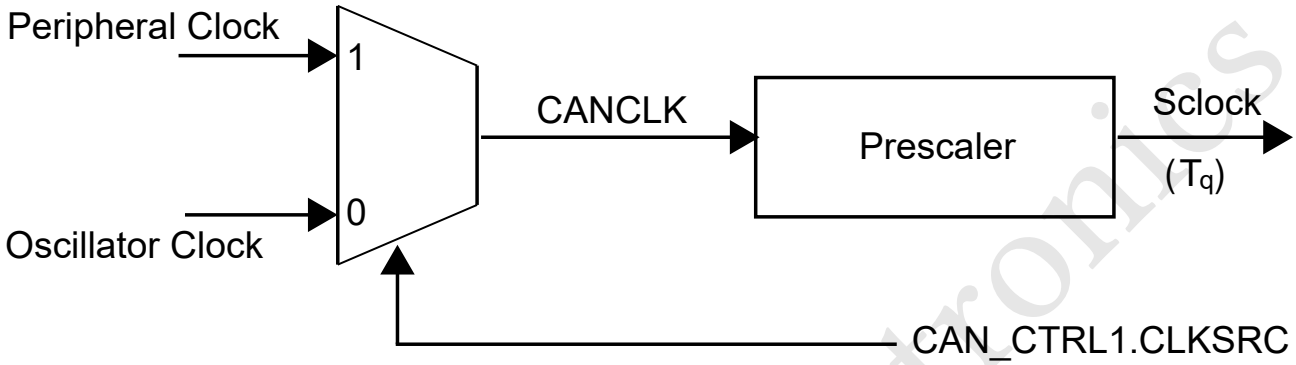


Figure 20.18: CAN Engine Clocking Scheme

The oscillator clock should be selected whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The crystal oscillator clock has better jitter performance than the peripheral clock.

The FlexCAN module supports a variety of means to set up bit timing parameters that are required by the CAN protocol. The Control 1 Register (CAN_CTRL1) has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2, and RJW.

The CAN Bit Timing register (CAN_CBT) extends the range of the CAN bit timing variables in CAN_CTRL1. The CAN FD Bit Timing register (CAN_FDCBT) provides a second set of CAN bit timing variables to be applied at the data phase of CAN FD frames with the Bit Rate Switch (BRS) set.

The Enhanced Nominal CAN Bit Timing register (CAN_ENCBT) extends the range of CAN bit timing variables in CAN_CBT. The Enhanced Data Phase CAN Bit Timing register (CAN_EDCBT) extends the range of CAN bit timing variables in CAN_FDCBT. When using CAN_ENCBT and CAN_EDCBT, the bit timing nominal and data phase serial clock (Sclock) dividers must be programed in the Enhanced CAN Bit Timing Prescalers register (CAN_EPRS).

NOTE: When the CAN FD feature is enabled, always set CAN_CBT.BTF and configure the CAN bit timing variables in CAN_CBT. See or. When the CAN FD feature is enabled, always set CAN_CBT.BTF or CAN_CTRL2.BTE and configure the CAN bit timing variables in CAN_CBT CAN_ENCBT. See ”CAN FD Bit Timing Register (CAN_FDCBT)” or ”Enhanced Nominal CAN Bit Timing Register (CAN_ENCBT)”.

The PRESDIV field (as well as its extended range EPRESDIV (or ENPRESDIV) and FDPRESDIV (or EDPRESDIV) for the data phase bits of CAN FD messages) defines the Prescaler Value (see the equation below) that generates the Serial Clock (Sclock), whose period defines the time quantum used to compose the CAN waveform. A time quantum (Tq) is the atomic unit of time handled by the CAN engine.

$$T_q = \frac{(PRESDIV + 1)}{f_{CANCLK}}$$

Yuntu Microelectronics

The bit rate, which defines the rate at which the CAN message is either received or transmitted, is given by the formula:

CAN Bit Time = (Number of Time Quanta in 1 bit time) * Tq

$$BitRate = \frac{1}{CANBitTime}$$

A bit time is subdivided into three segments:

- SYNC_SEG – Has a fixed length of one time quantum. Signal edges are expected to occur within this segment.
- Time Segment 1 – Includes the Propagation Segment and the Phase Segment 1 of the CAN standard. Time Segment 1 can be programmed by setting the PROPSEG and the PSEG1 fields of the CAN_CTRL1 register so that their sum (plus 2) is in the range of 2 to 16 time quanta. When CAN_CBT.BTF is asserted, the FlexCAN uses the EPROPSEG and EPSEG1 fields from the CAN_CBT register so that their sum (plus 2) is in the range of 2 to 96 time quanta. For messages in CAN FD format with the BRS bit set, the FlexCAN uses FDPROPSEG and FDPSEG1 from CAN_FDCBT instead, so that their sum (plus 1) is in the range of 2 to 39 time quanta.

If CAN_CTRL2.BTE is set, the FlexCAN uses CAN_ENCBT.NTSEG1 to configure time segment 1 in the range of 2 to 256 time quanta. For the data phase in CAN FD messages with BRS set, CAN_EDCBT.DTSEG1 must be used for configuring time segment 1 in the range of 2 to 32 time quanta.

- Time Segment 2 – Represents the Phase Segment 2 of the CAN standard. It can be programmed by setting the PSEG2 field of the CAN_CTRL1 register (plus 1) to be 2 to 8 time quanta long. When CAN_CBT.BTF is asserted, the FlexCAN uses the EPSEG2 field of CAN_CBT register so that its value (plus 1) is in the range of 2 to 32 time quanta. For messages in CAN FD format with the BRS bit set, the FlexCAN uses FDPSEG2 from CAN_FDCBT instead, so that its value (plus 1) is in the range of 2 to 8 time quanta. The Time Segment 2 cannot be smaller than the Information Processing Time (IPT), which is 2 time quanta in the FlexCAN.

If CAN_CTRL2.BTE is set, the FlexCAN uses CAN_ENCBT.NTSEG2 to configure time segment 2 in the range of 2 to 128 time quanta. For the data phase in CAN FD messages with BRS set, CAN_EDCBT.DTSEG2 must be used for configuring time segment 2 in the range of 2 to 16 time quanta.

see also ”Segments within the bit time (example using CAN_ CBT and CAN_FDCBT bit timing variables for CAN FD format)”, ”Segments within the bit time (example using CAN_CTRL1 bit timing variables for classical CAN format)” and ”CAN time segment syntax”.

NOTE: For further explanation of the underlying concepts, see ISO 11898-1. See also the CAN 2.0A/B protocol specification for bit timing.

NOTE: The bit time defined by the above time segments must not be smaller than 5 time quanta. For bit time calculations, use an Information Processing Time (IPT) of 2, which is the value implemented in the FlexCAN module.

Yuntu Microelectronics

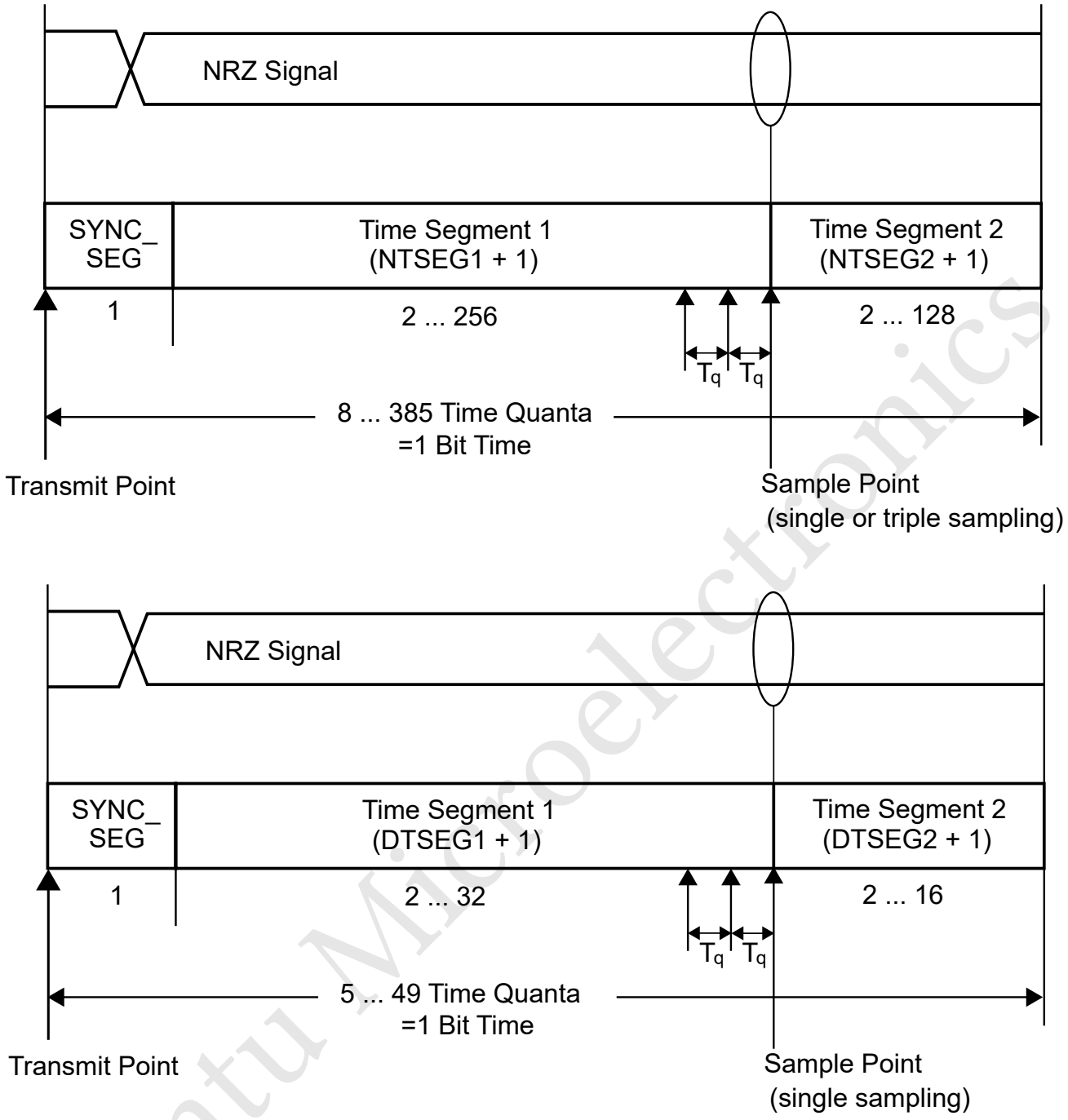


Figure 20.19: Segments Within the Bit Time Example Using CAN ENCBT and CAN EDCBT Bit Timing Variables

Yuntu Microelectronics

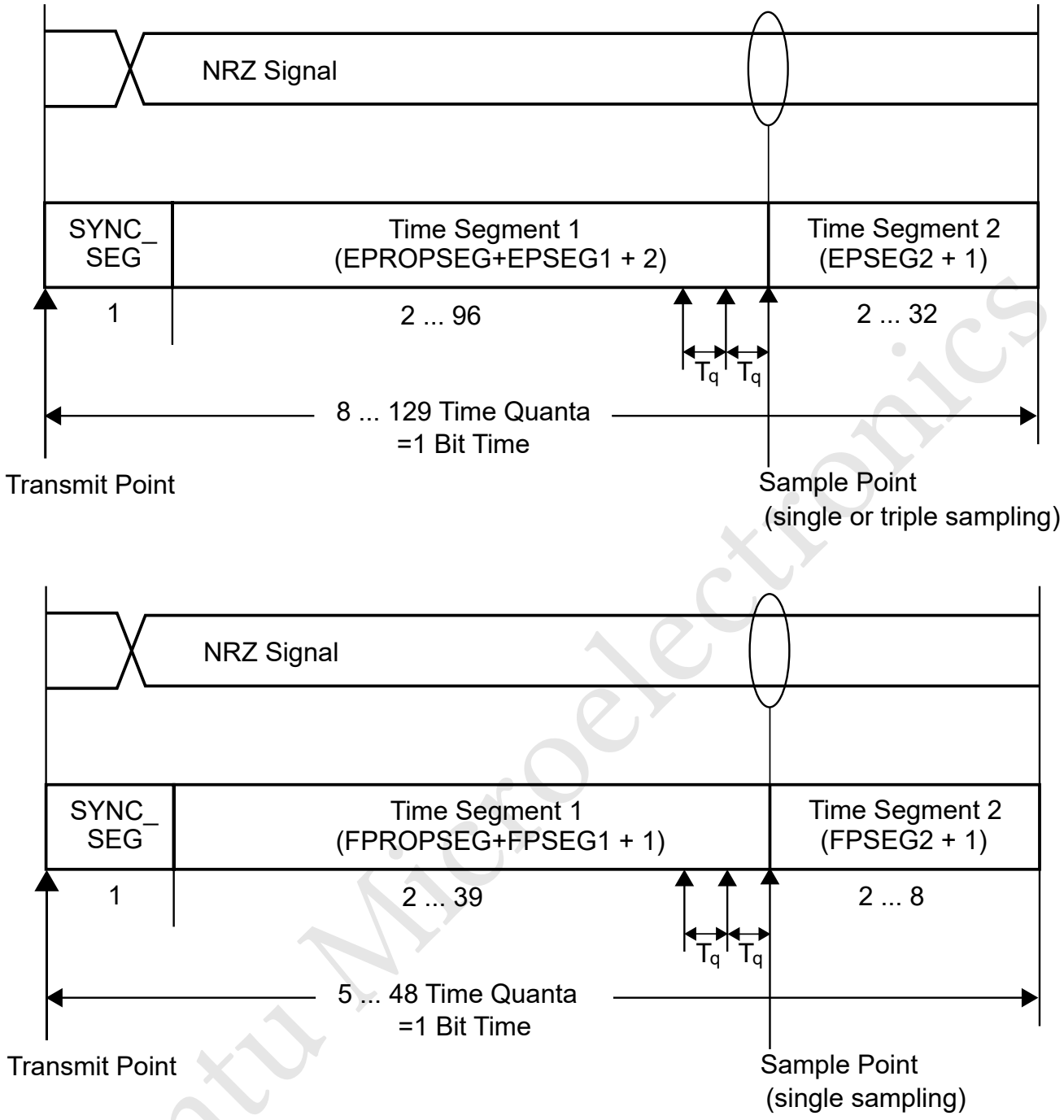


Figure 20.20: Segments Within the Bit Time Example Using CAN CBT and CAN FDCBT Bit Timing Variables for CAN FD Format

Yuntu Microelectronics

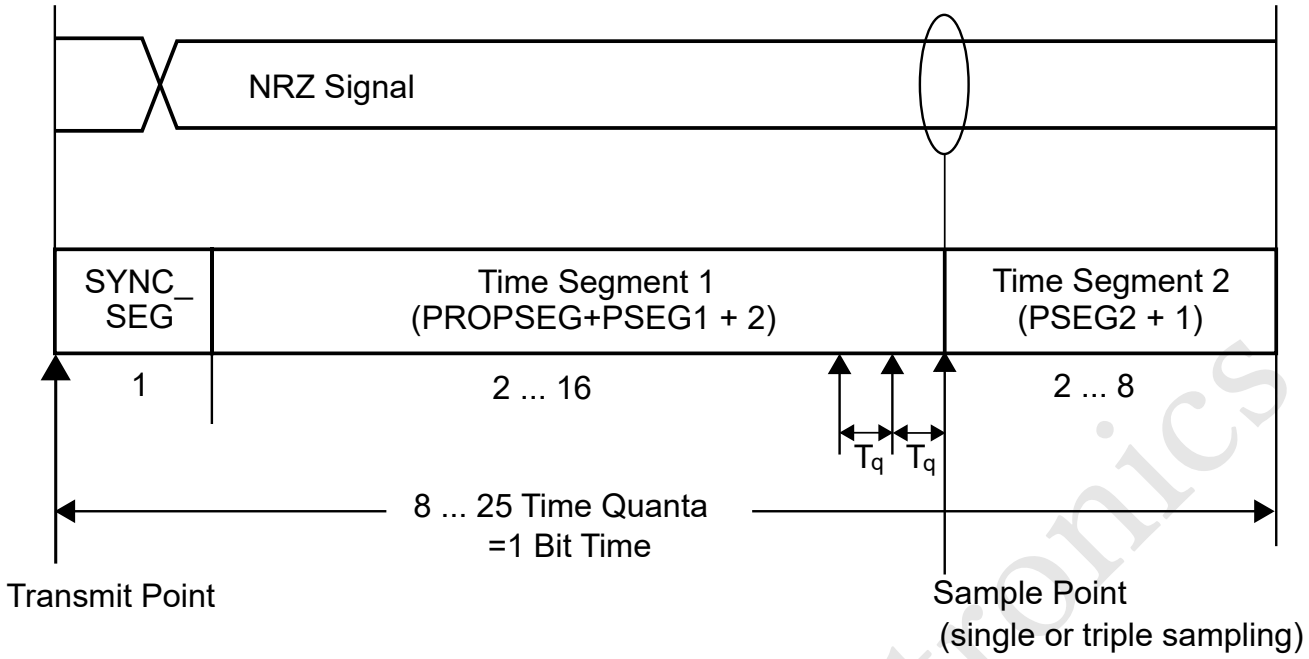


Figure 20.21: Segments Within the Bit Time Example Using CAN CTRL1 Bit Timing Variables for Classical CAN Format

Table 20.92: CAN Time Segment Syntax

Syntax	Description
SYNC_SEG	System expects transitions to occur on the bus during this period
TSEG1	Corresponds to the sum of PROPSEG and PSEG1
TSEG2	Corresponds to the PSEG2 value
Transmit Point	A node in transmit mode transfers a new value to the CAN bus at this point.
Sample Point	A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample.

Table: ”CAN BOSCH CAN 2P0B standard complicant bit segment settings” gives some examples of the CAN compliant segment settings for Classical CAN format (Bosch CAN 2.0B) (non-FD) messages.

Table 20.93: CAN BOSCH CAN 2.0B Standard Complicant Bit Segment Settings

Time Segment 1	Time Segment 2	Re-Synchronization Jump Width
5 … 10	2	1 … 2
4 … 11	3	1 … 3
5 … 12	4	1 … 4
6 … 13	5	1 … 4
7 … 14	6	1 … 4
8 … 15	7	1 … 4
9 … 16	8	1 … 4

Yuntu Microelectronics

NOTE: The user must ensure the bit time settings are in compliance with the CAN Protocol standard (ISO 11898-1).

Whenever CAN bit is used as a measure of time duration (for example, estimating the occurrence of a CAN bit event in a message), the number of peripheral clocks in one CAN bit (NumClkBit) can be calculated as:

$$NumClkBit = (\frac{f_{SYS}}{f_{CANCLK}}) \times (PRESDIV + 1) \times (PROPSEG + PSEG1 + PSEG2 + 4)$$

or, if CAN_CTRL2.BTE = 1:

$$NumClkBit = (\frac{f_{SYS}}{f_{CANCLK}}) \times (ENPRESDIV + 1) \times (NTSEG1 + NTSEG2 + 3)$$

where:

- NumClkBit is the number of peripheral clocks in one CAN bit.
- fCANCLK is the PE clock in Hz (see Figure: ”CAN engine clocking scheme”).
- fSYS is the frequency of operation of the system (CHI) clock in Hz.
- PSEG1 is the CAN_CTRL1.PSEG1 value.
- PSEG2 is the CAN_CTRL1.PSEG2 value.
- PROPSEG is the CAN_CTRL1.PROPSEG value.
- PRESDIV is the CAN_CTRL1.PRESDIV value.
- ENPRESDIV is the CAN_ENPRS.ENPRESDIV value.
- NTSEG1 is the CAN_ENCBT.NTSEG1 value.
- NTSEG2 is the CAN_ENCBT.NTSEG2 value.

The formula above is also applicable to the alternative CAN bit timing variables described in the CAN Bit Timing Register (CAN_CBT) and the CAN FD Bit Timing Register (CAN_FDCBT).

For example, 180 CAN bits = (180 x NumClkBit) peripheral clock periods.

20.4.11.9 Arbitration and Matching Timing

During normal reception and transmission, the matching, arbitration, move-in, and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

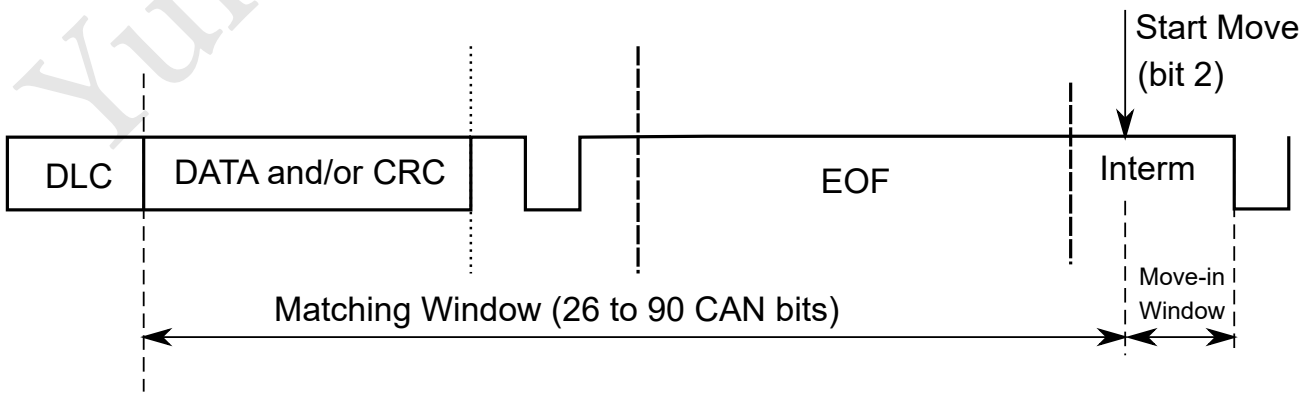


Figure 20.22: Matching and Move-in Time Windows

Yuntu Microelectronics

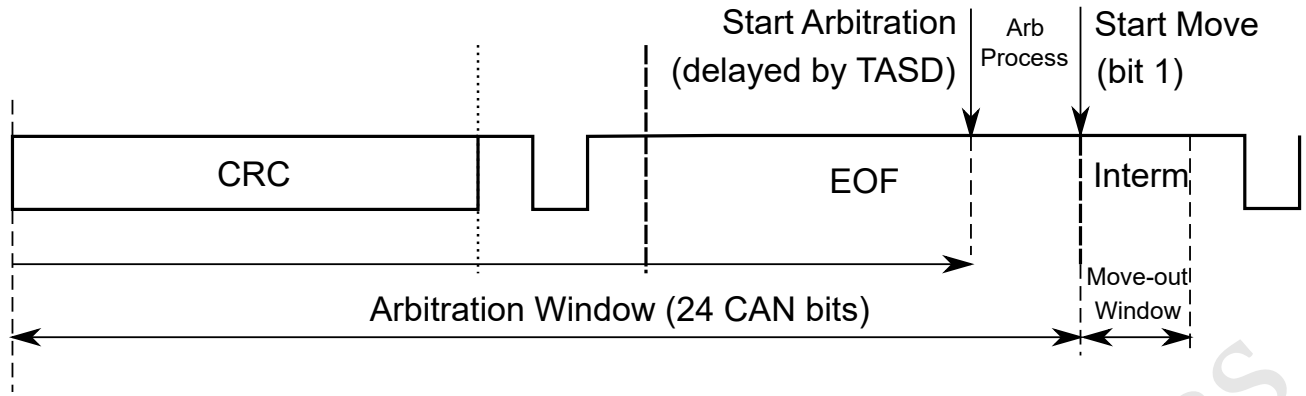


Figure 20.23: Arbitration and Move-out Time Windows

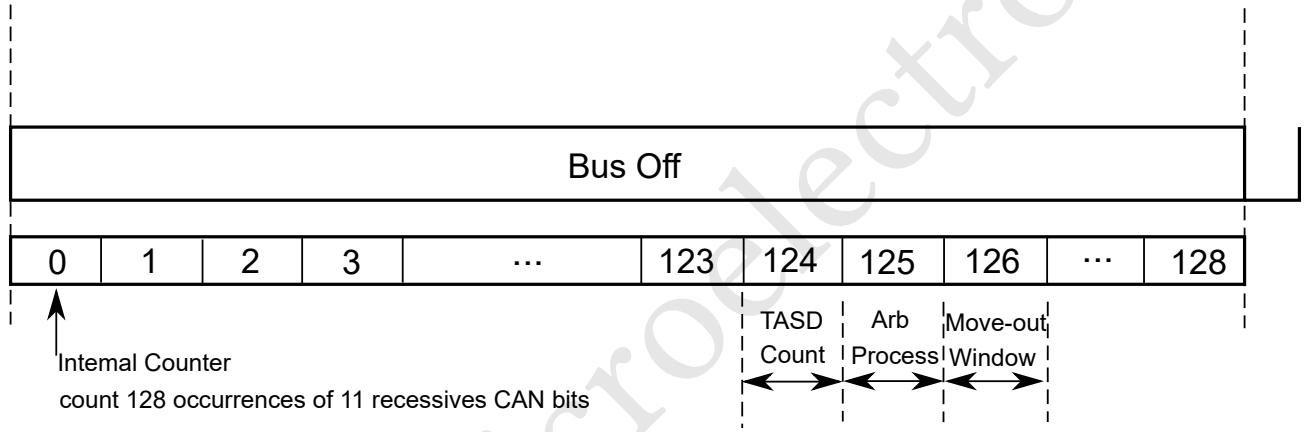


Figure 20.24: Arbitration at the End of Bus Off and Move-out Time Window

NOTE: In the preceding figures, the matching and arbitration timing does not take into account the delay caused by the concurrent memory access due to the CPU or other internal FlexCAN submodules.

20.4.11.10 Tx Arbitration Start Delay

The Tx Arbitration Start Delay (TASD) bit field in Control 2 register (CAN_CTRL2.TASD) is a variable that indicates the number of CAN bits used by the FlexCAN to delay the Tx arbitration process start point from the first bit of the CRC field of the current frame.

CAN_CTRL2.TASD can be written only in Freeze mode because it is blocked by hardware in other modes.

The transmission performance is impacted by the ability of the CPU to reconfigure MBs for transmission after the end of the internal arbitration process, in which the FlexCAN finds the winner MB for transmission (see ”Arbitration Process”). If the arbitration ends too early before the first bit of the Intermission field, then there is a chance that the CPU reconfigures some Tx MBs and the winner MB is no longer the best candidate to be transmitted.

TASD is useful to optimize the transmission performance by defining the arbitration start point, as shown in the next figure, based on factors such as:

Yuntu Microelectronics

- The peripheral-to-oscillator clock ratio
- CAN bit timing variables that determine the CAN bit rate
- The number of MBs in use by the matching and arbitration processes

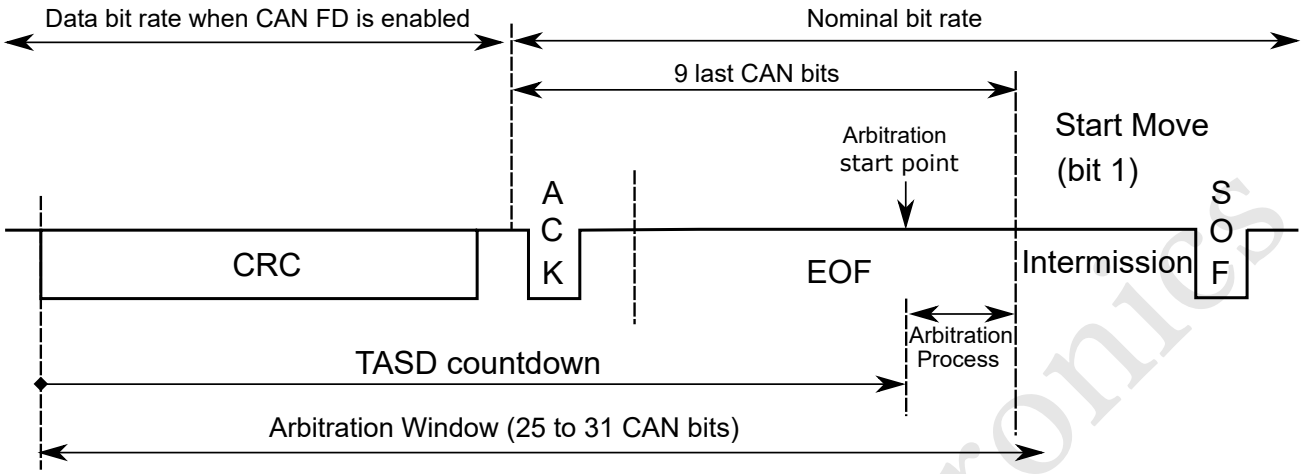


Figure 20.25: Optional TX Arbitration Start Point

The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and to the CAN bit rate, and inversely proportional to the peripheral clock frequency.

The optimal arbitration timing is that in which the last MB is scanned right before the first bit of the Intermission field of a CAN frame. For instance, if there are few MBs and the peripheral/oscillator clock ratio is high and the CAN baud rate is low, then the arbitration can be placed closer to the frame's end, adding more delay to its start point, and vice-versa.

If TASD is set to 0, the arbitration start is not delayed and more time is reserved for arbitration. On the other hand, if TASD is close to 24, then the CPU can configure a Tx MB later and less time is reserved for arbitration. If too little time is reserved for arbitration, the FlexCAN may be not be able to find a winner MB in time to be transmitted with the best chance to win the bus arbitration against external nodes on the CAN bus.

The optimal TASD value can be calculated as follows:

For CAN FD frames and $(MAXMB + 1) \leqslant NMB_{END}$

$$TASD = 31 - \frac{2 * (MAXMB + 1) + 4}{CPCN_N}$$

For CAN FD frames and $(MAXMB + 1) > NMB_{END}$

$$TASD = 22 - \frac{2 * (MAXMB + 1) - NMB_{END}}{CPCB_F}$$

For non-FD frames

$$TASD = 25 - \frac{2 * (MAXMB + 1) + 4}{CPCB}$$

Yuntu Microelectronics

where:

$$NMB_{END} = \frac{9 * CPCN_N - 4}{2}$$

$$BITRATE_F = \frac{f_{CANCLK}}{[1 + (FPSEG1 + 1) + (FPSEG2 + 1) + FPROPSEG] * (FPRESDIV + 1)}$$

$$BITRATE_N = \frac{f_{CANCLK}}{[1 + (EPSEG1 + 1) + (EPSEG2 + 1) + (EPROPSEG + 1)] * (EPRESDIV + 1)}$$

$$CPCB_N = \frac{f_{SYS}}{BITRATE_N}$$

$$CPCB_F = \frac{f_{SYS}}{BITRATE_F}$$

$$CPCB = CPCB_N$$

- MAXMB is the CAN_MCR.MAXMB value.
- NMBEND is the number of MBs that can be scanned by the arbitration process during the 9 last CAN bits at the end of a frame, see the figure above.
- BITRATEN is the CAN bit rate in bits per second calculated by the nominal CAN bit time variables.
- BITRATEF is the CAN bit rate in bits per second calculated by the data CAN bit time variables.
- CPCBN is the number of peripheral clocks per CAN bit in nominal bit rate for CAN FD frames.
- CPCBF is the number of peripheral clocks per CAN bit in data bit rate for CAN FD frames.
- CPCB is the number of peripheral clocks per CAN bit for non-FD frames.
- fCANCLK is the oscillator clock, in Hz.
- fSYS is the peripheral clock, in Hz.
- EPSEG1 is the CAN_CBT.EPSEG1 value. (CAN_CTRL1.PSEG1 can also be used.)
- EPSEG2 is the CAN_CBT.EPSEG2 value. (CAN_CTRL1.PSEG2 can also be used.)
- EPROPSEG is the CAN_CBT.EPROPSEG value. (CAN_CTRL1.PROPSEG can also be used.)
- EPRESDIV is the CAN_CBT.EPRESDIV value. (CAN_CTRL1.PRESDIV can also be used.)
- FPSEG1 is the CAN_FDCBT.FPSEG1 value.
- FPSEG2 is the CAN_FDCBT.FPSEG2 value.
- FPROPSEG is the CAN_FDCBT.FPROPSEG value.
- FPRESDIV is the CAN_FDCBT.FPRESDIV value.

If CAN_CTRL2.BTE = 1:

$$BITRATE_N = \frac{f_{CANCLK}}{[1 + (NTSEG1 + 1) + (NTSEG2 + 1)] * (ENPRESDIV + 1)}$$

See "Protocol Timing" for additional details.

The following tables give the TASD values calculated for some configuration cases.

Case 1:

Yuntu Microelectronics

- Clock ratio = 2:1 (example: peripheral clock 80 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 20.94: CAN TASD Values for Case1 Example

Number of MBs	TASD Value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	24	8.0
64	23	8.0
96	22	8.0
100	22	8.0
128	21	3.6

Case 2:

- Clock ratio = 1:1 (example: peripheral clock 40 MHz and oscillator clock 40 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 20.95: CAN TASD Values for Case2 Example

Number of MBs	TASD Value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	23	6.67
54	22	5.0
64	21	3.33
96	20	1.6
128	18	1.05

Case 3:

- Clock ratio = 2:1 (example: peripheral clock 40 MHz and oscillator clock 20 MHz)
- Bit rate in arbitration phase = 1 Mbaud

Table 20.96: CAN TASD Values for Case3 Example

Number of MBs	TASD Value	Maximum Bit Rate in Data Phase (Mbaud)
16	24	Invalid
32	23	4.0
54	22	4.0
64	21	3.33
96	20	1.54
128	18	1.05

20.4.12 Clock Domains and Restrictions

The FlexCAN module has two clock domains asynchronous to each other:

Yuntu Microelectronics

- The Bus Domain feeds the CHI submodule and is derived from the peripheral clock.
- The Oscillator Domain feeds the PE submodule and is derived directly from a crystal oscillator clock, so that very low jitter performance can be achieved on the CAN bus.

When CAN_CTRL1.CLKSRC is set, synchronous operation occurs because both domains are connected to the peripheral clock (creating a 1:1 ratio between the peripheral and oscillator clocks).

When the two domains are connected to clocks with different frequencies and/or phases, there are restrictions on the frequency relationship between the two clock domains. In the case of asynchronous operation, the Bus Domain clock frequency must always be greater than the Oscillator Domain clock frequency.

NOTE: Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

When doing matching and arbitration, the FlexCAN needs to scan the entire MB memory during the time slot of one CAN frame, comprising a number of CAN bits. In order to have sufficient time to do that, the following requirements must be observed:

- The peripheral clock frequency can not be lower than the oscillator clock frequency.
- There must be a minimum number of peripheral clocks per CAN bit, see below table:

Table 20.97: CAN Minimum Number of Peripheral Clocks Per CAN Bit for Classical CAN Format

Number of Mailboxes	Value of CAN_MCR.RFEN Bit	Minimum Number of Peripheral Clocks per CAN Bit
16	0	16
32	0	16
16	1	16
32	1	17

For Classical CAN frame format, the minimum number of peripheral clocks per CAN bit specified in ”CAN minimum number of peripheral clocks per CAN bit for classical CAN format” determines the minimum peripheral clock frequency for a given number of mailboxes and for an expected CAN bit rate. The CAN bit rate depends on the number of time quanta in a CAN bit, which can be defined by adjusting one or more of the bit timing values contained in either the CAN_CTRL1 register or the CAN_CBT register or the CAN_ENCBT register. The time quantum (Tq) is defined in ”Protocol Timing”. The minimum number of time quanta per CAN bit must be 8; therefore, the oscillator clock frequency should be at least 8 times the CAN bit rate.

For CAN FD frame format, there are some constraints that need to be satisfied. The number of peripheral clocks per CAN bit in nominal bit rate (NumClkNomBit) can be calculated by the following equation:

$$NumClkNomBit = (\frac{f_{SYS}}{f_{CANCLK}}) \times (PRESDIV + 1) \times (PROPSEG + PSEG1 + PSEG2 + 4) = \frac{f_{SYS}}{NomBitRate}$$

where PRESDIV, PSEG1 and PSEG2 are CAN bit time values in the CAN_CTRL1 register. Alternatively, the EPRESDIV, EPSEG1 and EPSEG2 values in the CAN_CBT register or the CAN_EPRS.ENPRESDIV, CAN_ENCBT.NTSEG1, and CAN_ENCBT.NTSEG2 values can be used instead. NumClkNomBit can also be calculated as a function of the expected nominal bit rate used in the Arbitration Phase (NomBitRate) as shown in the equation above.

Yuntu Microelectronics

The number of CAN bits in the Data Phase of a FD Frames with the BRS bit set (fast CAN bits) depends on the number of data bytes in the payload. The number of fast CAN bits (NumOfFastBits) can be determined as shown in Table: "CAN number of fast CAN bits in a CAN FD frame". The fewer the number of data bytes, the fewer the number of fast CAN bits, and less time is available for the FlexCAN to scan the whole MB memory during the internal matching and arbitration processes.

Table 20.98: CAN Number of Fast CAN Bits in a CAN FD Frame

Minimum Number of Data Bytes	DLC Field	NumOfFastBits
0	0x0	21
1	0x1	29
2	0x2	37
3	0x3	45
4	0x4	53
5	0x5	61
6	0x6	69
7	0x7	77
8	0x8	85
12	0x9	117
16	0xA	149
20	0xB	186
24	0xC	218
32	0xD	282
48	0xE	410
64	0xF	538

The critical part of a CAN FD frame is during the Data Phase, where the CAN bit rate is faster than in the Arbitration Phase. The minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated to guarantee that enough time is available for the FlexCAN to scan the MB memory during reception and transmission. The following equation below calculates this constraint:

$$MinNumClkFastBitA = \frac{(8.5 * MaxNumOfMb) + (ERFEN * (2 * NFE + 4)) + 64 - (9 * NumClkNomBit)}{NumOfFastBits}$$

where MaxNumOfMb is the maximum number of available mailboxes defined in CAN_MCR.MAXMB. ERFEN and NFE are fields in CAN_ERFCR.

The clock domain crossing circuit between the CHI and PE submodules also imposes a minimum number of peripheral clocks per fast CAN bit for the handshake mechanism to work properly without losing status information through the interface, as shown in the following equation:

$$MinNumClkFastBit_B = 3 * (1 + \frac{f_{SYS}}{f_{CANCLK}})$$

Therefore, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) is determined by the larger of the two values calculated above:

MinNumClkFastBit = Maximum (MinNumClkFastBitA, MinNumClkFastBitB)

Yuntu Microelectronics

Then, the maximum CAN bit rate in the Data Phase of CAN FD frames (DataBitRateMAX) can be calculated as follows:

$$DataBitRate_{MAX} = \frac{f_{CANCLK}}{ROUNDUP(\frac{MinNumClkFastBit * f_{CANCLK}}{f_{SYS}})}$$

The peripheral and oscillator clock frequencies, the maximum number of mailboxes, and the expected nominal bit rate affect the maximum data bit rate attainable by the FlexCAN in CAN FD mode. In addition, the data bit rate depends on the minimum payload size of FD frames used in a given application.

To illustrate how the CAN FD bit rate is affected by the configuration of FlexCAN variables, an application example with the peripheral and oscillator clock frequencies set to 50 MHz and 40 MHz, respectively, is considered.

Step 1 - Considering the nominal bit rate as 1 Mbps, the number of peripheral clocks per CAN bit in nominal bit rate is calculated as follows:

$$NumClkNomBit = \frac{50 * 10^6}{1 * 10^6} = 50$$

Step 2 - The number of fast CAN bits (NumOfFastBits) is determined as shown in table "CAN number of fast CAN bits in a CAN FD frame". For example, if the minimum payload in FD frames is 8 bytes, then there are 85 CAN bits in the Data Phase.

Step 3 - Assuming the maximum number of mailboxes is 96 and the Enhanced Rx FIFO feature is disabled, the minimum number of peripheral clocks per fast CAN bit (MinNumClkFastBit) can be calculated as follows:

$$MinNumClkFastBit_A = \frac{(8.5 * 96) + 64 - (9 * 50)}{85} = 5.06$$

$$MinNumClkFastBit_B = 3 * (1 + \frac{50}{40}) = 6.75$$

$$MinNumClkFastBit = Maximum(5.06, 6.75) = 6.75$$

Step 4 - The maximum CAN bit rate in the Data Phase can then be calculated as follows:

$$DataBitRate_{MAX} = \frac{40 * 10^6}{ROUNDUP(\frac{6.75 * 40 * 10^6}{50 * 10^6})} = 6.667 Mbps$$

As demonstrated in this example, even though the oscillator clock frequency (40 MHz) is adequate to generate a data rate of 8 Mbps in CAN FD mode, the specific FlexCAN configuration limits this rate to 6.667 Mbps. This limitation is mainly due to the low peripheral clock frequency that imposes the MinNumClkFastBitB limit.

Below table shows the maximum data rate for CAN FD according to clock frequencies, payload size, and number of available mailboxes. As shown in below table, for some cases, if the number of available mailboxes is reduced, the FlexCAN can then achieve a data rate up to 8 Mbps.

Yuntu Microelectronics

Table 20.99: CAN Maximum CAN Bit Rate in Data Phase of CAN FD Frames

Peripheral Clock Frequency (MHz)	Payload Size	Number of Available Mailboxes	Maximum Data Rate (Mbps)
40	8	94	6.667
40	8	114	5.0
40	12	117	6.667
40	12	128	5.714
50	12 to 64	128	6.667
60	8	126	8.0
60	12	128	8.0
67	6	128	8.0
80	3	128	8.0
100	0	128	8.0

20.4.13 Modes of Operation Details

The FlexCAN module has functional modes and low-power mode. See "Modes of Operation" for an introductory description of all the modes of operation. The following subsections contain functional details on Freeze mode and the low-power mode.

20.4.13.1 Freeze Mode

Freeze mode is requested either by the CPU through the assertion of the CAN_MCR.HALT bit or when the chip is put into Debug mode. In both cases, it is also necessary that the CAN_MCR.FRZ bit is asserted and the FlexCAN is not in a low-power mode. Freeze mode is also requested by FlexCAN through the automatic assertion of both HALT and FRZ bits when the CAN_MECR.NCEFAFRZ bit is set and a non-correctable error is detected in a memory read access performed by FlexCAN internal processes (see "response to Errors").

The acknowledgement is obtained through the assertion by the FlexCAN of CAN_MCR.FRZACK bit. The CPU must only consider the FlexCAN to be in Freeze mode when both the request and acknowledge conditions are satisfied.

When Freeze mode is requested, the FlexCAN performs the following operations:

- Waits to be in either the Intermission, Passive Error, Bus Off, or Idle state.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in does not prevent going to Freeze mode.
- Ignores the Rx input pin and drives the Tx pin as recessive.
- Stops the prescaler, thus halting all CAN protocol activities.
- Grants write access to the Error Counters Register, which is read-only in other modes.
- Sets the NOTRDY and FRZACK bits in CAN_MCR.

After requesting Freeze mode, the user must wait for the CAN_MCR.FRZACK bit to be asserted before executing any other action. Otherwise, the FlexCAN may operate in an unpredictable way. In Freeze mode, all memory mapped registers are accessible, except for the CAN_CTRL1.CLKSRC bit, which can be read but cannot be written.

Exiting Freeze mode is done in one of the following ways:

- CPU negates the CAN_MCR.FRZ bit.

Yuntu Microelectronics

- The chip is removed from Debug Mode and/or the HALT bit is negated.

The CAN_MCR.FRZACK bit is negated after the PE recognizes the negation of the freeze request. When out of Freeze mode, the FlexCAN tries to re-synchronize to the CAN bus by waiting for 11 consecutive recessive bits.

20.4.13.2 Module Disable Mode

Module Disable is a low-power mode normally used to temporarily disable a complete FlexCAN block, with no power consumption. Module Disable is requested by the CPU through the assertion of the CAN_MCR.MDIS bit, and the acknowledged through the assertion by the FlexCAN of the CAN_MCR.LPMACK bit. The CPU must only consider the FlexCAN to be in Disable mode when both request and acknowledge conditions are satisfied.

If the FlexCAN is disabled during Freeze mode, it requests to disable the clocks to the PE and CHI submodules, sets the LPMACK bit, and negates the FRZACK bit.

It is not recommended to use Module Disable mode under Pretended Networking mode. Negate the MDIS bit and wait for LPMACK to negate before setting CAN_MCR.PNET_EN.

If disabled during transmission or reception, the FlexCAN module performs the following operations:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules.
- Sets the NOTRDY and LPMACK bits in CAN_MCR.

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register, and the Legacy Rx FIFO Global Mask Register. The Legacy Rx FIFO Information Register, the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the FlexCAN is in Module Disable mode. Exiting from Module Disable mode is done by negating the MDIS bit by the CPU, which causes the FlexCAN to request to resume the clocks and to negate the LPMACK bit after the CAN PE recognizes the negation of the Module Disable mode request by the CPU.

20.4.13.3 Doze Mode

Doze mode is a system low-power mode in which the CPU bus is kept alive and a global Doze mode request is sent to all peripherals asking them to enter low-power mode. When Doze mode is globally requested, the CAN_MCR.DOZE bit needs to have been asserted previously for Doze mode to be triggered. The FlexCAN acknowledges the request by asserting the CAN_MCR.LPMACK bit. The CPU must only consider the FlexCAN to be in Doze mode when both request and acknowledge conditions are satisfied.

If Doze mode is triggered during Freeze mode, the FlexCAN requests to shut down the clocks to the PE and CHI submodules, sets the LPMACK bit and, negates the FRZACK bit.

If Doze mode is triggered during transmission or reception, the FlexCAN performs the following operations:

Yuntu Microelectronics

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Shuts down the clocks to the PE and CHI submodules.
- Sets the NOTRDY and LPMACK bits in CAN_MCR.

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register, and the Legacy Rx FIFO Global Mask Register. The Legacy Rx FIFO Information Register, the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the FlexCAN is in Doze mode.

Exiting Doze mode is done in one of the following ways:

- CPU removing the Doze mode request.
- CPU negating the CAN_MCR.DOZE bit.
- Self Wake Up mechanism.

In the Self Wake Up mechanism, if the CAN_MCR.SLFWAK bit was set at the time that the FlexCAN entered Doze mode, then upon detection of a recessive-to-dominant transition on the CAN bus, the FlexCAN negates the DOZE bit, requests to resume its clocks, and negates LPMACK after the CAN PE recognizes the negation of the Doze mode request.

The FlexCAN also sets the CAN_ESR.WAKINT bit and, if enabled by the CAN_MCR.WAKMSK bit, generates a Wake Up interrupt to the CPU. The FlexCAN then waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, the FlexCAN does not receive the frame that woke it up. Table: "CAN Wake-up from Doze Mode" details the effect of SLFWAK and WAKMSK upon wake-up from Doze mode.

Table 20.100: CAN Wake-up from Doze Mode

SLFWAK	WAKINT	WAKMSK	FlexCAN Clocks Enabled	Wake-up Interrupt Generated
0	–	–	No	No
0	–	–	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	Yes	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line while in Doze mode. See the CAN_MCR.WAKSRC bit description.

This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference in noisy environments.

20.4.13.4 Deepsleep Mode

Deepsleep mode is a system low-power mode in which all chip clocks can be stopped for maximum power savings. Deepsleep mode is globally requested by the CPU and the acknowledged by the FlexCAN through the assertion of a Deepsleep Acknowledgement signal (ipg_deepsleep_ack). The CPU must only

Yuntu Microelectronics

consider the FlexCAN to be in Deepsleep mode when both request and acknowledge conditions are satisfied.

If the FlexCAN receives the global Deepsleep mode request during Freeze mode, it sets the LPMACK bit, negates the FRZACK bit, and then sends the Deepsleep Acknowledge signal to the CPU, in order to shut down the clocks globally.

If Deepsleep mode is requested during transmission or reception, the FlexCAN performs the following operations:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive.
- Waits for all internal activities like arbitration, matching, move-in, and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive.
- Sets the NOTRDY and LPMACK bits in CAN_MCR.
- Sends a Deepsleep Acknowledge signal to the CPU, so that it can shut down the clocks globally.

Deepsleep mode is exited when the CPU resumes the clocks and removes the Deepsleep mode request. This can be as a result of the Self Wake Up mechanism.

In the Self Wake Up mechanism, if the CAN_MCR.SLFWAK bit was set at the time that the FlexCAN entered Deepsleep mode, then upon detection of a recessive-to-dominant transition on the CAN bus, the FlexCAN sets the CAN_ESR1.WAKINT bit and, if enabled by the CAN_MCR.WAKMSK bit, generates a Wake Up interrupt to the CPU. Upon receiving the interrupt, the CPU should resume the clocks and remove the Deepsleep mode request. The FlexCAN then waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, the FlexCAN does not receive the frame that woke it up. Table: ”Wake-up From Deepsleep Mode” details the effect of SLFWAK and WAKMSK upon wake-up from Deepsleep mode. Note that Wake-Up from Deepsleep mode only works when both bits are asserted.

After the CAN PE recognizes the negation of the Deepsleep mode request, the FlexCAN negates the LPMACK bit. The FlexCAN then waits for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, the FlexCAN does not receive the frame that woke it up.

Table 20.101: CAN Wake-Up from Deepsleep Mode

SLFWAK	WAKINT	WAKMSK	Chip Clocks Enabled	Wake-up Interrupt Generated
0	–	–	No	No
0	–	–	No	No
1	0	0	No	No
1	0	1	No	No
1	1	0	No	No
1	1	1	Yes	Yes

The sensitivity to CAN bus activity can be modified by applying a low-pass filter function to the Rx CAN input line while in Deepsleep mode. See the CAN_MCR.WAKSRC bit description.

This feature can be used to protect FlexCAN from waking up due to short glitches on the CAN bus lines. Such glitches can result from electromagnetic interference in noisy environments.

Yuntu Microelectronics

20.4.13.5 Pretended Networking Mode

Pretended Networking is a special low-power mode used to receive wake up messages with low-power consumption. Pretended Networking mode can be selected to operate together with Doze mode or Deepsleep mode. Before entering in one of these low-power mode, the CAN_MCR.PNET_EN bit must be asserted. Once in low-power mode, CHI submodule is shut down and CAN PE submodule is kept active so that the Rx process is still active to filter incoming messages (see "Receive Process under Pretended Networking Mode") as defined by the configuration registers (see "CAN CTRL1 PN Register (CAN_CTRL1_PN)"). Upon detecting a Wake Up event, a Wake Up interrupt is issued to the system.

To enter Pretended Networking mode, the FlexCAN must be in Normal mode (not in Freeze or Module Disable mode). Under Pretended Networking mode, the FlexCAN keeps itself synchronized with the CAN bus in Doze or Deepsleep mode. Then, when either Doze or Deepsleep mode is requested, FlexCAN performs the following steps:

- Waits to be in Idle state, or else waits for the third bit of Intermission, and then checks it to be recessive.
- Sets the LPMACK bit in CAN_MCR.
- Requests the shutdown of the CHI submodule clock, while keeping the PE submodule clock active.

The FlexCAN can exit Pretended Networking mode in the following ways:

- The CPU removing the Doze mode request.
- The CPU negating the CAN_MCR.DOZE bit.
- The CPU removing the Deepsleep mode request.

The FlexCAN will wait until Bus Idle or third bit of the Intermission state to negate the CAN_MCR.LPMACK bit.

The above exit methods can be triggered either by the FlexCAN action upon detecting a wake up event and issuing the respective interrupt, or by the CPU itself upon being waken up by another method. As a consequence, the FlexCAN will wait until the Bus Idle state or until the third bit of the Intermission state to negate the CAN_MCR.LPMACK bit and resume Normal mode. This procedure ensures that the FlexCAN will be synchronized to the CAN bus after exiting Pretended Networking mode. The CPU must wait for the CAN_MCR.LPMACK bit to be negated before performing any access to the FlexCAN.

When the CAN_MCR.PNET_EN bit is asserted, the CPU must disable the Self Wake Up feature by negating the CAN_MCR.SLFWAK bit.

20.4.14 Interrupts

The FlexCAN module has many interrupt sources: interrupts due to MBs and interrupts due to the OR'd interrupts from MBs, Bus Off, Bus Off Done, Error, Error Fast (errors detected in the data phase of CAN FD format messages with the BRS bit set), Wake Up, Wake Up Match, Wake Up Timeout, Tx Warning, and Rx Warning.

Each one of the MBs can be an interrupt source if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has an assigned flag bit in the CAN_IFLAG registers. The bit is set when the corresponding buffer completes a successful transfer and is cleared when the CPU writes '1' to it (unless another interrupt is generated at the same time).

NOTE: It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions

Yuntu Microelectronics

may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Legacy Rx FIFO is enabled (CAN_MCR.RFEN = 1) and DMA is disabled (CAN_MCR.DMA = 0), the interrupts corresponding to MBs 0 to 7 have different meanings:

- Bit 7 of the CAN_IFLAG1 register becomes the “Legacy Rx FIFO Overflow” flag.
- Bit 6 becomes the “Legacy Rx FIFO Warning” flag.
- Bit 5 becomes the “Frames Available in Legacy Rx FIFO” flag.
- Bit 4-0 are unused

See ”Interrupt Flags 1 Register (CAN_IFLAG1)” for more information.

If both Legacy Rx FIFO and DMA are enabled (CAN_MCR.RFEN =1 and CAN_MCR.DMA = 1), the FlexCAN does not generate any Legacy Rx FIFO interrupt. Bit 5 of the CAN_IFLAG1 register still indicates “Frames Available in Legacy Rx FIFO” and generates a DMA request. Bits 7, 6, and 4–0 are unused.

NOTE: The Legacy Rx Rx FIFO cannot be enabled when CAN FD feature is enabled.

For a combined interrupt where multiple MB interrupt sources are OR’d together, the interrupt is generated when any of the associated MBs (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the CAN_IFLAG registers to determine which MB or FIFO source caused the interrupt.

The interrupt sources for Bus Off, Bus Off Done, Error, Error Fast, Wake Up, Tx Warning and Rx Warning generate interrupts like the MB interrupt sources, and can be read from CAN_ESR1 register. The Bus Off, Error, Tx Warning, and Rx Warning interrupt mask bits are located in the CAN_CTRL1 register; the Wake Up interrupt mask bit is located in CAN_MCR.

The interrupt sources for Pretended Networking (Wake up by Match Flag and Wake Up by Timeout Flag) can be read in the CAN_WU_MTC register and the respective interrupts masks bits are located in CAN_CTRL1_PN register.

20.4.15 Bus Interface

CPU accesses to FlexCAN registers are subject to the following rules:

- Unrestricted read and write access to supervisor registers (registers identified with S/U in table: ”CAN register access and reset information” in Supervisor Mode or with S only) results in an access error.
- Read and write access to implemented reserved address space results in an access error.
- Write access to positions whose bits are all currently read-only results in an access error. If at least one of the bits is not read-only then no access error is issued. Write permission to positions or some of their bits can change depending on the mode of operation or transitory state. Refer to the register and bit descriptions for details.
- Read and write access to unimplemented address space results in an access error.
- Read and write access to RAM located positions during low-power mode results in an access error.
- If CAN_MCR.MAXMB is programmed with a value smaller than the available number of MBs, then the unused memory space can be used as general purpose RAM space. Reserved words within RAM cannot be used. As an example, suppose FlexCAN’s RAM can support up to 16 MBs, CAN_CTRL2.RFFN is 0x0, and CAN_MCR.MAXMB is programmed with zero. The maximum number of MBs in this case becomes one. The RAM starts at 0x0080, and the space from 0x0080 to 0x008F is used by the one MB. The memory space from 0x0090 to 0x017F is available. The space between 0x0180 and 0x087F is reserved. The space from 0x0880 to 0x0883 is used by the one

Yuntu Microelectronics

Individual Mask and the available memory in the Mask Registers space would be from 0x0884 to 0x08BF. From 0x08C0 through 0x09DF, there are reserved words for internal use which cannot be used as general purpose RAM. As a general rule, free memory space for general purpose depends only on MAXMB.

20.4.16 Detection and Correction of Memory Errors

The FlexCAN supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated with 5 parity bits, which ensures a Hamming distance of 4. The error correction mechanism ensures that in this 13-bit word, errors in one bit can be corrected (correctable errors) and errors in 2 bits can be detected but not corrected (non-correctable errors). Errors in more than 2 bits may not be detected. In case of non-correctable errors, the corrupted data is not changed by the error correction logic.

When a read access is performed, the parity bits are used to calculate a syndrome, which indicates the error in each byte.

The FlexCAN detects a non-correctable error in the event either an all-zeros or an all-ones read occurs. See ”Error Report Syndrome Register (CAN_RERRSYNR)”.

Memory errors are indicated to the host through CAN_ERRSR and bus transfer errors, and reported through CAN_RERRAR, CAN_RERRDR, and CAN_RERRSYNR.

The error detection and correction mechanism is activated by the ECCDIS bit in the Memory Error Control Register (CAN_MECR). When error detection and correction is disabled, updates on indications and reporting registers are stopped, but the parity bits are still calculated and written along with data in memory write operations to ensure that memory has consistent parity bits associated with the data.

NOTE: All FlexCAN memory must be initialized before starting its operation in order to have the parity bits in memory properly updated. Refer to ”RAM Initialization Addresses” for the peripheral bus addresses that must be written to initialize the FlexCAN memory for your FlexCAN hardware configuration. The WRMFRZ bit in the Control 2 Register (CAN_CTRL2) grants write access to all memory positions that require initialization. The CAN_MCR.RFEN bit must not be set during memory initialization.

To avoid accidentally changing the critical error correction configuration, the following protocol must be used to enable the update of the Memory Error Control Register (CAN_MECR):

- By default, the ECRWRE bit in Control 2 Register (CAN_CTRL2) is 0 and the ECRWRDIS bit in Memory Error Control Register (CAN_MECR) is 1.
- Set CAN_CTRL2.ECRWRE.
- Clear CAN_MECR.ECRWRDIS.
- All writes to CAN_MECR must keep ECRWRDIS cleared.
- After configuration is done, lock CAN_MECR by either setting ECRWRDIS or clearing ECRWRE.

20.4.16.1 Sources of Memory Access

The FlexCAN memory can be accessed by two major sources (or requestors):

- Host (CPU): The largest word accessed is 32 bits wide.
- FlexCAN internals processes (Rx Matching, Tx Arbitration, Move-in on reception, Move-out on transmission): The largest word accessed is 64 bits wide.

The way that non-correctable errors are indicated and reported depends on the source of access.

Yuntu Microelectronics

20.4.16.2 Error Indication

Memory errors are indicated by flags HANCEIF, FANCEIF, and CEIF in CAN_ERRSR. Noncorrectable errors detected in memory reads requested by the host are indicated separately from the errors detected in requests by FlexCAN internal processes. The FlexCAN makes no distinction of the source of the access when correctable errors are detected. There are 3 independent flags for these 3 cases, and each flag will raise an interrupt unless it is masked by mask bits in CAN_MECR. If both non-correctable and correctable errors are found in different bytes in the same read operation, both flags are set.

A non-correctable error detected in host access is also indicated as a bus transfer error. A bus wait request may be asserted to extend the memory transaction to the moment the report registers are updated. This indication cannot be masked. If the flag bit CAN_ERRSR.HANCEIF is not masked, the same non-correctable error will raise a bus transfer error and an interrupt request.

Each indication flag has one Overrun flag in CAN_ERRSR. The Overrun flags do not request interrupts. Overrun flags for non-correctable errors indicate that other errors of the same nature were detected after current error being treated, while overrun flags for correctable errors indicate that other errors of the same nature were detected before the current error being treated. The recommended handling sequence for error indication is:

- Get error report information from report registers.
- Use this information to take proper measures in the application.
- Clear the HANCEIF, FANCEIF, CEIF flags.
- If the Overrun flag is active:
  - Alert application that at least one error could not be managed.
  - Clear the Overrun flag.

The FlexCAN internal processes can access memory in transactions larger than 32 bits. For the indication, this kind of access is considered a consecutive sequence of 32-bit accesses.

If errors are found in 2 or more 32-bit words, the Interrupt and Overrun flags are set simultaneously.

20.4.16.3 Error Reporting

The report registers CAN_RERRAR, CAN_RERRDR, and CAN_RERRSYNR provide detailed information about the address read, raw data, and syndrome read with error and indicated by the flags described in ”Error Indication”. The address, data, and syndrome registers are updated simultaneously along with the error flags, according to these rules:

- If either of the 2 non-correctable error flags is currently set, the report registers are not updated (the previous non-correctable error reporting is preserved).
- Otherwise (either no error flag is currently set or only the correctable error flag is currently set), the report registers are updated according to the new error, or according to the most severe of new errors if non-correctable and correctable errors are simultaneously detected.

Reporting of errors detected in accesses larger than 32 bits follows the rules described in Error Indication and in ”Error Report Address Register (CAN_RERRAR)”.

The address reported in CAN_RERRAR and defined in CAN_ERRIAR are not the same as listed in the module memory map. The relation between the reported addresses and the respective ones in the module memory map is shown in ”Error Injection Address Register (CAN_ERRIAR)”.

Addresses reported when reading memory portions organized as FIFOs, such as the Legacy Rx FIFO Structure, Enhanced Rx FIFO Structure, and the Legacy Rx FIFO Information Register (CAN_RXFIR), refer to the address of the specific entry accessed in the FIFO, not to the FIFO base address.

Yuntu Microelectronics

To ensure coherence of the error report registers, it is necessary to turn off the report update by setting the CAN_MECR.RERRDIS bit before reading the report registers.

20.4.16.4 Response to Errors

Correctable errors have no consequence on FlexCAN operation because affected data is corrected before its use by the host or FlexCAN internal processes.

For host-initiated reads, a non-correctable error may affect the host, but does not affect FlexCAN operation.

Non-correctable errors detected on memory reads requested by the FlexCAN internal processes may result in incorrect operation depending on the state of the CAN_MECR.NCEFAFRZ bit, as follows:

- During reception (either matching or move-in processes), when a non-correctable error occurs, an incorrect destination may be selected to store the incoming frame, a corrupted frame may be stored in the correct destination, or both. If NCEFAFRZ is set, the FlexCAN stops operation automatically and enters Freeze mode to prevent corrupted data from being treated as valid by FlexCAN internal processes. If NCEFAFRZ is negated, the FlexCAN continues working and a corrupted frame is received.
- During the arbitration process, when a non-correctable error occurs, either a nonhighest priority Tx message buffer may be mistakenly selected for transmission or its data may be corrupted. If NCEFAFRZ is set, the FlexCAN stops operation automatically and enters Freeze mode before starting the move-out. If NCEFAFRZ is negated, the FlexCAN proceeds to move-out with a corrupted frame that will be transmitted on the CAN bus.
- During the move-out process, when a non-correctable error occurs, a corrupted frame is copied from the selected Tx MB that won the arbitration to the Tx SMB for transmission. If NCEFAFRZ is set, the FlexCAN stops operation automatically and enters Freeze mode before starting the transmission. If NCEFAFRZ is negated, the corrupted frame is transferred from the Tx SMB to the PE submodule and is transmitted on the CAN bus.
- A non-correctable error can also be detected beyond the move-out process, when Tx data is read from Tx SMB (buffer located in RAM) to be transferred to the PE submodule for transmission. In this case, a frame with corrupted ID and/or data is transmitted on the CAN bus. To prevent the frame from being successfully received by the external nodes, the FlexCAN inverts all bits in the CRC field (CRC sequence plus CRC Delimiter), and transmits an Error Flag just after CRC Delimiter as a result of self-detecting a Bit1 Error and a Form Error due to the CRC field inversion. If NCEFAFRZ is set, the FlexCAN stops operation automatically and enters Freeze mode just after the Error Frame. If NCEFAFRZ is negated, the FlexCAN may attempt to re-transmit the same frame, as long as no other higher priority Tx MB is subsequently configured for transmission. In the event the non-correctable error persists, the FlexCAN eventually reaches the Bus Off state because of consecutive error detections. The CAN_ECR.TXERRCNT field is updated every time the FlexCAN inverts the CRC field causing errors as described above.

When NCEFAFRZ is set and the FlexCAN enters Freeze mode, only the CPU can cause then FlexCAN to exit Freeze mode and resume Normal mode. Assertion of CAN_MECR.NCEFAFRZ is the only way to prevent corrupted frames from being transmitted on the CAN bus up to the move-out internal process.

The error report registers can provide information to the application for customized management of these situations.

Yuntu Microelectronics

20.4.16.5 Error Injection

The error injection registers CAN_ERRIAR, CAN_ERRIDPR, and CAN_ERRIPPR are used to inject errors in memory reads in order to force errors and consequently update the indication and reporting registers. The relation between the error injection addresses and the respective ones in the module memory map is shown in ”Error Injection Address Register (CAN_ERRIAR)”.

The injection is done by flipping the data and parity bits corresponding to the bits set to 1 in CAN_ERRIDPR and CAN_ERRIPPR. Injection can be selected specifically for memory accesses requested by host or by FlexCAN internal processes.

In case of accesses larger than 32 bits, the CAN_MECR.EXTERRIE bit extends the injection pattern, replicating it in 32-bit words to fill the width of the access.

NOTE: It is very unlikely, but error injection may correct a bit with error. This will not raise the error flags and reports as expected.

To ensure coherence among error injection registers and avoid spurious error injections, the HAERRIE and FAERRIE bits in CAN_MECR must be cleared when configuring the memory injection registers.

20.5 Initialization / Application Information

The FlexCAN module may be reset in three ways:

- Chip level hard reset, which resets all memory mapped registers asynchronously.
- SOFTRST bit in CAN_MCR, which resets some of the memory mapped registers synchronously. See ”CAN register access and reset information” to see what registers are affected by soft reset.
- Chip level soft reset, which has the same effect as the SOFTRST bit in CAN_MCR, see chapter IPC.

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. The CAN_MCR.SOFTRST bit remains asserted while soft reset is pending, so software can poll this bit to know when the reset has completed. Also, soft reset cannot be applied while clocks are shut down in a low-power mode. The low-power mode should be exited and the clocks resumed before applying soft reset.

The clock source should be selected when the FlexCAN is in Module Disable mode (see CAN_CTRL1.CLKSRC bit). After the clock source is selected and the module is enabled (CAN_MCR.MDIS bit negated), the FlexCAN automatically enters Freeze mode. In Freeze mode, the FlexCAN is un-synchronized to the CAN bus, the HALT and FRZ bits in CAN_MCR are set, the internal state machines are disabled, and the FRZACK and NOTRDY bits in CAN_MCR are set. The Tx pin is in recessive state and the FlexCAN does not initiate any transmission or reception of CAN frames. Note that the MBs and the Rx Individual Mask Registers are not affected by reset, so they are not automatically initialized.

20.5.1 RAM Initialization

The FlexCAN RAM should be initialized prior to use to avoid ‘X’ propagation in simulation and to ensure that all registers implemented in RAM contain known data.

When ECC is enabled for the FlexCAN RAM (configuration parameter ECC_EN = 1), all FlexCAN memory must be initialized before starting its operation so that the parity bits in memory are properly updated.

NOTE: The CAN_MCR.RFEN bit must not be set during memory initialization.

Yuntu Microelectronics

To initialize the FlexCAN RAM:

1. If ECC is enabled, set CAN_CTRL2.WRMFRZ to enable write access to all memory positions that require initialization.
2. Write the addresses needed to initialize the complete FlexCAN RAM according to your selected FlexCAN hardware configuration, as described in "RAM Initialization Addresses".

20.5.2 FlexCAN Initialization Sequence

For any configuration change/initialization, it is required that the FlexCAN is put into Freeze mode (see "Freeze Mode"). The following steps are a generic initialization sequence applicable to the FlexCAN module:

1. Initialize the FlexCAN RAM as described in "RAM Initialization"RAM Initialization.
2. Initialize the Module Configuration Register (CAN_MCR):
   - Enable the individual filtering per MB and reception queue features by setting IRMQ.
   - Enable the warning interrupts by setting WRNEN.
   - If required, disable frame self reception by setting SRXDIS.
   - Enable the Legacy Rx FIFO by setting RFEN or the Enhanced Rx FIFO by setting ERFEN.
   - If the Legacy Rx FIFO or Enhanced Rx FIFO is enabled and DMA is required, set the DMA bit.
   - If Pretended Networking mode is required, set PNET_EN.
   - Enable the abort mechanism by setting AEN.
   - Enable the local priority feature by setting LPRIOEN.
3. Initialize the Control 1 Register (CAN_CTRL1) and optionally the CAN Bit Timing Register (CAN_CBT). Initialize also the CAN FD CAN Bit Timing Register (CAN_FDCBT):
   - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, and RJW.
   - Optionally determine the extended bit timing parameters: EPROPSEG, EPSEG1, EPSEG2, and ERJW.
   - Determine the CAN FD bit timing parameters: FPROPSEG, FPSEG1, FPSEG2, and FRJW.
   - Determine the bit rate by programming the PRESDIV field and optionally the EPRESDIV field.
   - Determine the CAN FD bit rate by programming the FPRESDIV field.
   - Determine the internal arbitration mode (LBUF bit).
4. Initialize the Message Buffers:
   - The C/S word of all MBs must be initialized.
   - If either the Legacy Rx FIFO or Enhanced Rx FIFO was enabled, the corresponding ID filter table must be initialized.
   - Other entries in each MB should be initialized as required.
5. Initialize the Rx Individual Mask Registers (CAN_RXIMRn).
6. Set required interrupt mask bits in the CAN_IMASKn registers (for all MB interrupts), in CAN_MCR for Wake Up interrupt, and in CAN_CTRL1/CAN_CTRL2 for Bus Off and Error interrupts.
7. Negate the HALT bit in CAN_MCR.

After the last step listed above, the FlexCAN attempts to synchronize to the CAN bus.

20.6 FlexCAN Ram Usage and Initialization

The FlexCAN Controller requires an externally connected RAM for storage of message buffers and register data for memory-mapped register that are implemented in external RAM. As shown in below

Yuntu Microelectronics

Figure, the RAM is accessed:

- By the host CPU (through the IPS bus) when writing/reading message buffers and registers that are implemented in RAM.
- By FlexCAN internal processes when accessing RAM data that is shared with the CPU and for storage of data such as SMBs, which are used only by FlexCAN internal processes and not directly accessible by the CPU when the FlexCAN is operating in Normal mode.

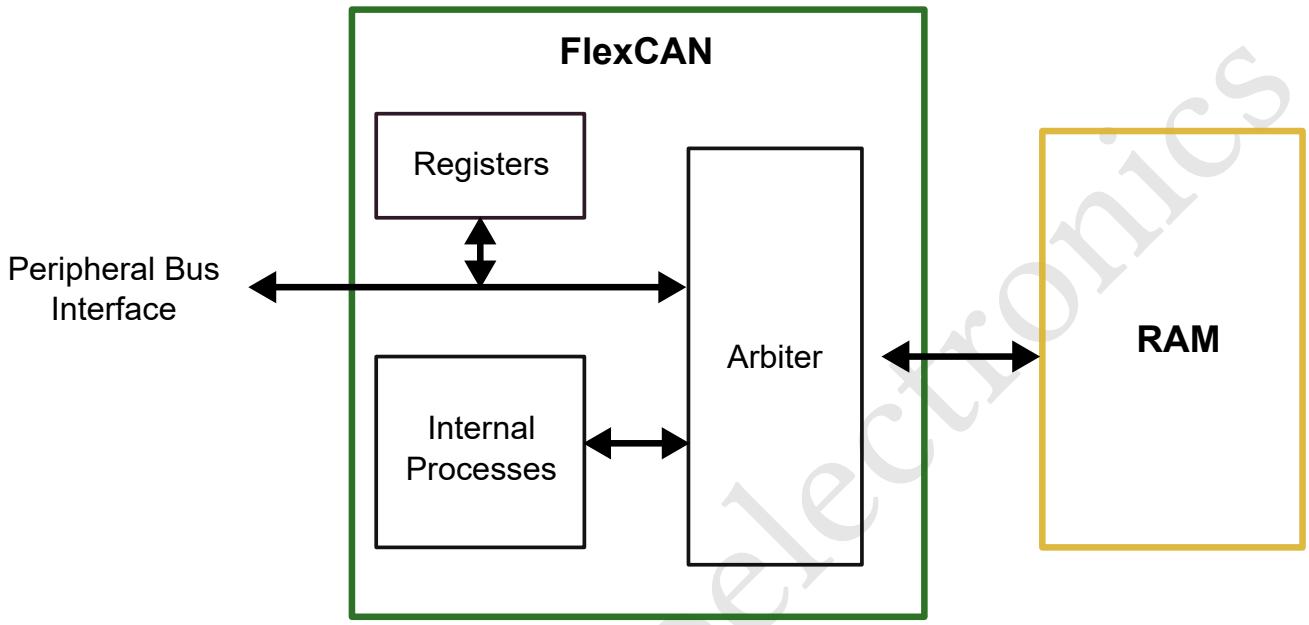


Figure 20.26: Sharing of RAM Between CPU and CAN Internal Processes

20.6.1 RAM Size

The size required for the RAM depends on FlexCAN configuration. The data bus width is:

- 64 bits when ECC_EN = 0 (ECC not supported)
- 104 bits when ECC_EN = 1 (ECC is supported)

The number of addresses required and, therefore, the required RAM address bus width depends on the following configuration parameters, as detailed in the FlexCAN Controller Integration Guide:

- NUMBER_OF_MB – Number of message buffers supported
- IRMQ_EN – Support for Rx Individual Mask Registers
- FD_EN – Support for CAN FD
- ENHANCE_MB_MEM – Support for additional message buffer memory space
- TIME_STAMP_EN – Support for high-resolution time stamp
- ERX_FIFO_EN – Support for Enhanced Rx FIFO
- ERX_FIFO_SIZE – Enhanced Rx FIFO size (number of messages that can be stored in the Enhanced Rx FIFO)
- ERX_FIFO_FLT_ELEM – Number of Enhanced Rx FIFO filter elements

20.6.2 RAM Usage

As shown below 5 Figures, the RAM stores the following groups/regions of data:

Yuntu Microelectronics

- Message buffers – 16 bytes for each configured message buffer.
- Rx Individual Mask Registers (CAN_RXIMRn) – 4 bytes per message buffer when configuration parameter IRMQ_EN = 1.
- Rx Mask Registers CAN_RXMGMASK, CAN_RX14MASK, CAN_RX15MASK, CAN_RXFGMASK – 4 bytes per register.
- Rx FIFO Information Register – 24 bytes accessed as FIFO through CAN_RXFIR during normal operation.
- Classic (non-CAN-FD) mode SMBs – One Tx SMB and two Rx SMBs used when CAN_MCR.FDEN = 0. Each SMB is 16 bytes.
- CAN-FD mode SMBs – One Tx SMB and two Rx SMBs used when CAN_MCR.FDEN = 1. Each SMB is 72 bytes.
- Enhanced Rx FIFO – When configuration parameter ERX_FIFO_EN = 1, there is an 80-byte (FD_EN = 1) or 24-byte (FD_EN = 0) region for each message that can be stored in the Enhanced Rx FIFO, as determined by configuration parameter ERX_FIFO_SIZE.
- Enhanced Rx FIFO filter elements – When configuration parameter ERX_FIFO_EN = 1, there is a 4-byte region for each Enhanced Rx FIFO filter element, as determined by configuration parameter ERX_FIFO_FLT_ELEM.

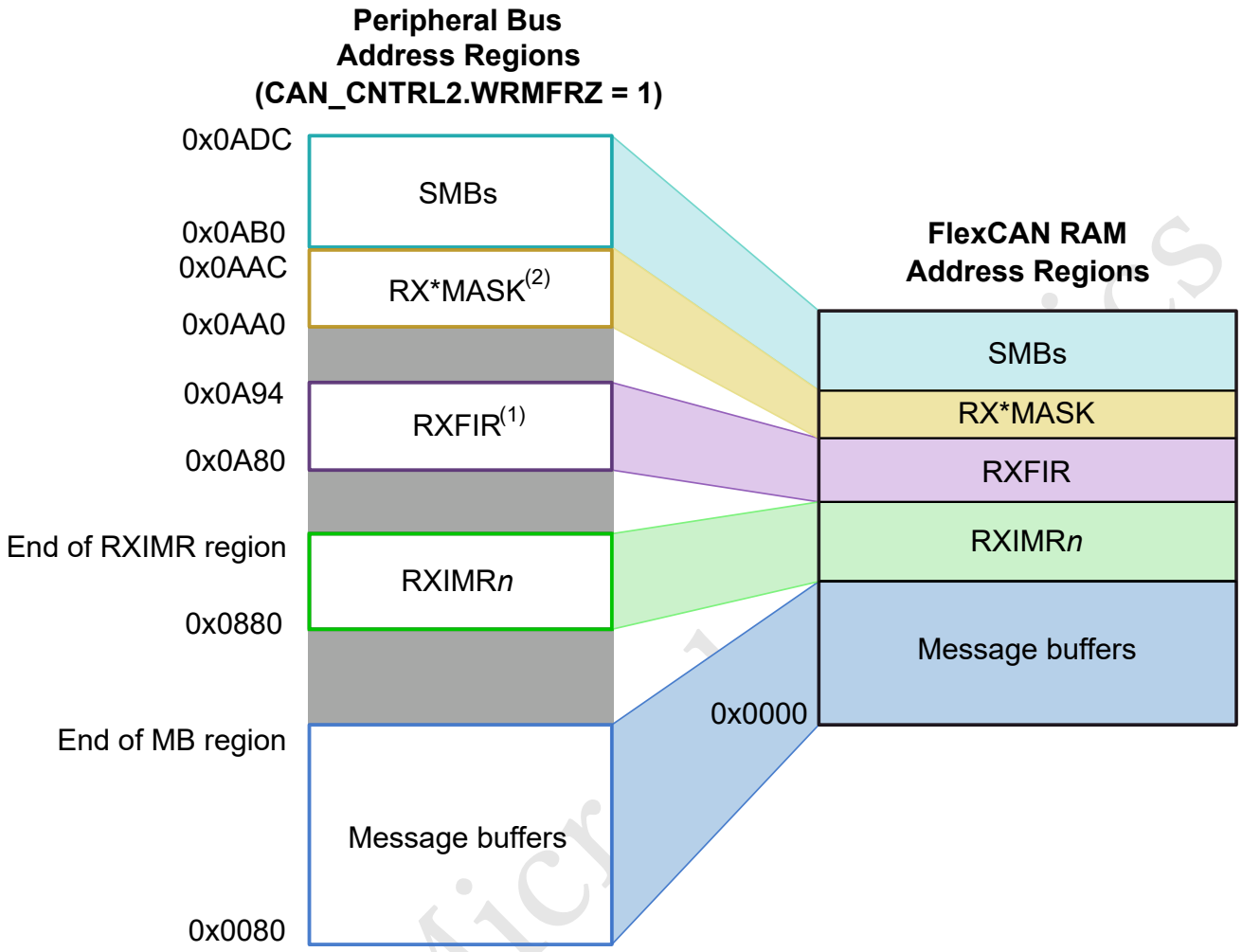
From the peripheral (IPS) bus, the base address of each region is the same regardless of the selected FlexCAN configuration, although the sizes of the message buffer and CAN_RXIMRn regions depend on configuration parameter NUMBER_OF_MB.

On the RAM interface, the message buffer region begins at address 0x0000. The addresses of regions above the message buffer region scale according to NUMBER_OF_MB and IRMQ_EN, in order to fully utilize the smallest possible RAM required for the selected configuration.

Not all RAM locations are accessible from the peripheral bus during normal operation. Therefore, when ECC_EN = 1 (ECC is enabled), the CAN_CNTRL2.WRMFRZ bit can be set to enable write access to all RAM locations so that the ECC bits can be initialized.

Yuntu Microelectronics

Below figure shows the peripheral bus to RAM address mapping when CAN-FD is not supported (configuration parameter FD_EN = 0).



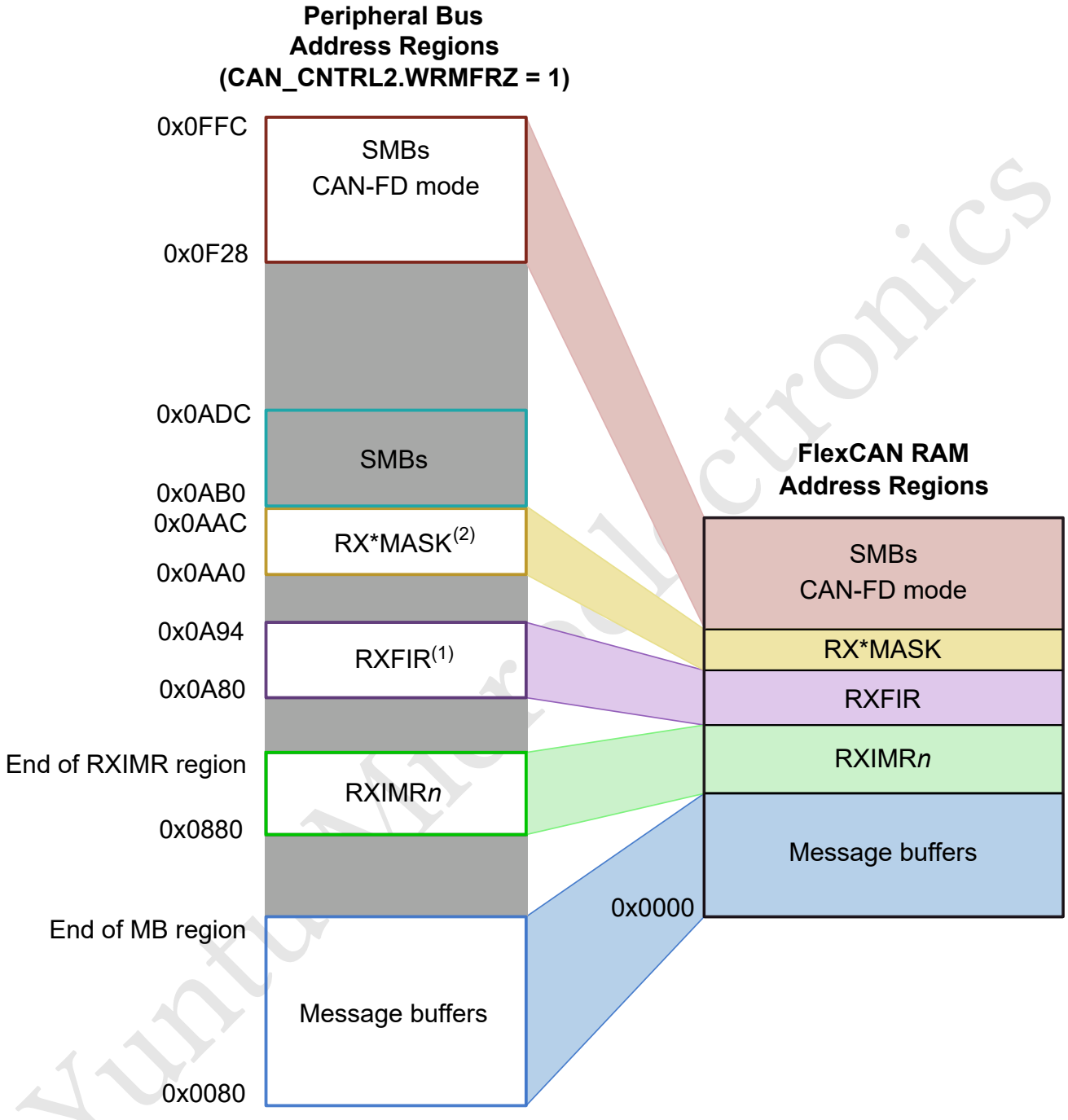
1.Accessed as FIFO at 0x004C in Normal mode.

2.Accessed at addresses 0x0010,0x0014,0x0018,0x0048 in Normal mode.

Figure 20.27: Mapping of Peripheral Bus Address to CAN RAM when CAN FD Is Not Supported $(FD_EN = 0)$

Yuntu Microelectronics

Below figure shows the peripheral bus to RAM address mapping when CAN-FD is supported (configuration parameter FD_EN = 1) and enabled (CAN_MCR.FDEN = 1), and ENHANCE_MB_MEM = 0.



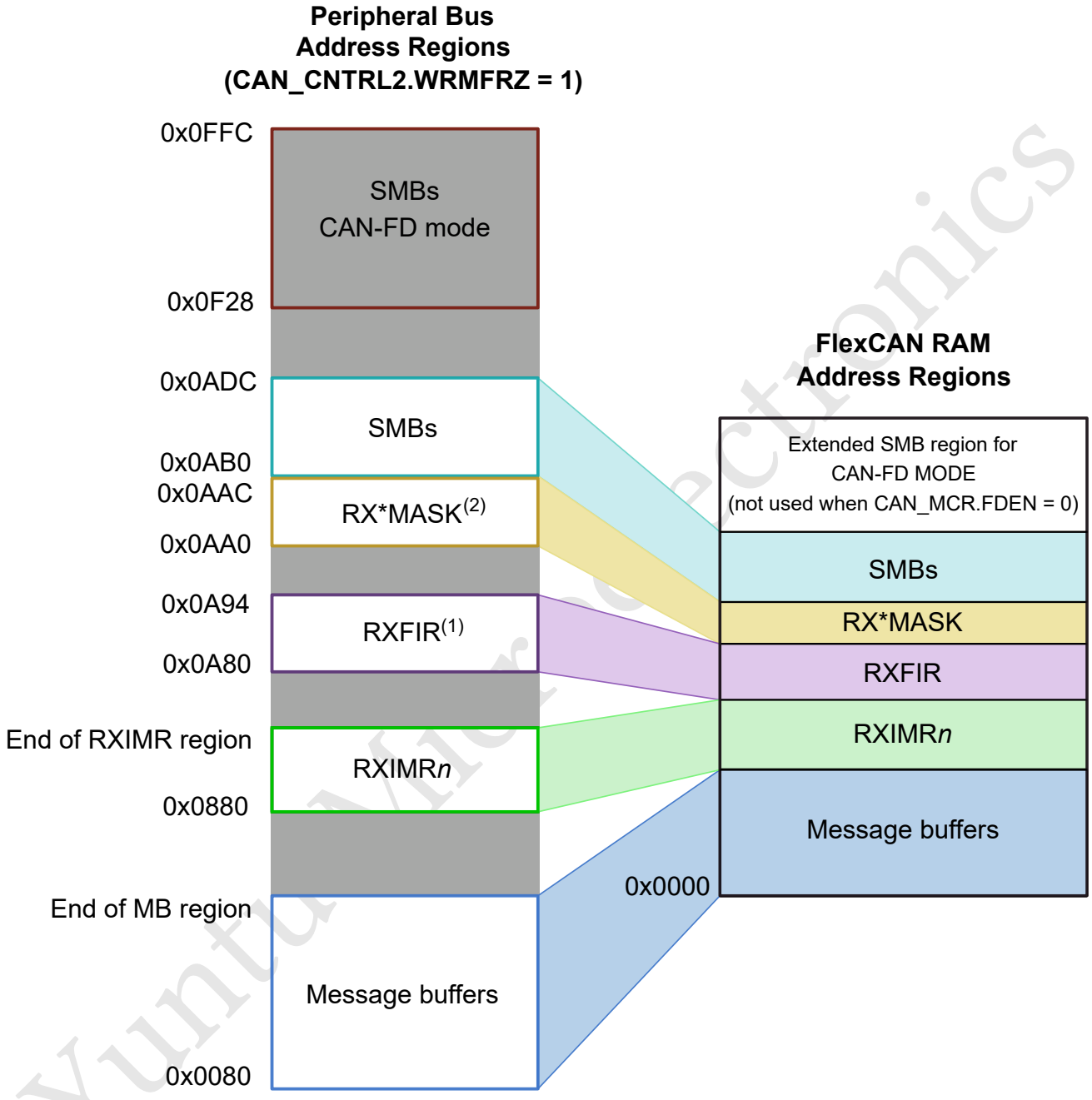
1.Accessed as FIFO at 0x004C in Normal mode.

2.Accessed at addresses 0x0010,0x0014,0x0018,0x0048 in Normal mode.

Figure 20.28: Mapping of Peripheral Bus Address to CAN RAM when CAN FD Is Supported and Enabled $(FD_EN = 1, CAN_MCR.FDEN = 1)$

Yuntu Microelectronics

Below figure shows the peripheral bus to RAM address mapping when CAN-FD is supported (configuration parameter FD_EN = 1) and disabled (CAN_MCR.FDEN = 0). The implemented RAM size supports the larger SMBs used in CAN-FD mode. However, since CAN-FD mode is not enabled and only the smaller non-FD mode SMBs are used, there is unused RAM space.



1.Accessed as FIFO at 0x004C in Normal mode.

2.Accessed at addresses 0x0010,0x0014,0x0018,0x0048 in Normal mode.

Figure 20.29: Mapping of Peripheral Bus Address to CAN RAM when CAN FD Is Supported But Not Enabled $(FD_EN = 1, CAN_MCR.FDEN = 0)$

Yuntu Microelectronics

Below figure shows the peripheral bus to RAM address mapping when the enhanced MB memory feature is implemented (ENHANCE_MB_MEM = 1). Enhanced MB memory can only be implemented when CAN-FD is supported (FD_EN = 1) and NUMBER_OF_MB = 128. Therefore, the RAM address region for enhanced MB memory is always at RAM addresses 0x0800–0x0FFC and maps to peripheral bus addresses 0x1000–0x17FC.

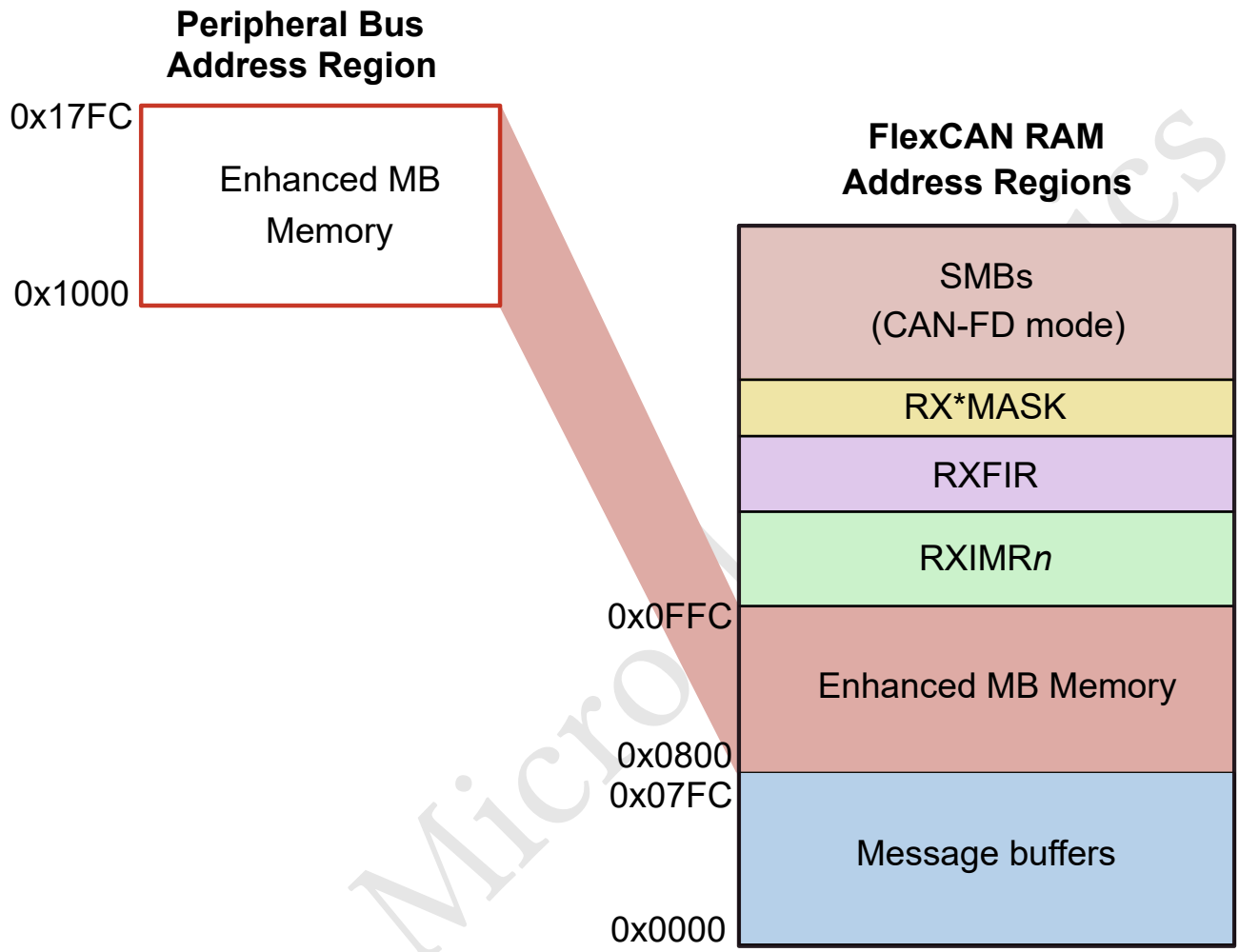


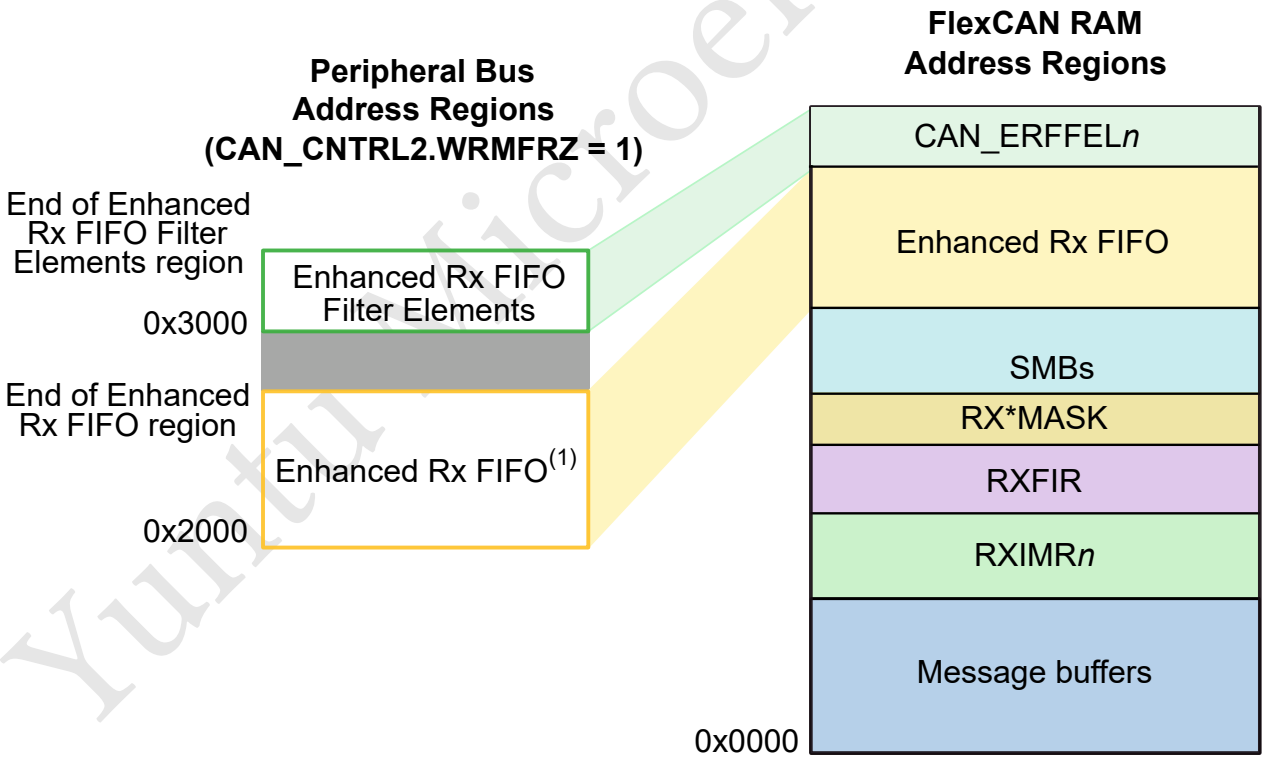
Figure 20.30: Mapping of Peripheral Bus Address for Enhanced MB Memory Region

Yuntu Microelectronics

Below figure shows peripheral bus to RAM address mapping for the Enhanced Rx FIFO and Enhanced Rx FIFO filter elements regions, which are present when ERX_FIFO_EN = 1:

- The size of the Enhanced Rx FIFO region depends on the value of configuration parameter ERX_FIFO_SIZE, which is the number of messages that can be stored in the Enhanced Rx FIFO. There are 80 bytes allocated for each message:
  - In Freeze mode with ECC enabled and CAN_CTRL2.WRMFRZ = 1, the entire Enhanced Rx FIFO region is accessible from the peripheral bus for FlexCAN RAM initialization. The start address is 0x2000. The end address (word-aligned) is 0x2000 + (80 x ERX_FIFO_SIZE) - 4. For example, if ERX_FIFO_SIZE = 2, the range of the Enhanced Rx FIFO region is 0x2000–0x209C. If ERX_FIFO_SIZE = 32, the range of the Enhanced Rx FIFO region is 0x2000–0x29FC.
  - In Normal mode, only the oldest unread entry in the Enhanced Rx FIFO is accessible from the peripheral bus (read-only), at addresses 0x2000–0x204C.
- The size of the Enhanced Rx FIFO filter elements region depends on the value of configuration parameter ERX_FIFO_FLT_ELEM, which is the Enhanced Rx FIFO filter elements implemented. There are 4 bytes allocated for each filter element.

  The Enhanced Rx FIFO filter elements region start address is 0x3000. The end address (word-aligned) is 0x3000 + (4 x ERX_FIFO_FLT_ELEM) - 4. For example, if ERX_FIFO_FLT_ELEM = 2, the range of the Enhanced Rx FIFO filter elements region is 0x3000–0x3004. If ERX_FIFO_FLT_ELEM = 128, the range of the Enhanced Rx FIFO region is 0x3000–0x31FC.



1. Oldest unread Enhanced Rx FIFO entry is accessible at 0x2000-0x204C in Normal mode.

Figure 20.31: Mapping of Peripheral Bus Address to CAN RAM for Enhanced RX FIFO Regions

Yuntu Microelectronics

Below figure shows peripheral bus to RAM address mapping for the high-resolution time stamp region, which is present when TIME_STAMP_EN = 1. In the RAM, the high-resolution time stamp region starts at the next address above the SMBs region:

- High-resolution time stamps for the Rx SMBs occupy the first two words:
  - Rx_SMB0_TIME_STAMP is mapped to peripheral bus address 0x0C20.
  - Rx_SMB1_TIME_STAMP is mapped to peripheral bus address 0x0C24.
- The CAN_HR_TIME_STAMPn registers are also implemented in RAM starting at peripheral bus address 0x03C0.

  There is one CAN_HR_TIME_STAMPn register for each implemented message buffer. Therefore, the end address of the CAN_HR_TIME_STAMPn registers region is 0x03C0 + (4 x NUMBER_OF_MB) - 4. For example, if NUMBER_OF_MB = 128, the range of the CAN_HR_TIME_STAMPn registers region is 0x03C0–0x0E2C.

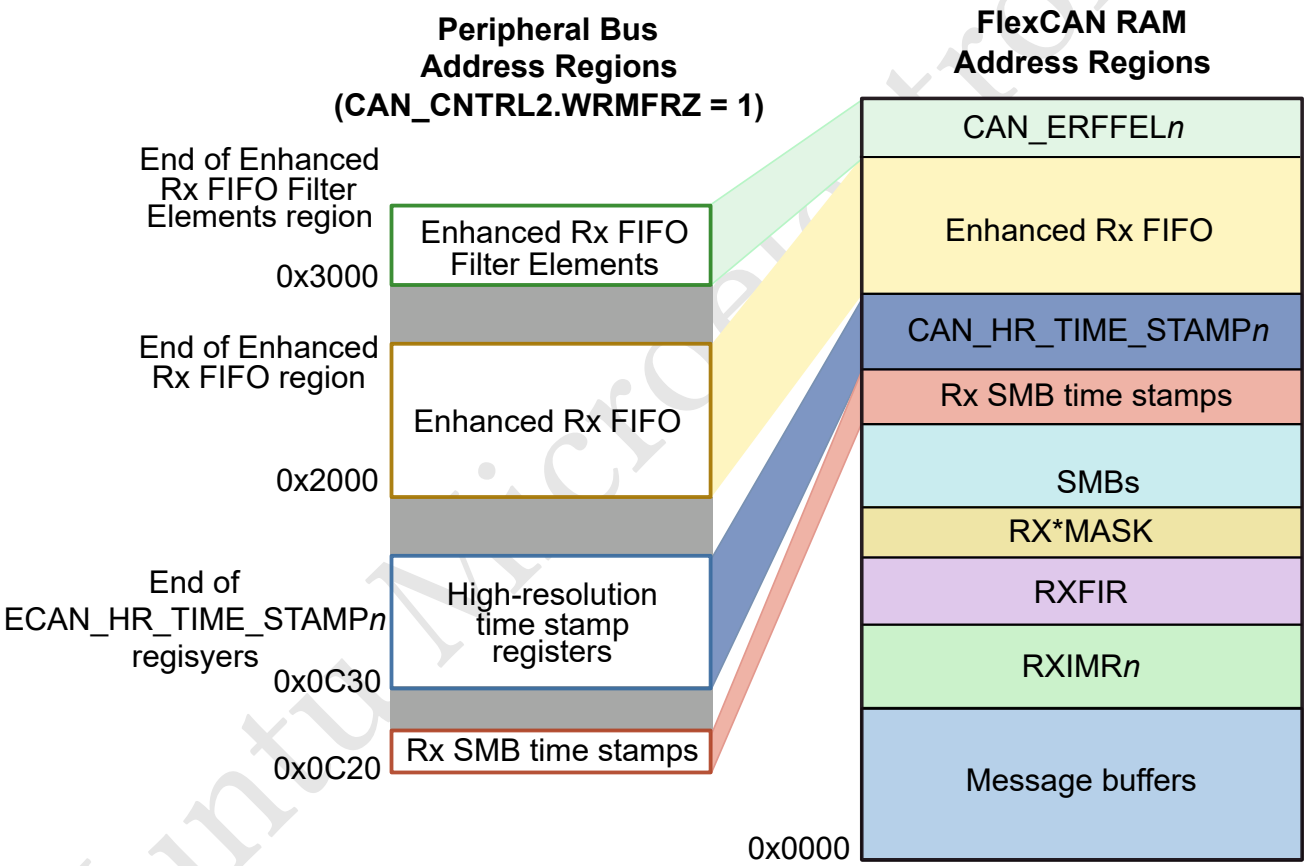


Figure 20.32: Mapping of Peripheral Bus Address to CAN RAM for High-resolution Time Stamp Regions

20.6.3 RAM Initialization Addresses

When ECC is enabled (parameter ECC_EN = 1), the entire FlexCAN RAM must be initialized at startup so that each location and its corresponding ECC bits are written before being read.

When ECC is not enabled (parameter ECC_EN = 0), all accessible RAM locations should also be initialized so that registers implemented in RAM contain known data and to avoid 'X' propagation in simulation.

NOTE: The CAN_MCR.RFEN bit must not be set during RAM initialization.

To initialize the FlexCAN RAM when ECC is enabled:

1. Set CAN_CTRL2.WRMFRZ to enable Freeze mode write access to all memory positions that require initialization.
2. While in Freeze mode, write to the required locations depending on your FlexCAN configuration as shown in below Table.
3. If enhanced MB memory is supported (ENHANCE_MB_MEM = 1), also write to the following locations:
   - 0x1000 through 0x17FF
4. If the Enhanced Rx FIFO is supported (ERX_FIFO_EN = 1), also write to the following locations:
   - 0x2000 through [0x2000 + (80 x ERX_FIFO_SIZE) - 4]
   - 0x3000 through [0x3000 + (4 x ERX_FIFO_FLT_ELEM) - 4]
5. If the high-resolution time stamp feature is supported (TIME_STAMP_EN = 1), also write to the following locations:
   - 0x0C20 and 0x0C24
   - 0x0C30 through [0x0C30 + (4 x NUMBER_OF_MB) - 4]

Table 20.102: CAN Initialization Address for Basic Features (ECC Enabled)

Number of MBs	Init Address (IRMQ_EN=1)	Init Address (IRMQ_EN=1)
128	0x0080-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC	0x0080-0x087C 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC
96	0x0080-0x067C 0x0880-0x09FC 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC	0x0080-0x067C 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC
64	0x0080-0x047C 0x0880-0x097C 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC	0x0080-0x047C 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC
32	0x080-0x27C 0x880-0x8FC 0xA80-0xAAC 0xAB0-0xADC 0xF28-0xFFC	0x080-0x27C 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC
16	0x0080-0x017C 0x0880-0x08BC 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC	0x0080-0x017C 0x0A80-0x0AAC 0x0AB0-0x0ADC 0x0F28-0x0FFC

1. Initialization of addresses 0x0AB0–0x0ADC is required when CAN-FD mode is disabled (CAN_MCR.FDEN = 0) or is not supported.
2. Initialization of addresses 0x0F28–0x0FFC is required when CAN-FD mode is enabled (CAN_MCR.FDEN = 1).

Yuntu Microelectronics

To initialize the FlexCAN RAM when ECC is disabled:

1. While in Freeze mode, write to the required locations depending on your FlexCAN configuration as shown in below Table.
2. If enhanced MB memory is supported (ENHANCE_MB_MEM = 1), also write to the following locations:
   - 0x1000 through 0x17FF
3. If the Enhanced Rx FIFO is supported (ERX_FIFO_EN = 1), also write to the following locations:
   - 0x3000 through [0x3000 + (4 x ERX_FIFO_FLT_ELEM) - 4]
4. If the high-resolution time stamp feature is supported (TIME_STAMP_EN = 1), also write to the following locations:
   - 0x0C30 through [0x0C30 + (4 x NUMBER_OF_MB) - 4]

Table 20.103: CAN Initialization Address for Basic Features (ECC Disabled)

Number of MBs	Init Address (IRMQ_EN=1)	Init Address (IRMQ_EN=1)
128	0x0010-0x0018 0x0048 0x0080-0x0A7C	0x0010-0x0018 0x0048 0x0080-0x087C
96	0x0010-0x0018 0x0048 0x0080-0x067C 0x0080-0x09FC	0x0010-0x0018 0x0048 0x0080-0x067C
64	0x0010-0x0018 0x0048 0x0080-0x047C 0x0880-0x097C	0x0010-0x0018 0x0048 0x0080-0x047C
32	0x0010-0x0018 0x0048 0x0080-0x027C 0x0880-0x08FC	0x0010-0x0018 0x0048 0x0080-0x027C
16	0x0010-0x0018 0x0048 0x0080-0x017C 0x0080-0x08BC	0x0010-0x0018 0x0048 0x0080-0x017C

20.6.4 RAM Addresses for Error Injection and Reporting

When ECC is enabled, the memory addresses used for ECC error injection and error reporting correspond to the RAM addresses generated at the FlexCAN can_addr[13:0] output to the FlexCAN RAM. As described in “RAM Usage”, the mapping of peripheral bus address to FlexCAN RAM address depends on the values of FlexCAN hardware configuration parameters.

20.6.4.1 RAM Addresses for FlexCAN Standard Features

The following tables show the RAM address range for each RAM region and its corresponding peripheral (IPS) bus address range for different values of NUMBER_OF_MB and IRMQ_EN. The RAM locations are accessible from the IPS bus addresses shown Table ”CAN RAM address mappings when number of MB 128” through Table “CAN RAM address mappings when number of MB 16” in Freeze mode when CAN_CNTRL2.WRMFRZ = 1.

Yuntu Microelectronics

The RAM ranges and corresponding IPS addresses of the Tx and Rx SMBs differ depending on whether CAN-FD mode is enabled. Therefore, each table shows the SMB RAM address mapping for both cases:

- CAN-FD disabled (CAN_MCR.FDEN = 0)
- CAN-FD enabled (CAN_MCR.FDEN = 1)

Table 20.104: CAN RAM Address Mappings when Number of MB 128

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
MBs	0x0080-0x087C	0x0000-0x07FC	0x0000-0x07FC
RXIMRs	0x0880-0x0A7C	0x0800-0x09FC	N/A
RXFIR	0x0A80-0x0A94	0x0A00-0x0A14	0x0800-0x0814
RXMGMASK	0x0AA0	0x0A20	0x0820
RXFGMASK	0x0AA4	0x0A24	0x0824
RX14MASK	0x0AA8	0x0A28	0x0828
RX15MASK	0x0AAC	0x0A2C	0x082C
Tx_SMB (FDEN = 0)	0x0AB0-0x0ABC	0x0A30-0x0A3C	0x0830-0x083C
Rx_SMB0 (FDEN = 0)	0x0AC0-0x0ACC	0x0A40-0x0A4C	0x0840-0x084C
Rx_SMB1 (FDEN = 0)	0x0AD0-0x0ADC	0x0A50-0x0A5C	0x0850-0x085C
Tx_SMB (FDEN = 1)	0x0F28-0x0F6C	0x0A30-0x0A74	0x0830-0x0874
Rx_SMB0 (FDEN = 1)	0x0F70-0x0FB4	0x0A78-0x0ABC	0x0878-0x08BC
Rx_SMB1 (FDEN = 1)	0x0FB8-0x0FFC	0x0AC0-0x0B04	0x08C0-0x0904

Table 20.105: CAN RAM Address Mappings when Number of MB 96

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
MBs	0x0080-0x067C	0x0000-0x05FC	0x0000-0x05FC
RXIMRs	0x0880-0x09FC	0x0600-0x077C	N/A
RXFIR	0x0A80-0x0A94	0x0780-0x0794	0x0600-0x0614
RXMGMASK	0x0AA0	0x07A0	0x0620
RXFGMASK	0x0AA4	0x07A4	0x0624
RX14MASK	0x0AA8	0x07A8	0x0628
RX15MASK	0x0AAC	0x07AC	0x062C
Tx_SMB (FDEN = 0)	0x0AB0-0x0ABC	0x07B0-0x07BC	0x0630-0x063C
Rx_SMB0 (FDEN = 0)	0x0AC0-0x0ACC	0x07C0-0x07CC	0x0640-0x064C
Rx_SMB1 (FDEN = 0)	0x0AD0-0x0ADC	0x07D0-0x07DC	0x0650-0x065C
Tx_SMB (FDEN = 1)	0x0F28-0x0F6C	0x07B0-0x07B4	0x0630-0x0674
Rx_SMB0 (FDEN = 1)	0x0F70-0x0FB4	0x07B8-0x083C	0x0678-0x06BC
Rx_SMB1 (FDEN = 1)	0x0FB8-0x0FFC	0x0840-0x0884	0x06C0-0x0704

Table 20.106: CAN RAM Address Mappings When Number of MB 64

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
MBs	0x0080-0x047C	0x0000-0x03FC	0x0000-0x03FC
RXIMRs	0x0880-0x097C	0x0400-0x04FC	N/A
RXFIR	0x0A80-0x0A94	0x0500-0x0514	0x0400-0x0414
RXMGMASK	0x0AA0	0x0520	0x0420
RXFGMASK	0x0AA4	0x0524	0x0424
RX14MASK	0x0AA8	0x0528	0x0428
RX15MASK	0x0AAC	0x052C	0x042C
Tx_SMB (FDEN = 0)	0x0AB0-0x0ABC	0x0530-0x053C	0x0430-0x043C
Rx_SMB0 (FDEN = 0)	0x0AC0-0x0ACC	0x0540-0x054C	0x0440-0x044C
Rx_SMB1 (FDEN = 0)	0x0AD0-0x0ADC	0x0550-0x055C	0x0450-0x045C

Yuntu Microelectronics

Table 20.106 continued from previous page

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
Tx_SMB (FDEN = 1)	0x0F28-0x0F6C	0x0530-0x0574	0x0430-0x0474
Rx_SMB0 (FDEN = 1)	0x0F70-0x0FB4	0x0578-0x05BC	0x0478-0x04BC
Rx_SMB1 (FDEN = 1)	0x0FB8-0x0FFC	0x05C0-0x0604	0x04C0-0x0504

Table 20.107: CAN RAM Address Mappings When Number of MB 32

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
MBs	0x0080-0x027C	0x0000-0x01FC	0x0000-0x01FC
RXIMRs	0x0880-0x08FC	0x0200-0x027C	N/A
RXFIR	0x0A80-0x0A94	0x0280-0x0294	0x0200-0x0214
RXMGMASK	0x0AA0	0x02A0	0x0220
RXFGMASK	0x0AA4	0x02A4	0x0224
RX14MASK	0x0AA8	0x02A8	0x0228
RX15MASK	0x0AAC	0x02AC	0x022C
Tx_SMB (FDEN = 0)	0x0AB0-0x0ABC	0x02B0-0x02BC	0x0230-0x023C
Rx_SMB0 (FDEN = 0)	0x0AC0-0x0ACC	0x02C0-0x02CC	0x0240-0x024C
Rx_SMB1 (FDEN = 0)	0x0AD0-0x0ADC	0x02D0-0x02DC	0x0250-0x025C
Tx_SMB (FDEN = 1)	0x0F28-0x0F6C	0x02B0-0x02F4	0x0230-0x0274
Rx_SMB0 (FDEN = 1)	0x0F70-0x0FB4	0x02F8-0x033C	0x0278-0x02BC
Rx_SMB1 (FDEN = 1)	0x0FB8-0x0FFC	0x0340-0x0384	0x02C0-0x0304

Table 20.108: CAN RAM Address Mappings When Number of MB 16

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
MBs	0x0080-0x017C	0x0000-0x00FC	0x0000-0x00FC
RXIMRs	0x0880-0x08BC	0x0100-0x013C	N/A
RXFIR	0x0A80-0x0A94	0x0140-0x0154	0x0100-0x0114
RXMGMASK	0x0AA0	0x0160	0x0120
RXFGMASK	0x0AA4	0x0164	0x0124
RX14MASK	0x0AA8	0x0168	0x0128
RX15MASK	0x0AAC	0x016C	0x012C
Tx_SMB (FDEN = 0)	0x0AB0-0x0ABC	0x0170-0x017C	0x0130-0x013C
Rx_SMB0 (FDEN = 0)	0x0AC0-0x0ACC	0x0180-0x018C	0x0140-0x014C
Rx_SMB1 (FDEN = 0)	0x0AD0-0x0ADC	0x0190-0x019C	0x0150-0x015C

20.6.4.2 RAM Addresses for FlexCAN Advanced Features

The following advanced features of the FlexCAN Controller also use the FlexCAN RAM and affect the RAM address allocation:

- Enhanced MB memory (implemented when ENHANCE_MB_MEM = 1)
- High-resolution time stamp (implemented when TIME_STAMP_EN = 1)
- Enhanced Rx FIFO (implemented when ERX_FIFO_EN = 1)

20.6.4.2.1 Enhanced MB Memory Region

Enhanced MB memory is implemented when the following configuration parameter value are set:

- NUMBER_OF_MB = 128
- FD_EN = 1

Yuntu Microelectronics

- ENHANCE_MB_MEM = 1

The RAM address region for enhanced MB memory is at RAM addresses 0x0800–0x0FFC and maps to peripheral bus addresses 0x1000–0x17FC.

Below Table shows the RAM address mapping when ENHANCE_MB_MEM = 1.

Table 20.109: CAN RAM Address Mapping when ENHANCE MB MEM 1

RAM Region	IPS Address	RAM Address(IRMQ_EN=1)	RAM Address(IRMQ_EN=0)
MBs	0x0080-0x087C	0x0000-0x07FC	0x0000-0x07FC
Enhanced MB memory	0x1000-0x17FC	0x0800-0x0FFC	0x0800-0x0FFC
RXIMRs	0x0880-0x0A7C	0x1000-0x11FC	N/A
RXFIR	0x0A80-0x0A94	0x1200-0x1214	0x1000-0x1014
RXMGMASK	0x0AA0	0x1220	0x1020
RXFGMASK	0x0AA4	0x1224	0x1024
RX14MASK	0x0AA8	0x1228	0x1028
RX15MASK	0x0AAC	0x122C	0x102C
Tx_SMB (FDEN = 1)	0x0F28-0x0F6C	0x1230-0x1274	0x1030-0x1074
Rx_SMB0 (FDEN = 1)	0x0F70-0x0FB4	0x1278-0x12BC	0x1078-0x10BC
Rx_SMB1 (FDEN = 1)	0x0FB8-0x0FFC	0x12C0-0x1304	0x10C0-0x1104

20.6.4.2.2 High-Resolution Time Stamp Region

The high-resolution time stamp region of RAM begins at the next word address above the last address used for the SMBs, as shown in Below Table.

Within the high-resolution time stamp region, there are three subregions:

- Rx_SMB0_TIME_STAMP – One word at RAM address HrTsStart
- Rx_SMB1_TIME_STAMP – One word at RAM address HrTsStart + 0x4
- CAN_HR_TIME_STAMPn registers – NUMBER_OF_MB words starting at RAM address HrTsStart + 0x8

Table below shows the RAM addresses and associated IPS peripheral bus addresses for the high-resolution time stamp region.

Table 20.110: CAN Start Address of High-resolution Time Stamp RAM Region

NUMBER_OF_MB	IRMQ_EN	FD_EN	Last Address of SMB Region	First Address of High-Resolution Time Stamp Region(HrTsStart)
128 (ENHANCE_MB_MEM = 0)	1	1	0x0B04	0x0B08
		0	0x0A5C	0x0A60
	0	1	0x0904	0x0908
		0	0x085C	0x0860
128 (ENHANCE_MB_MEM = 1)	1	1	0x1304	0x1308
	0	1	0x1104	0x1108
96	1	1	0x0884	0x0888
		0	0x07DC	0x07E0
	0	1	0x0704	0x0708

Yuntu Microelectronics

Table 20.110 continued from previous page

NUMBER_OF_MB	IRMQ_EN	FD_EN	Last Address of SMB Region	First Address of High-Resolution Time Stamp Region(HrTsStart)
		0	0x065C	0x0660
64	1	1	0x0604	0x0608
		0	0x055C	0x0560
	0	1	0x0504	0x0508
		0	0x045C	0x0460
32	1	1	0x0384	0x0388
		0	0x02DC	0x02E0
	0	1	0x0304	0x0308
		0	0x025C	0x0260
16	1	0	0x019C	0x01A0
	0	0	0x015C	0x0160

Table 20.111: CAN RAM Address Mapping for High-resolution Time Stamp Region

High-Resolution Time Stamp Sub-Region	NUMBER_OF_MB	IPS address	RAM Address
Rx_SMB0_TIME_STAMP	x	0x0C20	HrTsStart
Rx_SMB1_TIME_STAMP	x	0x0C24	HrTsStart + 0x4
CAN_HR_TIME_STAMPn	128	0x0C30-0xE2C	(HrTsStart + 0x8) - (HrTsStart + 0x8 + 0x1FC)
	96	0x0C30-0xDAC	(HrTsStart + 0x8) -(HrTsStart + 0x8 + 0x17C)
	64	0x0C30–0xD2C	(HrTsStart + 0x8) –(HrTsStart + 0x8 + 0xFC)
	32	0x0C30–0xCAC	(HrTsStart + 0x8) –(HrTsStart + 0x8 + 0x7C)
	16	0x0C30–0xC6C	(HrTsStart + 0x8) –(HrTsStart + 0x8 + 0x3C)

20.6.4.2.3 Enhanced Rx FIFO Region

The peripheral bus address range for the Enhanced Rx FIFO always starts at peripheral bus address 0x2000. The range of addresses allocated to the Enhanced Rx FIFO depends on:

- Enhanced Rx FIFO depth (ERX_FIFO_SIZE)
- CAN FD support (FD_EN)

The full peripheral bus address range for the Enhanced Rx FIFO is calculated as follows:

- If FD_EN = 1, the Enhanced Rx FIFO peripheral bus address range is 0x2000 through [0x2000 + (80 x ERX_FIFO_SIZE) - 4]
- If FD_EN = 0, the Enhanced Rx FIFO peripheral bus address range is 0x2000 through [0x2000 + (24 x ERX_FIFO_SIZE) - 4]

The RAM region allocated to the Enhanced Rx FIFO begins after the last RAM location used for the:

- SMBs when TIME_STAMP_EN = 0 (see “Mapping of peripheral bus address to CAN RAM for enhanced RX FIFO regions”)

Yuntu Microelectronics

- High-resolution time stamp region when TIME_STAMP_EN = 1 (see “Mapping of peripheral bus address to CAN RAM for high-resolution time stamp regions”)

Below Table shows the first RAM address of Enhanced Rx FIFO region (ErxFifoStart) for different FlexCAN configurations.

Table 20.112: CAN Start Address of Enhanced RX FIFO RAM Region

NUMBER_OF_MB	IRMQ_EN	FD_EN	TIME_STAMP_EN	First Address of Enhanced Rx FIFO Region(ErxFifoStart)
128 (ENHANCE_MB_MEM = 0)	1	1	1	0x0D10
			0	0x0B08
		0	1	0x0C68
			0	0x0A60
	0	1	1	0x0B10
			0	0x0908
		0	1	0x0A68
			0	0x0860
128 (ENHANCE_MB_MEM = 1)	1	1	1	0x1510
		1	0	0x1308
	0	1	1	0x1310
		1	0	0x1108
96	1	1	1	0x0A10
			0	0x0888
		0	1	0x0968
			0	0x07E0
	0	1	1	0x0890
			0	0x0708
		0	1	0x07E8
			0	0x0660
64	1	1	1	0x0710
			0	0x0608
		0	1	0x0668
			0	0x0560
	0	1	1	0x0610
			0	0x0508
		0	1	0x0568
			0	0x0460
32	1	1	1	0x0410
			0	0x0388

Yuntu Microelectronics

Table 20.112 continued from previous page

NUMBER_OF_MB	IRMQ_EN	FD_EN	TIME_STAMP_EN	First Address of Enhanced Rx FIFO Region(ErxFifoStart)
		0	1	0x0368
			0	0x02E0
	0	1	1	0x0390
			0	0x0308
		0	1	0x02E8
			0	0x0260
16	1	0	1	0x01E8
			0	0x01A0
	0	0	1	0x01A8
			0	0x0160

The full RAM address range for the Enhanced Rx FIFO is calculated as follows:

- If FD_EN = 1, the Enhanced Rx FIFO RAM bus address range is ErxFifoStart through [ErxFifoStart + (80 x ERX_FIFO_SIZE) - 4]
- If FD_EN = 0, the Enhanced Rx FIFO peripheral bus address range is ErxFifoStart through [ErxFifoStart + (24 x ERX_FIFO_SIZE) - 4]

“CAN RAM address mapping for enhanced RX FIFO (FD_EN=1)” shows the Enhanced Rx FIFO RAM address MAPPING for each value of ERX_FIFO_SIZE when FD_EN = 1. ”CAN RAM address mapping for enhanced RX FIFO (FD_EN=0)” shows the Enhanced Rx FIFO RAM address MAPPING for each value of ERX_FIFO_SIZE when FD_EN = 0.

NOTE: In Normal mode, only the oldest unread Enhanced Rx FIFO entry is accessible through the peripheral address bus, at peripheral bus addresses 0x2000–0x204C (FD_EN = 1) or 0x2000–0x2014 (FD_EN = 0).

Table 20.113: CAN RAM Address Mapping for Enhanced RX FIFO FD EN 1

ERX_FIFO_SIZE	IPS Address	Enhanved Rx FIFO RAM Address(Start)	Enhanved Rx FIFO RAM Address(End)
2	0x2000-0x209C	ErxFifoStart	ErxFifoStart + 0x009C
3	0x2000-0x20EC	ErxFifoStart	ErxFifoStart + 0x00EC
4	0x2000-0x213C	ErxFifoStart	ErxFifoStart + 0x013C
5	0x2000-0x218C	ErxFifoStart	ErxFifoStart + 0x018C
6	0x2000-0x21DC	ErxFifoStart	ErxFifoStart + 0x01DC
7	0x2000-0x222C	ErxFifoStart	ErxFifoStart + 0x022C
8	0x2000-0x227C	ErxFifoStart	ErxFifoStart + 0x027C
9	0x2000-0x22CC	ErxFifoStart	ErxFifoStart + 0x02CC
10	0x2000-0x231C	ErxFifoStart	ErxFifoStart + 0x031C
11	0x2000-0x236C	ErxFifoStart	ErxFifoStart + 0x036C
12	0x2000-0x23BC	ErxFifoStart	ErxFifoStart + 0x03BC
13	0x2000-0x240C	ErxFifoStart	ErxFifoStart + 0x040C
14	0x2000-0x245C	ErxFifoStart	ErxFifoStart + 0x045C

Yuntu Microelectronics

Table 20.113 continued from previous page

ERX_FIFO_SIZE	IPS Address	Enhanved Rx FIFO RAM Address(Start)	Enhanved Rx FIFO RAM Address(End)
15	0x2000-0x24AC	ErxFifoStart	ErxFifoStart + 0x04AC
16	0x2000-0x24FC	ErxFifoStart	ErxFifoStart + 0x04FC
17	0x2000-0x254C	ErxFifoStart	ErxFifoStart + 0x054C
18	0x2000-0x259C	ErxFifoStart	ErxFifoStart + 0x059C
19	0x2000-0x25EC	ErxFifoStart	ErxFifoStart + 0x05EC
20	0x2000-0x263C	ErxFifoStart	ErxFifoStart + 0x063C
21	0x2000-0x268C	ErxFifoStart	ErxFifoStart + 0x068C
22	0x2000-0x26DC	ErxFifoStart	ErxFifoStart + 0x06DC
23	0x2000-0x272C	ErxFifoStart	ErxFifoStart + 0x072C
24	0x2000-0x277C	ErxFifoStart	ErxFifoStart + 0x077C
25	0x2000-0x277C	ErxFifoStart	ErxFifoStart + 0x07CC
26	0x2000-0x281C	ErxFifoStart	ErxFifoStart + 0x081C
27	0x2000-0x286C	ErxFifoStart	ErxFifoStart + 0x086C
28	0x2000-0x28BC	ErxFifoStart	ErxFifoStart + 0x08BC
29	0x2000-0x290C	ErxFifoStart	ErxFifoStart + 0x090C
30	0x2000-0x295C	ErxFifoStart	ErxFifoStart + 0x095C
31	0x2000-0x29AC	ErxFifoStart	ErxFifoStart + 0x09AC
32	0x2000-0x29FC	ErxFifoStart	ErxFifoStart + 0x09FC

Table 20.114: CAN RAM Address Mapping for Enhanced RX FIFO FD EN 0

ERX_FIFO_SIZE	IPS Address	Enhanved Rx FIFO RAM Address(Start)	Enhanved Rx FIFO RAM Address(End)
2	0x2000-0x202C	ErxFifoStart	ErxFifoStart + 0x002C
3	0x2000-0x2044	ErxFifoStart	ErxFifoStart + 0x0044
4	0x2000-0x205C	ErxFifoStart	ErxFifoStart + 0x005C
5	0x2000-0x2074	ErxFifoStart	ErxFifoStart + 0x0074
6	0x2000-0x208C	ErxFifoStart	ErxFifoStart + 0x008C
7	0x2000-0x20A4	ErxFifoStart	ErxFifoStart + 0x00A4
8	0x2000-0x20BC	ErxFifoStart	ErxFifoStart + 0x00BC
9	0x2000-0x20D4	ErxFifoStart	ErxFifoStart + 0x00D4
10	0x2000-0x20EC	ErxFifoStart	ErxFifoStart + 0x00EC
11	0x2000-0x2104	ErxFifoStart	ErxFifoStart + 0x0104
12	0x2000-0x211C	ErxFifoStart	ErxFifoStart + 0x011C
13	0x2000-0x2134	ErxFifoStart	ErxFifoStart + 0x0134
14	0x2000-0x214C	ErxFifoStart	ErxFifoStart + 0x014C
15	0x2000-0x2164	ErxFifoStart	ErxFifoStart + 0x0164
16	0x2000-0x217C	ErxFifoStart	ErxFifoStart + 0x017C
17	0x2000-0x2194	ErxFifoStart	ErxFifoStart + 0x0194
18	0x2000-0x21AC	ErxFifoStart	ErxFifoStart + 0x01AC
19	0x2000-0x21C4	ErxFifoStart	ErxFifoStart + 0x01C4
20	0x2000-0x21DC	ErxFifoStart	ErxFifoStart + 0x01DC
21	0x2000-0x21F4	ErxFifoStart	ErxFifoStart + 0x01F4
22	0x2000-0x220C	ErxFifoStart	ErxFifoStart + 0x020C

Yuntu Microelectronics

Table 20.114 continued from previous page

ERX_FIFO_SIZE	IPS Address	Enhanved Rx FIFO RAM Address(Start)	Enhanved Rx FIFO RAM Address(End)
23	0x2000-0x2224	ErxFifoStart	ErxFifoStart + 0x0224
24	0x2000-0x223C	ErxFifoStart	ErxFifoStart + 0x023C
25	0x2000-0x2254	ErxFifoStart	ErxFifoStart + 0x0254
26	0x2000-0x226C	ErxFifoStart	ErxFifoStart + 0x026C
27	0x2000-0x2284	ErxFifoStart	ErxFifoStart + 0x0284
28	0x2000-0x229C	ErxFifoStart	ErxFifoStart + 0x029C
29	0x2000-0x22B4	ErxFifoStart	ErxFifoStart + 0x02B4
30	0x2000-0x22CC	ErxFifoStart	ErxFifoStart + 0x02CC
31	0x2000-0x22E4	ErxFifoStart	ErxFifoStart + 0x02E4
32	0x2000-0x22FC	ErxFifoStart	ErxFifoStart + 0x02FC

20.6.4.2.4 Enhanced Rx FIFO Filter Elements Region

The peripheral bus address range for the Enhanced Rx FIFO filter elements always starts at peripheral bus address 0x3000. The range of addresses allocated to the Enhanced Rx FIFO depends on the number of Enhanced Rx FIFO filter elements (ERX_FIFO_FLT_ELEM).

The full peripheral bus address range for the Enhanced Rx FIFO filter elements is: 0x3000 through [0x3000 + (4 x ERX_FIFO_FLT_ELEM) - 4].

As shown in Figure ”Mapping of peripheral bus address to CAN RAM for enhanced RX FIFO regions”, the RAM region allocated to the Enhanced Rx FIFO filter elements begins after the last RAM location used for the Enhanced Rx FIFO.

Example: Full FlexCAN Configuration

For a full FlexCAN configuration with NUMBER_OF_MB = 128, ENHANCE_MB_MEM = 1, IRMQ_EN = 1, FD_EN = 1, TIME_STAMP_EN = 1, ERX_FIFO_SIZE = 32, and ERX_FIFO_FLT_ELEM = 128, the last RAM address used by the Enhanced Rx FIFO is 0x1F0C, as shown in Table “CAN RAM address mapping for enhanced RX FIFO (FD_EN=1)”. The first RAM address allocated to the Enhanced Rx FIFO filter elements (ErxFifoFltElemStart) is therefore 0x1F10.

The full RAM address range for the Enhanced Rx FIFO filter elements is: ErxFifoFltElemStart through [ErxFifoFltElemStart + (4 x ERX_FIFO_FLT_ELEM) - 4].

For the full FlexCAN configuration described above, the Enhanced Rx FIFO filter elements RAM region uses:

- Peripheral bus addresses 0x3000–0x31FC
- RAM addresses 0x1F10–0x210C

Example: Full FlexCAN Configuration minus ENHANCE_MB_MEM and TIME_STAMP_EN

For a FlexCAN configuration with NUMBER_OF_MB = 128, ENHANCE_MB_MEM = 0, IRMQ_EN = 1, FD_EN = 1, TIME_STAMP_EN = 0, ERX_FIFO_SIZE = 32, and ERX_FIFO_FLT_ELEM = 128, the last RAM address used by the Enhanced Rx FIFO is 0x1504, as shown in Table ”CAN RAM address mapping for enhanced RX FIFO (FD_EN=1)”. The first RAM address allocated to the Enhanced Rx FIFO filter elements (ErxFifoFltElemStart) is therefore 0x1508.

The full RAM address range for the Enhanced Rx FIFO filter elements is: ErxFifoFltElemStart through [ErxFifoFltElemStart + (4 x ERX_FIFO_FLT_ELEM) - 4].

Yuntu Microelectronics

For the full configuration described above and the maximum of 128 Enhanced Rx FIFO filter elements, the Enhanced Rx FIFO filter elements RAM region uses:

- Peripheral bus addresses 0x3000–0x31FC
- RAM addresses 0x1508–0x1704

Yuntu Microelectronics

Chapter 21

Local Interconnect Network (LINFlexD)

21.1 Chip-specific LINFlexD Information

21.1.1 Instantiation Information

YTM32B1HA0x contains up to 10 instances of LINFlexD.

21.1.2 Signal Description

Table 21.1: LINFlexD Signal Table

Signal	Description	Direction
LINFlexD_TX	LINFlexD transmit line	O
LINFlexD_RX	LINFlexD receive line	I

21.1.3 Power Mode Discription

Table 21.2: LINFlexD Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Keeps working
Deepsleep	Keeps working
Standby	Disable
Powerdown	Disable

21.2 Introduction

21.2.1 General

The LINFlexD Controller is a semiconductor IP block supporting the Local Interconnect Network (LIN) protocol for both master and slave applications.

Yuntu Microelectronics

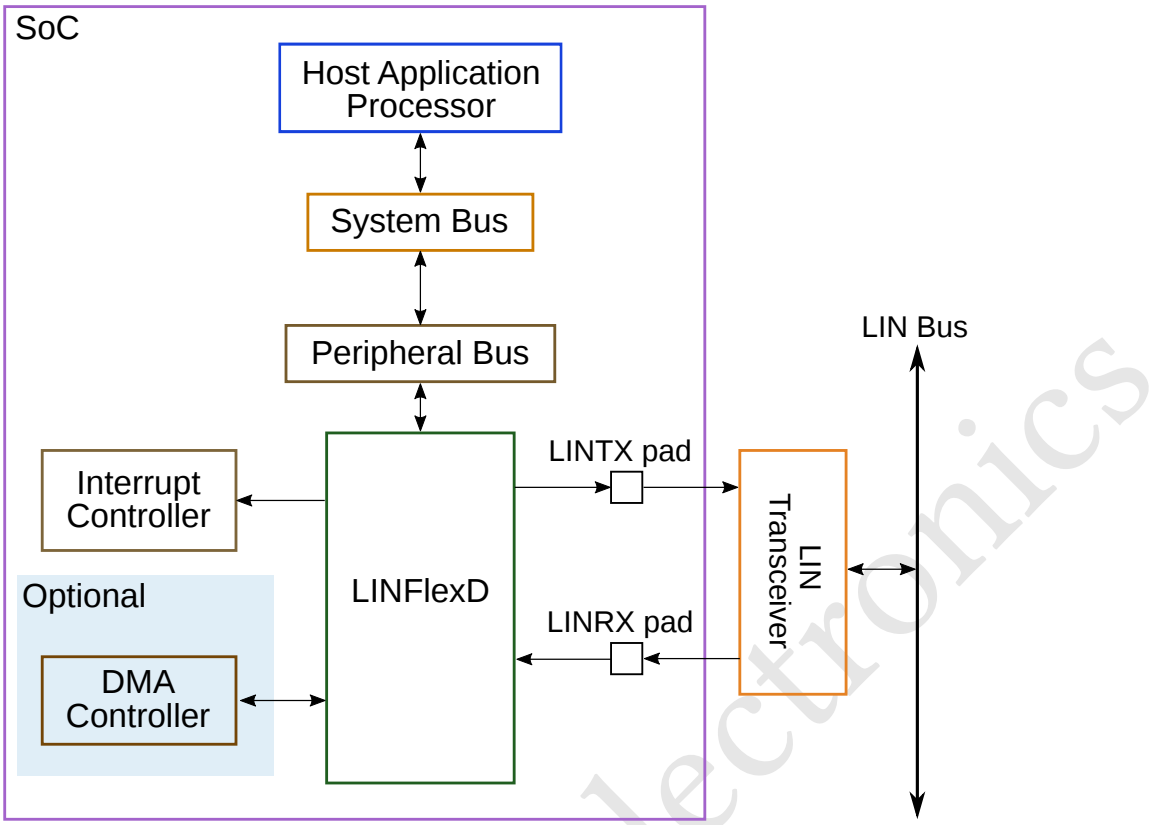


Figure 21.1: Example System Using LINFlexD

The LINFlexD manages a high number of LIN messages efficiently with a minimum of CPU load. To reduce the CPU loading in master mode, the LINFlexD autonomously handles the LIN messages after software has triggered the header transmission, until the next header transmission request in transmitter mode or until checksum reception in the receiver mode.

The LINFlexD supports LIN protocol versions 1.3, 2.0, 2.1, and 2.2, and provides an 8-byte buffer for transmission/reception data.

The LINFlexD also supports a UART mode for UART transfers of 8-bit, 9-bit, or 13-bit data frames.

Optional DMA support is available for both transmit (Tx) and receive (Rx) data.

21.2.2 Features

The features of the LINFlexD module are listed below.

- LINFlexD common features in both LIN and UART
  - Fractional baud rate generator
  - Three operating modes for power saving and configuration registers lock
    - Initialization
    - Normal
    - Sleep
  - Test mode: Loop Back
  - Maskable interrupts
- LIN mode features

Yuntu Microelectronics

  - Support for LIN protocol versions 1.3, 2.0, 2.1, and 2.2
  - Bit rates up to 20 Kbit/s (LIN protocol)
  - Master/slave modes
  - Classic and enhanced checksum calculation and check
  - Single 8-byte buffer or FIFO for transmission/reception
  - Timeout management
  - Identifier filters
  - DMA interface
  - Support for 16 identifiers
  - Master mode with autonomous message handling
  - Wakeup event on dominant bit detection
  - True LIN field state machine
  - Advanced LIN error detection
  - Header, response, and frame timeout
  - Slave mode
    - Autonomous header handling
    - Autonomous transmit/receive data handling
  - Identifier filters for autonomous message handling in slave mode
  - Separate clock for baud rate calculation
- UART mode features
  - Full-duplex communication
  - Separate clock for baud rate calculation
  - Baud rate is a function of baud clock, and the LINIBRR and LINFBRR registers.
  - 7/8 bits data, parity
  - 1/2/3 stop bits
  - 12-bit + parity reception
  - 4-byte buffer for reception; 4-byte buffer for transmission
  - 12-bit counter for timeout management
  - The maximum baud rate achievable is baud clock/4 Mbit/s
  - For bit rate ≤ baud clock/16 Mbit/s
    - 16 times oversampling
    - 3:1 majority voting
  - For baud clock/16 Mbit/s < bit rate ≤ baud clock/8 Mbit/s
    - Reduced oversampling programmable by software
    - 3:1 majority voting for reduced oversampling of 8 samples per bit
  - For baud clock/8 Mbit/s < bit rate ≤ baud clock/4 Mbit/s
    - Reduced oversampling programmable by software
    - 1:1 voting for all reduced oversampling of 4, 5, and 6 samples per bit

21.3 Register Definition

21.3.1 LINFlexD Memory Map

LINFlexD0 base address 4006c000h
LINFlexD1 base address 40070000h
LINFlexD2 base address 40074000h
LINFlexD3 base address 40078000h
LINFlexD4 base address 4007c000h

Yuntu Microelectronics

LINFlexD5 base address 40080000h
LINFlexD6 base address 40084000h
LINFlexD7 base address 40088000h
LINFlexD8 base address 4008c000h
LINFlexD9 base address 40090000h

Table 21.3: LINFlexD Memory Map

Offset	Register	Width	Access	Reset Value
0h	LIN Control Register 1 (LINCR1)	32	RW	00000082h
4h	LIN Interrupt Enable Register (LINIER)	32	RW	00000000h
8h	LIN Status Register (LINSR)	32	RW	00000040h
ch	LIN Error Status Register (LINESR)	32	RW	00000000h
10h	UART Mode Control Register (UARTCR)	32	RW	00000000h
14h	UART Mode Status Register (UARTSR)	32	RW	00000000h
18h	LIN Timeout Control Status Register (LINTCSR)	32	RW	00000200h
1ch	LIN Output Compare Register (LINOCR)	32	RW	0000FFFFh
20h	LIN Timeout Control Register (LINTOCR)	32	RW	00000E2Ch
24h	LIN Fractional Baud Rate Register (LINFBRR)	32	RW	00000000h
28h	LIN Integer Baud Rate Register (LINIBRR)	32	RW	00000000h
2ch	LIN Checksum Field Register (LINCFR)	32	RW	00000000h
30h	LIN Control Register 2 (LINCR2)	32	RW	00006000h
34h	Buffer Identifier Register (BIDR)	32	RW	00000000h
38h	Data in byte (DATA_DATA80)	8	RW	00h
39h	Data in byte (DATA_DATA81)	8	RW	00h
3ah	Data in byte (DATA_DATA82)	8	RW	00h
3bh	Data in byte (DATA_DATA83)	8	RW	00h
3ch	Data in byte (DATA_DATA84)	8	RW	00h
3dh	Data in byte (DATA_DATA85)	8	RW	00h
3eh	Data in byte (DATA_DATA86)	8	RW	00h
3fh	Data in byte (DATA_DATA87)	8	RW	00h
38h	Data in half-word (DATA_DATA160)	16	RW	0000h
39h	Data in half-word (DATA_DATA161)	16	RW	0000h
3ah	Data in half-word (DATA_DATA162)	16	RW	0000h
3bh	Data in half-word (DATA_DATA163)	16	RW	0000h
38h	Data in word (DATA_DATA320)	32	RW	00000000h
39h	Data in word (DATA_DATA321)	32	RW	00000000h
40h	Identifier Filter Enable Register (IFER)	32	RW	00000000h
44h	Identifier Filter Match Index (IFMI)	32	RO	00000000h
48h	Identifier Filter Mode Register (IFMR)	32	RW	00000000h
4ch	Identifier Filter Control Register (IFCR0)	32	RW	00000000h
50h	Identifier Filter Control Register (IFCR1)	32	RW	00000000h
54h	Identifier Filter Control Register (IFCR2)	32	RW	00000000h
58h	Identifier Filter Control Register (IFCR3)	32	RW	00000000h
5ch	Identifier Filter Control Register (IFCR4)	32	RW	00000000h
60h	Identifier Filter Control Register (IFCR5)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
64h	Identifier Filter Control Register (IFCR6)	32	RW	00000000h
68h	Identifier Filter Control Register (IFCR7)	32	RW	00000000h
6ch	Identifier Filter Control Register (IFCR8)	32	RW	00000000h
70h	Identifier Filter Control Register (IFCR9)	32	RW	00000000h
74h	Identifier Filter Control Register (IFCR10)	32	RW	00000000h
78h	Identifier Filter Control Register (IFCR11)	32	RW	00000000h
7ch	Identifier Filter Control Register (IFCR12)	32	RW	00000000h
80h	Identifier Filter Control Register (IFCR13)	32	RW	00000000h
84h	Identifier Filter Control Register (IFCR14)	32	RW	00000000h
88h	Identifier Filter Control Register (IFCR15)	32	RW	00000000h
8ch	Global Control Register (GCR)	32	RW	00000000h
90h	UART Preset Timeout Register (UARTPTO)	32	RW	00000FFFh
94h	UART Current Timeout Register (UARTCTO)	32	RO	00000000h
98h	DMA Tx Enable Register (DMATXE)	32	RW	00000000h
9ch	DMA Rx Enable Register (DMARXE)	32	RW	00000000h

21.3.1.1 LINFlexD LINCR1 Register

The LINCR1 register consists of control bits used to configure features of the LINFlexD.

NOTE. When accessing the LINCR1 register, each reserved bit should be written with its original reset value.

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																NLSE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CCD	CFD	LASE	AUTO WU	MBL				BF		LBKM	MME	SSBL	RBLM	SLEEP	INIT
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0

Table 21.4: LINFlexD LINCR1 Register Description

Field	Function
16	**LIN State Capture Enable on Bit Error**

Yuntu Microelectronics

Field	Function
NLSE	NLSE enables/disables capture of the LIN state (LINSR.LINS) whenever a bit error flag occurs (LINESR.BEF is set to 1). 0b - LINSR.LINS shows the current LIN state and does not capture the LIN state on bit error. 1b - LINSR.LINS is captured when a bit error flag occurs. **NOTE**. If NLSE = 1 and LINCR2.IOBE = 1, the internal LIN state is not captured when state transition to Idle occurs. LINSR.LINS shows the state in which the bit error occurred.
15 CCD	**Checksum Calculation Disable** 0b - Checksum calculation is done by hardware. The LINCFR register is read-only. 1b - Checksum calculation is disabled. The LINCFR register is read/write. If the checksum field is enabled (CFD = 0), LINCFR can be programmed to send a software-calculated checksum/CRC. CCD can be read in any mode, but can be written only during Initialization mode.
14 CFD	**Checksum Field Disable** 0b - Checksum field is sent after the required number of data bytes are sent. 1b - No checksum field is sent in the frame. CFD can be read in any mode, but can be written only during Initialization mode.
13 LASE	**LIN Autosynchronization Enable** Enables the autosynchronization, feature described in “Automatic Resynchronization”. 0b - Disable autosynchronization. 1b - Enable autosynchronization. LASE can be read in any mode, but can be written only during Initialization mode.
12 AUTOWU	**Automatic Wake-Up** 0b - The SLEEP bit is cleared by software only. 1b - The SLEEP bit is cleared by hardware whenever the LINSR.WUF bit is set. AUTOWU is effective in both LIN and UART modes. AUTOWU can be read in any mode, but can be written only during Initialization mode.
11 - 8	**Master Break Length**

Yuntu Microelectronics

Field	Function
MBL	Controls the length of the break field generated in LIN master mode. 0000b - 10-bit break length 0001b - 11-bit break length 0010b - 12-bit break length 0011b - 13-bit break length 0100b - 14-bit break length 0101b - 15-bit break length 0110b - 16-bit break length 0111b - 17-bit break length 1000b - 18-bit break length 1001b - 19-bit break length 1010b - 20-bit break length 1011b - 21-bit break length 1100b - 22-bit break length 1101b - 23-bit break length 1110b - 36-bit break length 1111b - 50-bit break length MBL can be read in any mode, but can be written only during Initialization mode.
7 BF	**Bypass Filter** Controls the receive filter bypass function. 0b - Receiver ignores incoming frame if ID does not match any ID filter or if no ID filters are active. 1b - If no ID filters are active, receiver responds to incoming frames. If ID filters are active, receiver responds if frame is received but does not match any filter. BF can be read in any mode, but can be written only during Initialization mode.
5 LBKM	**Loop Back Mode** Enables or disables Loop Back mode as described in “Loop Back Mode”. 0b - Disable Loop Back mode. 1b - Enable Loop Back mode. LBKM is effective in both LIN and UART modes. LBKM can be read in any mode, but can be written only during Initialization mode.
4 MME	**Master Mode Enable** 0b - Slave mode. 1b - Master mode. MME can be read in any mode, but can be written only during Initialization mode.
3 SSBL	**Slave Mode Sync Break Length** Sets the number of bit times for break field detection in slave mode. 0b - 11-bit break length. 1b - 10-bit break length. SSBL can be read in any mode, but can be written only during Initialization mode.
2	**Receiver Buffer Locked Mode**

Yuntu Microelectronics

Field	Function
RBLM	0b - Receiver is buffer is not locked. The next incoming message overwrites the previous one. 1b - Receiver buffer is locked against overrun. Once the buffer is full, the next incoming message is discarded if the buffer is not released by software clearing the LINSR.RMB bit. RBLM is effective in both LIN and UART modes. RBLM can be read in any mode, but can be written only during Initialization mode.
1 SLEEP	**Sleep Mode Request** Set by software to request LINFlexD to enter Sleep mode. The SLEEP bit is cleared by software to exit sleep mode. If LINCR1.AUTOWU and LINSR.WUF are set, SLEEP is automatically cleared by hardware to exit sleep mode. SLEEP is effective in both LIN and UART modes.
0 INIT	**Initialization Mode Request** Set by software to switch the hardware into Initialization mode. When software clears the INIT bit (and if the SLEEP bit is also 0), the LINFlexD enters Normal mode. INIT is effective in both LIN and UART modes.

21.3.1.2 LINFlexD LINIER Register

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W															REDIE	SDIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SZIE	OCIE	BEIE	CEIE	HEIE			FEIE	BOIE	LSIE	WUIE		TOIE	DRIE	DTIE	HRIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.5: LINFlexD LINIER Register Description

Field	Function
17	**Receive Edge Detection Interrupt Enable**

Yuntu Microelectronics

Field	Function
REDIE	0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Receive Edge Detection Flag (REDF) is set in LINSR.
16 SDIE	**Start Detection Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Start Detection Flag (SDF) is set in LINSR.
15 SZIE	**Stuck at Zero Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Stuck at Zero Flag (SZF) is set in LINESR or UARTSR.
14 OCIE	**Output Compare Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Output Compare Flag (OCF) is set in LINESR or UARTSR.
13 BEIE	**Bit Error Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Bit Error Flag (BEF) is set in LINESR.
12 CEIE	**Checksum Error Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Checksum Error Flag (CEF) is set in LINESR.
11 HEIE	**Header Error Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the any of the following flags are set in LINESR: SFEF, SDEF, IDPEF.
8 FEIE	**Frame Error Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Frame Error Flag (FEF) is set in LINESR or UARTSR.
7 BOIE	**Buffer Overrun Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated when the Buffer Overrun Flag (BOF) is set in LINESR or UARTSR.
6	**LIN State Interrupt Enable**

Yuntu Microelectronics

Field	Function
LSIE	0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt generated on entering the following states: Sync Del, Sync Field, Identifier Field, Checksum.
5 WUIE	**Wakeup Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt generated when the Wakeup Flag (WUF) is set in LINESR or UARTSR.
3 TOIE	**Timeout Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt generated when the LINFlexD is in UART mode and the Timeout bit (TO) is set in UARTSR.
2 DRIE	**Data Reception Complete Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt generated when the Data Received Flag (DRF) is set in LINESR or UARTSR.
1 DTIE	**Data Transmitted Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt generated when the Data Transmitted Flag (DTF) is set in LINESR or UARTSR.
0 HRIE	**Header Received Interrupt Enable** 0b - Interrupt disabled. 1b - Interrupt enabled. An interrupt is generated Header Received Flag (HRF) is set in LINSR.

21.3.1.3 LINFlexD LINSR Register

The LINSR register contains status bits indicating the state of the LINFlexD hardware.

NOTE. Reset value of RDI reflects the LINFlexD_RX pin state.

Table 21.6: LINS encodings

LINS[3:0]	LIN State	Description
0000b	Sleep mode	The LINFlexD is in Sleep mode to reduce power consumption.
0001b	Init mode	The LINFlexD is in Initialization mode.
0010b	Idle	The LINFlexD enters the Idle state when: · LINCR1.SLEEP and LINCR1.INIT are cleared by software. · A wakeup pulse is received LINFlexD_RX and LINCR1.AUTOWU is set. · The previous frame transmission or reception is completed or aborted.

Yuntu Microelectronics

Table 21.6 continued from previous page

LINS[3:0]	LIN State	Description
0011b	Break	In slave mode, a falling edge followed by a dominant state was detected. The LINFlexD is receiving a break field. In master mode, break field transmission is ongoing. **NOTE**. In slave mode, upon any error, the LIN state is either Idle or Break, depending on the last bit detected on LINFlexD_RX. If last bit detected is dominant, the new LIN state is Break; otherwise, the new LIN state is Idle.
0100b	Break Delimiter	In slave mode, a valid break field was detected (10 or 11 bit times depending on LINCR1.SSBL) and the LINFlexD is waiting for a rising edge. In master mode, break field transmission is complete, break delimiter transmission is ongoing.
0101b	Sync Field	In slave mode, a valid break delimiter was detected (recessive state for at least one bit time) and the LINFlexD is receiving a sync byte field. In master mode, sync byte field transmission is ongoing.
0110b	Identifier Field	In slave mode, a valid sync field was received and the LINFlexD is receiving an identifier filed. In master mode, identifier field transmission is ongoing.
0111b	Header Reception/ Transmission	In slave mode, a valid header was received and the identifier field is available in BIDR. In master mode, header transmission is complete.
1000b	Data Reception/ Transmission	In receiver mode, reception is ongoing. In transmitter mode, response transmission is ongoing.
1001b	Checksum	Data transmission/reception is complete. Checksum transmission/reception is ongoing.

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R											REDF		AUTOSYNC_COMP	RDC		
W											W1C		W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LINS						RMB	DRBNE	RXBUSY	RDI	WUF			DRF	DTF	HRF
W							W1C	W1C			W1C			W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Table 21.7: LINFlexD LINSR Register Description

Field	Function
21	**Receive Edge Detection Flag**

Yuntu Microelectronics

Field	Function
REDF	REDF is set by hardware when receiving falling edge is detected.
19 AUTOSYNC _COMP	**Autosynchronization Complete** AUTOSYNC_COMP is set when autosynchronization is enabled (LINCR1.LASE is set) and autosynchronization is complete. After AUTOSYNC_COMP is set, the contents of LINIBRR and LINFBRR registers can be read.
18 - 16 RDC	**Receive Data Byte Count** RDC contains the number of entries (bytes) in the receive data buffer in LIN mode. 000b - 1 byte 001b - 2 bytes 010b - 3 bytes 011b - 4 bytes 100b - 5 bytes 101b - 6 bytes 110b - 7 bytes 111b - 8 bytes RDC is a read-only field available for debug purposes.
15 - 12 LINS	**LIN State.** In UART mode, the Idle, Init, Sleep, and Data Transmission/Reception states are flagged by the LIN status bits. See LINS encodings for details. **NOTE**. The value of the LINS bit field does not change by any write operation to it. Writing 0xF to LINS clears the Rx interrupt only when set due to the LIN state event.
9 RMB	**Release Message Buffer** 0b - Buffer is free. 1b - Buffer data is ready to be read by software. RMB should be cleared by software after reading the data received in the buffer. RMB is cleared by hardware in Initialization mode.
8 DRBNE	**Data Reception Buffer Not Empty Flag** DRBNE is set by hardware as soon as the first byte of response is received and stored in DATA (when there is at least one data byte in reception buffer). Software should clear DRBNE after reading all the buffers. DRBNE can be checked by software in the case of a response timeout event. DRBNE is cleared by hardware in Initialization mode.
7 RXBUSY	**Receiver Busy Flag** 0b - Receiver is idle. 1b - Reception is ongoing. **NOTE**. In slave mode, after header reception, if BIDR.DIR = 0 and reception starts, the RXBUSY bit is set. In this case, software cannot set the LINCR2.DTRQ bit.
6	**Receiver Data Input**

Yuntu Microelectronics

Field	Function
RDI	Indicates the current state of the LINFlexD_RX pin. **NOTE**. After reset is released, RDI reflects the actual value of the LINFlexD_RX pin.
5 WUF	**Wakeup Flag** WUF is set by hardware when a falling edge is detected on the LINFlexD_RX pin in either of the following conditions: • The LINFlexD is in slave mode and in Sleep mode. • The LINFlexD is in master mode is in either Sleep mode or Idle state. WUF is cleared by hardware in Initialization mode.
2 DRF	**Data Reception Completed Flag** DRF is set by hardware and indicates that data reception has completed. DRF is cleared by hardware in Initialization mode. **NOTE**. DRF is not set when a framing error or checksum error occurs.
1 DTF	**Data Transmission Completed Flag** DTF is set by hardware and indicates that data transmission has completed. DTF is cleared by hardware in Initialization mode. **NOTE**. DTF is not set in LIN mode when a bit error occurs and LINCR2.IOBE = 0.
0 HRF	**Header Received Flag** HRF is set by hardware when header reception completes and one of the following conditions is true. • All filters are inactive and LINCR1.BF = 1. • No match in any filter and LINCR1.BF = 1. • Tx filter match. HRF is reset by hardware in Initialization mode and at end of completed or aborted frame.

Yuntu Microelectronics

21.3.1.4 LINFlexD LINESR Register

The LINESR register contains status bits for various error conditions. For detailed descriptions the error conditions, see “Errors” .

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SZF	OCF	BEF	CEF	SFEF	SDEF	IDPEF	FEF	BOF							NF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C							W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.8: LINFlexD LINESR Register Description

Field	Function
15 SZF	**Stuck at Zero Flag** Set by hardware when a stuck-at-zero timeout error occurs.
14 OCF	**Output Compare Flag** In master mode, OCF is set when counter LINTCSR.CNT matches the content of LINOCR.OC2. In slave mode, OCF is set when the content of the counter LINTCSR.CNT matches the content of LINOCR.OC1 or LINOCR.OC2. If OCF is set when LINTCSR.MODE = 0 and LINTCSR.IOT = 1, the LINFlexD goes to Idle state. When LINTCSR.MODE = 0, OCF is cleared by hardware in Initialization mode. When LINTCSR.MODE = 1, OCF maintains its status regardless of the LIN state.
13 BEF	**Bit Error Flag** BEF is set by hardware when the LINFlexD detects a bit error. BEF is cleared by hardware in Initialization mode.
12 CEF	**Checksum Error Flag** CEF is set by hardware when the received checksum does not match the hardware calculated checksum. CEF is cleared by hardware in Initialization mode. **NOTE**. CEF is never set if LINCR1.CCD or LINCR1.CFD is set.
11 SFEF	**Sync Field Error Flag** SFEF is set by hardware when a sync field error occurs due to an inconsistent sync field. SFEF is cleared by hardware in Initialization mode.
10	**Break Delimiter Error Flag**

Yuntu Microelectronics

Field	Function
SDEF	SDEF is set by hardware when a break delimiter error occurs because a break delimiter is too short (less than one bit time). SDEF is cleared by hardware in Initialization mode.
9 IDPEF	**ID Parity Error Flag** IDPEF is set by hardware LINFlexD detects an error in the ID parity. IDPEF is cleared by hardware in Initialization mode.
8 FEF	**Framing Error Flag** FEF is set by hardware when a framing error (invalid stop bit) occurs. FEF is cleared by hardware in Initialization mode.
7 BOF	**Buffer Overrun Flag** BOF is set by hardware when a new byte is received and LINSR.RMB bit is not cleared. BOF is cleared by hardware in Initialization mode.
0 NF	**Noise Flag** NF is set by hardware when noise is detected in a receive character. NF is cleared by hardware in Initialization mode.

21.3.1.5 LINFlexD UARTCR Register

The UARTCR register contains control bits for UART mode.

NOTE. Always write ‘0’ to reserved bits when writing to UARTCR.

NOTE. In UART mode, the LINFlexD does not support communication with Special Word Length (UARTCR.WLS = 1) in buffer mode.

NOTE. In UART mode, the LINFlexD supports communication with Special Word Length (UARTCR.WLS = 1) in FIFO mode only with UARTCR.WL[1:0] = 11.

Offset: 10h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MIS	CSP			OSR				ROSE	NEF			DTU	SBUR		WLS
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TDFL_TFC			RDFL_RFC			RFBM	TFBM	WL1	PC1	RxEn	TxEn	PC0	PCE	WL0	UART
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.9: LINFlexD UARTCR Register Description

Field	Function
31 MIS	**Monitor Idle State** 0b - UARTCTO monitors the number of bits to be received. 1b - UARTCTO monitors the idle state of the reception line. MIS can be read in any mode, but can be written only during Initialization mode.
30 - 28 CSP	**Configurable Sample Point** The CSP bits select the sample point during reduced oversampling. CSP can have the following range of values for the specified oversampling rate: OSR / CSP: 4 → 2, 3; 5 → 2, 3, 4; 6 → 3, 4, 5; 8 → N/A CSP can be read in any mode, but can be written only during Initialization mode.
27 - 24 OSR	**Oversampling Rate** Selects the number of samples taken for a bit when reduced oversampling is enabled. The values allowed for OSR are 4, 5, 6, and 8. **NOTE**. When idle state monitoring is enabled (MIS = 1), the allowed OSR values are 4 and 8. OSR can be read in any mode, but can be written only during Initialization mode.
23 ROSE	**Reduced Oversampling Enable** 0b - Each bit is oversampled sixteen times. 1b - OSR bits determine the oversampling rate. ROSE can be read in any mode, but can be written only during Initialization mode.
22 - 20	**Number of Expected Frames**

Yuntu Microelectronics

Field	Function
NEF	The NEF bits set the number of expected frames in UART reception mode. If the DTU bit is set, then the UART timeout counter is reset after the configured number of frames have been received. NEF can be read in any mode, but can be written only during Initialization mode.
19 DTU	**Disable Timeout in UART mode** 0b - Timeout must be handled by software. 1b - Timeout in UART mode is disabled after the configured number of data frames are received. There is an internal counter that counts the number of frames received. **In Buffer mode:** The counter is reset whenever the number of bytes received (rec_byte_cnt) is equal to RDFL. At that time, the DRF bit is set, and num_of_frames (number of frames received) and rec_byte_cnt are reset. When num_of_frames is equal to NEF, disable_timeout is generated. However, this clears num_of_frames. So, the timer restarts again. The value of counter is 0 when it restarts. **In FIFO mode:** When num_of_frames is equal to NEF, disable_timeout is generated, which clears num_of_frames and causes the timer to restart again. The value of counter is 0 when it restarts. **NOTE**. • disable_timeout causes the timer to restart, which enables timeout again. • Timer reset means it resets and starts counting again and the timeout is enabled. DTU can be programmed in Initialization mode only when the UART bit is set.
18 - 17 SBUR	**Number of Stop Bits in UART Reception Mode** 00b - 1 stop bit 01b - 2 stop bits 10b - 3 stop bits 11b - Reserved When the UART is used for transmission and reception, set the same number of stop bits in GCR.STOP and UARTCR.SBUR. When the UART is used only as receiver, only the SBUR bits need to be set the specify the number of stop bits. SBUR can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.
16 WLS	**Special Word Length in UART Mode** 0b - Disable 12-bit + parity bit in reception. 1b - Enable 12-bit + parity bit in reception (UART mode). When WLS = 1, the WL and PCE bits have no effect in UART reception, although parity check is enabled by default in this mode and PC0/1 bits can be used to select the parity control. The WLS bit has priority in UART reception mode. WLS can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.
15 - 13	**Transmitter Data Field Length / Tx FIFO Counter**

Yuntu Microelectronics

Field	Function
TDFL_TFC	The TDFL_TFC field has one of two functions depending on the mode of operation. **UART Buffer Mode:** When the LINFlexD is in UART buffer mode (TFBM = 0), TDFL_TFC is read/write and defines the number of bytes to be transmitted. • x00: 1 byte • x01: 2 bytes • x10: 3 bytes • x11: 4 bytes The first bit is reserved and not implemented. When the UART data length is configured as half-word (WL = 10 or 11), the only valid values for TDFL_TFC are x01 and x11. **UART FIFO Mode:** When the LINFlexD is in UART FIFO mode (TFBM = 1), TDFL_TFC is read-only and contains the number of entries (bytes) in the Tx FIFO. • 000: Empty • 001: 1 byte • 010: 2 bytes • 011: 3 bytes • 100: 4 bytes • All other values: Reserved TDFL_TFC can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.
12 - 10	**Receiver Data Field Length / Rx FIFO Counter**

Yuntu Microelectronics

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>RDFL_RFC</td><td>The RDFL_RFC field has one of two functions depending on the mode of operation.
UART Buffer Mode:
When the LINFlexD is in UART buffer mode (RFBM = 0), RDFL_RFC is read/write and defines the number of bytes to be received:
• x00: 1 byte
• x01: 2 bytes
• x10: 3 bytes
• x11: 4 bytes
The first bit is reserved and not implemented.
When the UART data length is configured as half-word (WL = 10 or 11), the only valid values for RDFL_RFC are x01 and x11.
RDFL_RFC should be programmed to be greater than or equal to NEF (number of expected frames).
UART FIFO Mode:
When the LINFlexD is in UART FIFO mode (RFBM = 1), RDFL_RFC is read-only and contains the number of entries (bytes) in the Rx FIFO:
• 000: Empty
• 001: 1 byte
• 010: 2 bytes (1.5 byte if WLS = 1)
• 011: 3 bytes
• 100: 4 bytes (2x1.5 bytes if WLS = 1)
• All other values: Reserved
RDFL_RFC can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.</td></tr>
<tr><td>9
RFBM</td><td>Rx FIFO / Buffer Mode
0b - Rx buffer mode enabled.
1b - Rx FIFO mode enabled (mandatory in DMA Rx mode).
RFBM can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.</td></tr>
<tr><td>8
TFBM</td><td>Tx FIFO / Buffer Mode
0b - Tx buffer mode enabled.
1b - Tx FIFO mode enabled (mandatory in DMA Tx mode).
TFBM can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.</td></tr>
<tr><td>7
WL1</td><td>Word Length 1 in UART Mode
<table>
<tr><th>WL1</th><th>WL0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>7 bits data + parity</td></tr>
<tr><td>0</td><td>1</td><td>8 bits data when PCE = 0 or
8 bits data + parity when PCE = 1</td></tr>
<tr><td>1</td><td>0</td><td>15 bits data + parity</td></tr>
<tr><td>1</td><td>1</td><td>16 bits data when PCE = 0 or
16 bits data + parity when PCE = 1</td></tr>
</table>
WL1 can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.</td></tr>
<tr><td>6</td><td>Parity Control 1</td></tr>
</table>

Yuntu Microelectronics

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>PC1</td><td>
<table>
<tr><th>PC1</th><th>PC0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Parity sent is even.</td></tr>
<tr><td>0</td><td>1</td><td>Parity sent is odd.</td></tr>
<tr><td>1</td><td>0</td><td>A logic 0 is always transmitted/
checked as parity bit.</td></tr>
<tr><td>1</td><td>1</td><td>A logic 1 is always transmitted/
checked as parity bit.</td></tr>
</table>
PC1 can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.
</td></tr>
<tr><td>5
RxEn</td><td>Receiver Enable
0b - Receiver disabled.
1b - Receiver enabled.
RxEn is writable only when the UART bit is set.</td></tr>
<tr><td>4
TxEn</td><td>Transmitter Enable
0b - Transmitter disabled.
1b - Transmitter enabled.
Transmission starts when TxEn is set and Data Byte 0 (DATA0) is programmed.
TxEn is writable only when the UART bit is set.</td></tr>
<tr><td>3
PC0</td><td>Parity Control 0
<table>
<tr><th>PC1</th><th>PC0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>Parity sent is even.</td></tr>
<tr><td>0</td><td>1</td><td>Parity sent is odd.</td></tr>
<tr><td>1</td><td>0</td><td>A logic 0 is always transmitted/
checked as parity bit.</td></tr>
<tr><td>1</td><td>1</td><td>A logic 1 is always transmitted/
checked as parity bit.</td></tr>
</table>
PC0 can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.
</td></tr>
<tr><td>2
PCE</td><td>Parity Control Enable
0b - Parity transmit/check disabled.
1b - Parity transmit/check enabled.
PCE can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.</td></tr>
<tr><td>1
WL0</td><td>Word Length 0 in UART Mode
<table>
<tr><th>WL1</th><th>WL0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>7 bits data + parity</td></tr>
<tr><td>0</td><td>1</td><td>8 bits data when PCE = 0 or
8 bits data + parity when PCE = 1</td></tr>
<tr><td>1</td><td>0</td><td>15 bits data + parity</td></tr>
<tr><td>1</td><td>1</td><td>16 bits data when PCE = 0 or
16 bits data + parity when PCE = 1</td></tr>
</table>
WL0 can be read in any mode, but can be written only during Initialization mode, when the UART bit is set.
</td></tr>
</table>

Yuntu Microelectronics

Field	Function
0 UART	**UART Mode Enable** 0b - LIN mode 1b - UART mode UART can be read in any mode, but can be written only during Initialization mode.

21.3.1.6 LINFlexD UARTSR Register

The UARTSR register contains status bits for UART mode.

Offset: 14h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																SDF
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SZF	OCF	PE				RMB	FEF	BOF	RDI	WUF	RFNE	TO	DRF_RFE	DTF_TFF	NF
W	W1C	W1C	W1C				W1C	W1C	W1C		W1C		W1C			W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.10: LINFlexD UARTSR Register Description

Field	Function
16 SDF	**Start Detection Flag** SDF is set by hardware when receiving start bit is detected.
15 SZF	**Stuck at Zero Flag** SZF is set by hardware when 100 dominant bits are detected. An interrupt is generated if LINIER.SZIE = 1. SZF reflects the same value as LINESR.SZF when the LINFlexD is not in Initialization mode and UARTCR.UART = 1.
14 OCF	**Output Compare Flag** 0b - No output compare event occurred. 1b - The content of the timeout counter matched the content of LINOCR. When OCF is set, an interrupt is generated if LINIER.OCIE = 1. OCF reflects the same value as LINESR.OCF when the LINFlexD is not in Initialization mode and UARTCR.UART = 1. **NOTE**. For baud rates above 1 Mbit/s, the OCF flag is not usable.
13 - 10	**Parity Error Flag**

Yuntu Microelectronics

Field	Function
PE	Indicates parity error in corresponding receive data byte. 0000b - No parity error detected. 0001b - Parity error detected. Parity is checked after the complete frame is received. PE[3:0] bit usage depends on the values of UARTCR.WLS and UARTCR.WL1. **When UARTCR.WLS = 0 and UARTCR.WL1 = 0:** PE[3] = Parity error flag for fourth byte received. PE[2] = Parity error flag for third byte received. PE[1] = Parity error flag for second byte received. PE[0] = Parity error flag for first byte received. **When UARTCR.WLS = 1 or UARTCR.WL1 = 1:** PE[3] = Not used. PE[2] = Parity error flag for third or fourth byte received. PE[1] = Not used. PE[0] = Parity error flag for first or second byte received.
9 RMB	**Release Message Buffer** 0b - Buffer is free. 1b - Buffer data is ready to be read by software. RMB reflects the same value as LINSR.RMB when the LINFlexD is not in Initialization mode and UARTCR.UART = 1.
8 FEF	**Framing Error Flag** FEF is set by hardware when the LINFlexD detects a framing error (invalid stop bit). An interrupt is generated if LINIER.FEIE = 1. FEF reflects the same value as LINESR.FEF when the LINFlexD is not in Initialization mode and UARTCR.UART = 1.
7 BOF	**FIFO/Buffer Overrun Flag** In UART buffer mode, BOF is set by hardware when there is a new byte received and the RMB bit is not cleared. If LINCR.RBLM = 1, the newly received message is discarded. If LINCR.RBLM = 0, the newly received message overwrites the buffer. In UART FIFO mode, BOF is set when there is a new byte received and the Rx FIFO is full. In UART FIFO mode, once Rx FIFO is full, the new received message is discarded regardless of the value of the RBLM bit. When BOF is set, an interrupt is generated if LINIER.BOIE = 1. BOF reflects the same value as LINESR.BOF when the LINFlexD is not in Initialization mode and UARTCR.UART = 0.
6 RDI	**Receiver Data Input** Reflects the current state of the LINFlexD_RX pin when UARTCR.UART = 1
5 WUF	**Wakeup Flag** WUF is set by hardware when a falling edge is detected on the LINFlexD_RX pin in Sleep mode. An interrupt is generated if LINIER.WUIE = 1. WUF reflects the same value as LINSR.WUF when the LINFlexD is not in Initialization mode and UARTCR.UART = 1.
4	**Receive FIFO Not Empty**

Yuntu Microelectronics

Field	Function
RFNE	RFNE is set by hardware in UART FIFO mode (RFBM = 1), when there is at least one data byte present in the receive FIFO. RFNE is a read-only bit for debugging purposes. RFNE can be used by software in case of a timeout event.
3 TO	**Timeout** TO is set by hardware when a UART timeout occurs —in other words, the value of UARTCTO becomes equal to the preset value of the timeout (UARTPTO register setting). An interrupt is generated if LINIER.TOIE = 1. The GCR.SR bit should be used to reset the receiver FSM to the Idle state in the event of timeout for UART reception in both UART buffer and UART FIFO modes.
2 DRF_RFE	**Data Reception Completed Flag / Rx FIFO Empty Flag** The DRF_RFE bit function depends on whether the LINFlexD is operating in UART buffer mode or UART FIFO mode. **UART Buffer Mode:** In UART buffer mode (RFBM = 0), the DRF (data received flag) function of DRF_RFE is used. DRF is set by hardware in UART buffer mode when the number of bytes programmed in UARTCR.RDFL are received. An interrupt is generated if LINIER.DRIE = 1. DRF is set when the configured number of valid stop bits are received for the last frame. DRF is set regardless of parity error, overrun error, or framing error if the framing error is in the last STOP bit configured. DRF reflects the same value as LINSR.DRF when the LINFlexD is not in Initialization mode and UARTCR.UART = 1. DRF can be read/cleared by software. Writing ‘1’ clears DRF. **UART FIFO Mode:** In UART FIFO mode (RFBM = 1), the RFE (Rx FIFO empty) function of DRF_RFE is used. RFE is set by hardware in UART FIFO mode when the Rx FIFO is empty. RFE is a read-only bit for debugging purposes. It is internally used by the DMA Rx interface.
1	**Data Transmission Completed Flag / Tx FIFO Full Flag**

Yuntu Microelectronics

Field	Function
DTF_TFF	The DTF_TFF bit function depends on whether the LINFlexD is operating in UART buffer mode or UART FIFO mode. **UART Buffer Mode:** In UART buffer mode (RFBM = 0), the DTF (data transmitted flag) function of DTF_TFF is used. DTF is set by hardware in UART buffer mode when data transmission completes. An interrupt is generated if LINIER.DTIE = 1. DTF reflects the same value as LINSR.DTF when the LINFlexD is not in Initialization mode and UARTCR.UART = 1. DTF can be read/cleared by software. Writing ‘1’ clears DTF. **UART FIFO Mode:** In UART FIFO mode (RFBM = 1), the TFF (Tx FIFO full) function of DTF_TFF is used. TFF is set by hardware in UART FIFO mode when Tx FIFO is full. TFF is a read-only bit for debugging purposes. It is internally used by the DMA Tx interface.
0 NF	**Noise Flag** Set by hardware when noise is detected in the received character. NF reflects the same value as LINESR.NF when the LINFlexD is not in Initialization mode and the UARTCR.UART = 1. During reduced oversampling (UARTCR.ROSE = 1), the NF bit is enabled only when UARTCR.OSR = 8.

21.3.1.7 LINFlexD LINTCSR Register

The LINTCSR register contains control and status bits for the LIN timeout feature.

NOTE. When LINTCSR.MODE = 0, any activity on the transmit or receive pins will cause an unwanted change in the value of the 8-bit field Output Compare Value 2 (OC2) of the LIN Output Compare register (LINOCR). If the LINFlexD is enabled in LIN mode and the value of LINTCSR.MODE is changed from ‘1’ to ‘0’, the old values of LINOCR.OC1 and LINOCR.OC2 are retained. As a consequence, if the LINFlexD is reconfigured from UART mode to LIN mode, or LINTCSR.MODE is changed from ‘1’ to ‘0’, an incorrect timeout exception is generated when a LIN communication starts. To avoid this, set LINTCSR.MODE to ‘1’, reconfigure the LINFlexD in LIN mode, then reset LINTCSR.MODE to ‘0’. Before switching LINTCSR.MODE from ‘1’ to ‘0’ between frames, load LINOCR with 0xFFFF.

Offset: 18h

Yuntu Microelectronics

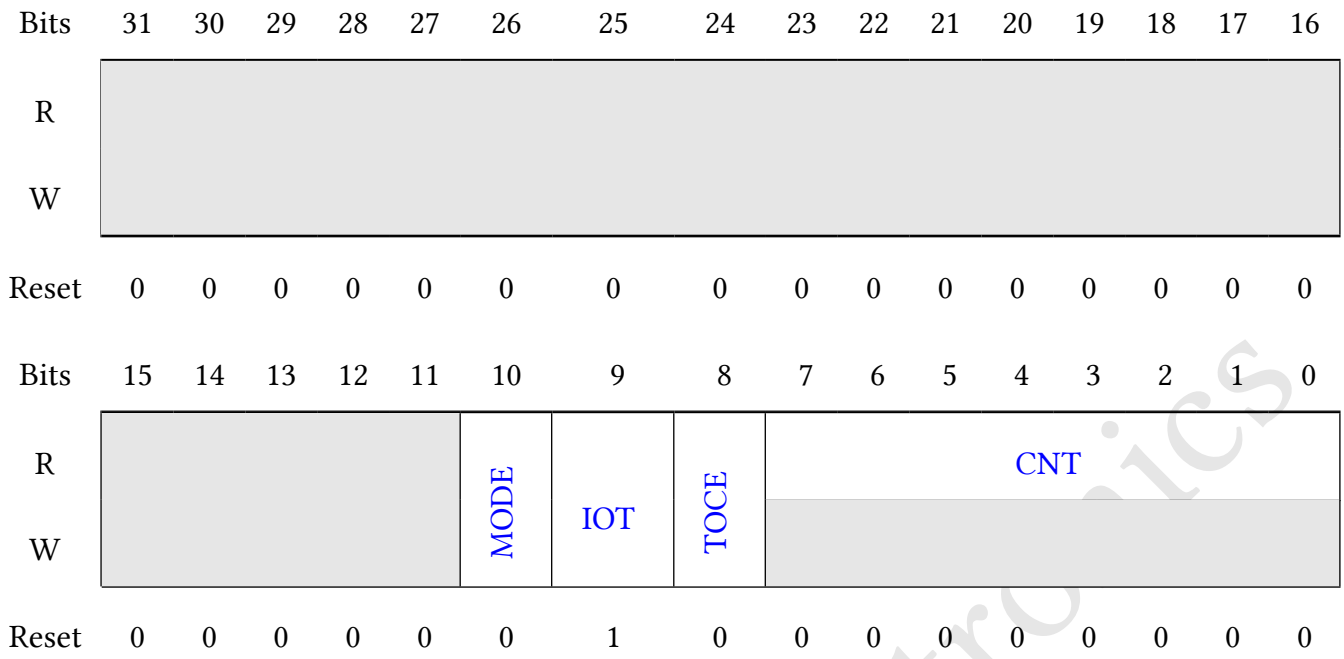


Table 21.11: LINFlexD LINTCSR Register Description

Field	Function
10 MODE	**Timeout Counter Mode** 0b - LIN mode 1b - Output compare mode **NOTE**. Always set LINTCSR.MODE to '1' when UARTCR.UART = 1. **NOTE**. Before switching LINTCSR.MODE from '1' to '0' between frames, load LINOCR with 0xFFFF. MODE can be read in any mode, but can be written only during Initialization mode.
9 IOT	**Idle on Timeout** 0b - LIN state machine does not reset to Idle on timeout. 1b - LIN state machine resets to Idle on timeout event. The idle on timeout feature is applicable only when MODE = 0. IOT can be read in any mode, but can be written only during Initialization mode.
8 TOCE	**Timeout Counter Enable** 0b - Timeout counter disabled. UARTSR.OCF flag is not set on an output compare event. 1b - Timeout counter enabled. UARTSR.OCF is set if an output compare event occurs. TOCE is always configurable by software in Initialization mode. If if the timer is configured in LIN mode and the LINFlexD is not in Initialization mode, then hardware takes control of TOCE.
7 - 0 CNT	**Counter Value** The value of the timeout counter. **NOTE**. For proper functionality of this counter, LINIBRR should be greater than or equal to 5.

Yuntu Microelectronics

21.3.1.8 LINFlexD LINOCR Register

The LINOCR register contains the value to be compared to the LINTCSR.CNT value.
LINOCR is writable by software only in when the counter is in output compare mode.

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OC2								OC1							
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 21.12: LINFlexD LINOCR Register Description

Field	Function
15 - 8 OC2	**Output compare value 2**
7 - 0 OC1	**Output compare value 1**

21.3.1.9 LINFlexD LINTOCR Register

The LINTOCR register contains the LIN mode response and header timeout values.

Offset: 20h

Yuntu Microelectronics

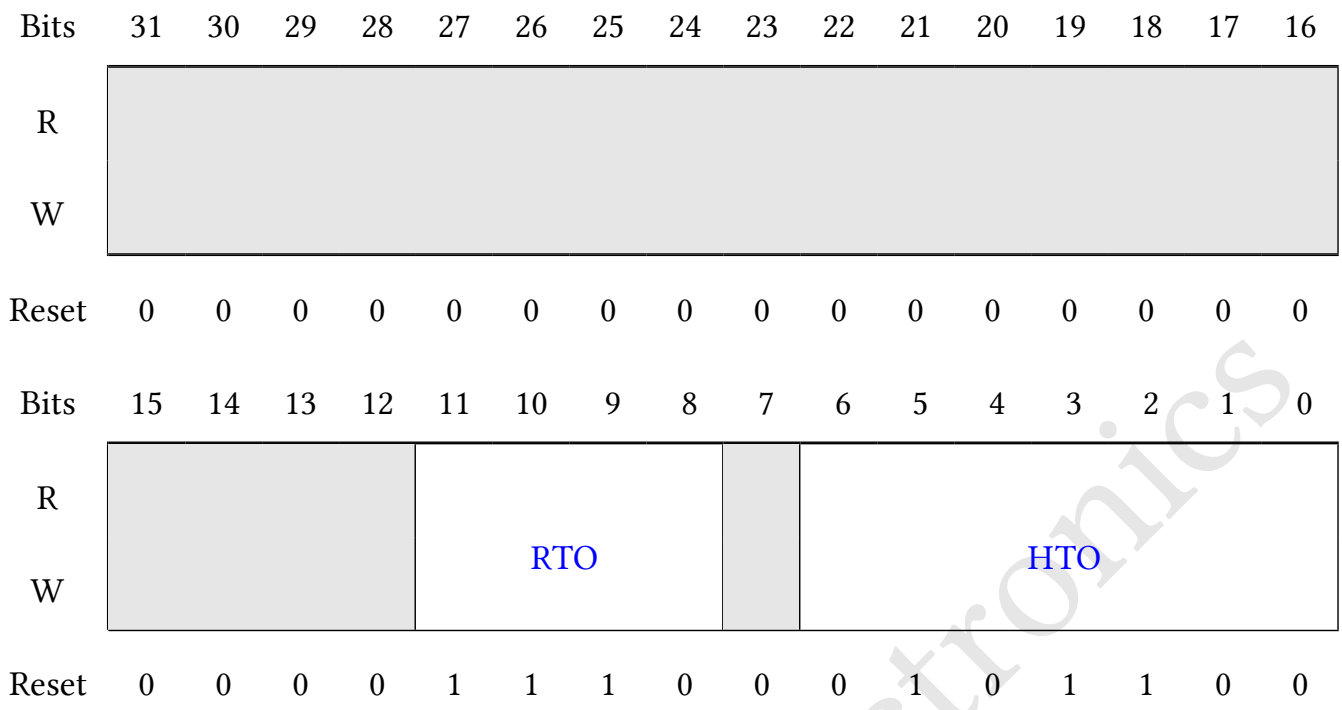


Table 21.13: LINFlexD LINTOCR Register Description

Field	Function
11 - 8 RTO	**Response Timeout Value** The response timeout duration (in bit time) for 1 byte. The reset value is 0Eh = 14, corresponding to $T_{Response_Maximum} = 1.4 \times T_{Response_Nominal}$.
6 - 0 HTO	**Header Timeout Value** HTO specifies the header timeout duration (in bit time). HTO can be written only for slave mode. The header timeout should be programmed without considering 11 bits of break field.

21.3.1.10 LINFlexD LINFBRR Register

LINFBRR sets the fractional part of the LIN baud rate, as described in "Baud Rate Generation".

NOTE. When LINCR1.LASE = 1, LINFBRR should be read only after LINSR.AUTOSYNC_COMP is set to obtain the correct value.

NOTE. LINFBRR cannot be used when reduced oversampling is enabled (UARTCR.ROSE = 1).

Offset: 24h

Yuntu Microelectronics

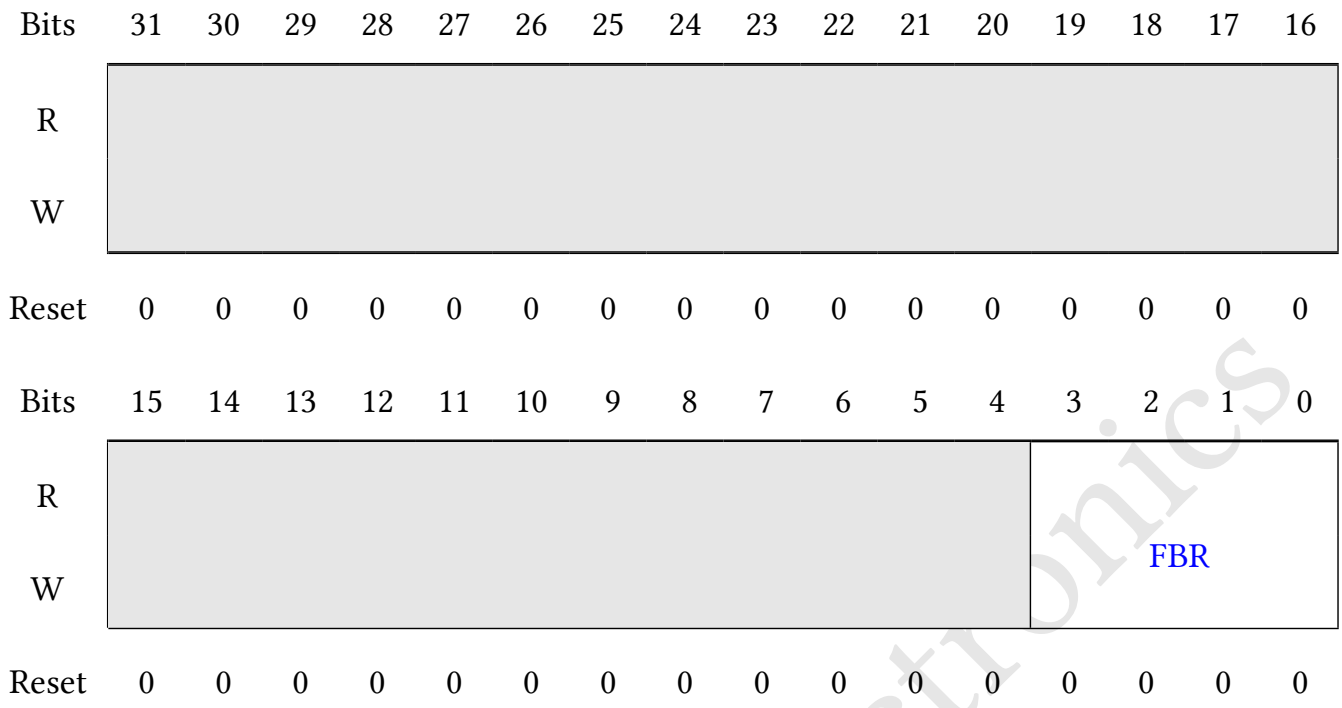


Table 21.14: LINFlexD LINFBRR Register Description

Field	Function
3 - 0 FBR	**Fractional Baud Rate** 0000b - Fraction(LDIV) = 0 0001b - Fraction(LDIV) = 1/16 0010b - Fraction(LDIV) = 2/16 0011b - Fraction(LDIV) = 3/16 0100b - Fraction(LDIV) = 4/16 0101b - Fraction(LDIV) = 5/16 0110b - Fraction(LDIV) = 6/16 0111b - Fraction(LDIV) = 7/16 1000b - Fraction(LDIV) = 8/16 1001b - Fraction(LDIV) = 9/16 1010b - Fraction(LDIV) = 10/16 1011b - Fraction(LDIV) = 11/16 1100b - Fraction(LDIV) = 12/16 1101b - Fraction(LDIV) = 13/16 1110b - Fraction(LDIV) = 14/16 1111b - Fraction(LDIV) = 15/16 FBR can be read in any mode, but can be written only during Initialization mode.

21.3.1.11 LINFlexD LINIBRR Register

LINIBRR sets the integer part of the LIN baud rate, as described in "Baud Rate Generation" .

NOTE. When LINCR1.LASE = 1, LINIBRR should be read only after LINSR.AUTOSYNC_COMP is set to obtain the correct value.

Yuntu Microelectronics

Offset: 28h

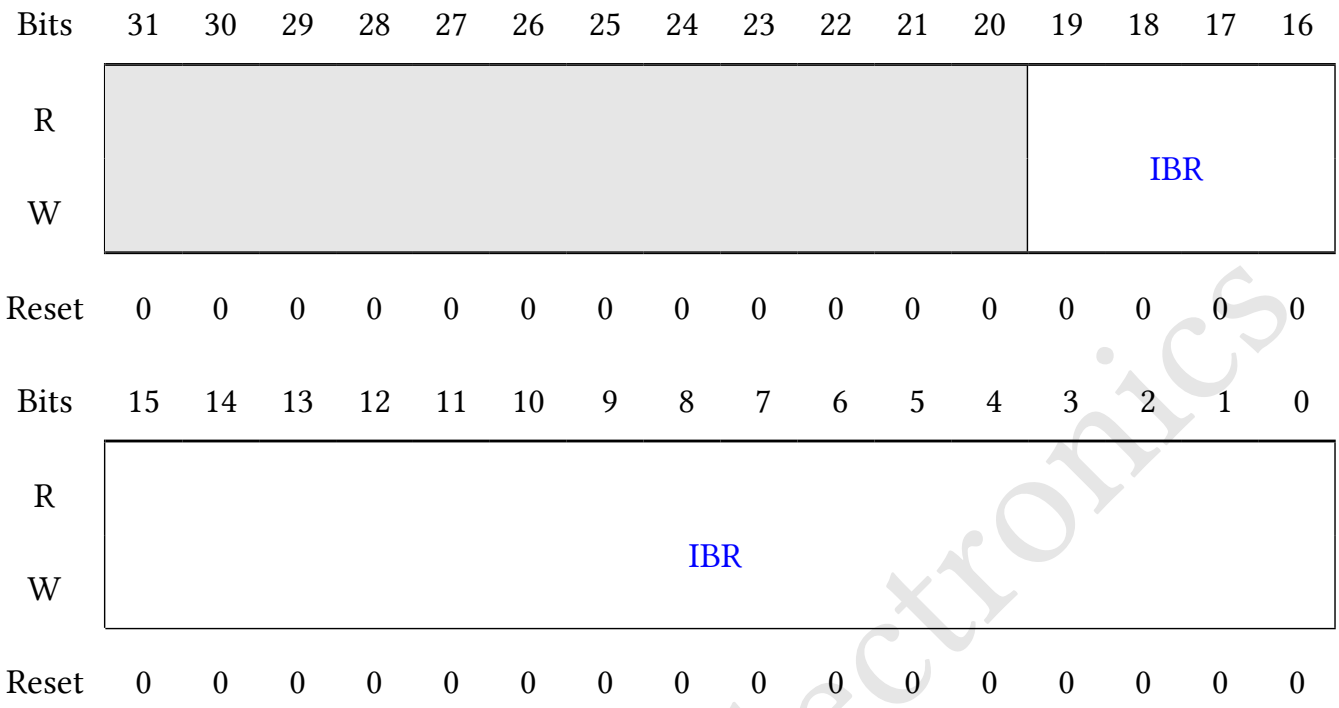


Table 21.15: LINFlexD LINIBRR Register Description

Field	Function
19 - 0 IBR	**Integer Baud Rate** The IBR bits, along with the fractional baud rate bits in LINFBRR, set the LIN baud rate. • IBR = 0x0: LIN clock disabled • IBR = 0x1: Mantissa(LDIV) = 1 • ... • IBR = 0xFFFFE: Mantissa(LDIV) = 1048574 • IBR = 0xFFFFF: Mantissa(LDIV) = 1048575 IBR can be read in any mode, but can be written only during Initialization mode.

21.3.1.12 LINFlexD LINCFR Register

NOTE. There is a delay between 4 to 6 clock cycles of bus clock for the internal checksum’ s value (which is clocked with ipg_baud_clk/16 * LDIV) to reflect on LINCFR.

NOTE. Due to synchronization structures between the two input clocks to the LINFlexD, the write to this register is delayed between 2 to 3 clock cycles of bus clock.

Yuntu Microelectronics

Table 21.16: LINCFR usage

LINCR1.CFD	LINCR1.CCD	LINCFR Read/Write	Checksum Sent
1	1	Read/write	None
1	0	Read-only	None
0	1	Read/write	Programmed checksum
0	0	Read-only	Calculated checksum

Offset: 2ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									CF							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.17: LINFlexD LINCFR Register Description

Field	Function
7 - 0 CF	**Checksum Bits** When LINCR1.CCD = 0, the CF bits are read-only and are calculated by hardware. When LINCR1.CCD = 1, the CF bits can be written by software.

21.3.1.13 LINFlexD LINCR2 Register

The LINCR2 register contains control and status bits related to buffer operations.

NOTE. When accessing LINCR2, each reserved bit should be written with its original reset value.

Offset: 30h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TBDE	IOBE	IOPE	WURQ	DDRQ	DTRQ	ABRQ	HTRQ								
Reset	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.18: LINFlexD LINCR2 Register Description

Field	Function
15 TBDE	**Two-Bit Delimiter Enable** 0b - Delimiter length in break field is 1 bit. 1b - Delimiter length in break field is 2 bits. TBDE can be read in any mode, but can be written only during Initialization mode.
14 IOBE	**Idle on Bit Error** 0b - Bit error does not reset LIN state machine. 1b - Bit error resets LIN state machine. IOBE can be read in any mode, but can be written only during Initialization mode.
13 IOPE	**Idle on Identifier Parity Error** 0b - Parity error does not reset LIN state machine. 1b - Parity error resets LIN state machine. IOPE can be read in any mode, but can be written only during Initialization mode.
12 WURQ	**Wakeup Generate Request** Setting WURQ generates a wakeup request. The character sent during wakeup is copied from DATA0. WURQ is reset by hardware when the wakeup character has been transmitted. WURQ cannot be set in Sleep mode. Software must exit Sleep mode before setting WURQ. Bit error is not checked when transmitting the wakeup request.
11	**Data Discard Request**

Yuntu Microelectronics

Field	Function
DDRQ	DDRQ is set by software to stop data reception if the frame does not concern the node. DDRQ is reset by hardware once the LINFlexD ignores the response and moves to Idle state. For a LIN slave, DDRQ can be set only when LINSR.HRF = 1 and the identifier is software-filtered.
10 DTRQ	**Data Transmission Request** DTRQ is set by software in slave mode to request the transmission of the LIN data field stored in the buffer data register. DTRQ can be set only when LINSR.HRF = 1 (to ensure that data transmission is requested only after a header reception). DTRQ is cleared by hardware when the request has been completed, or on abort request or error condition. In master mode, DTRQ is set by hardware when BIDR.DIR = 1 and header transmission is complete.
9 ABRQ	**Abort Request** ABRQ is set by software to abort the current transmission. The LINFlexD aborts the transmission at the end of the current bit. ABRQ is cleared by hardware when the transmission has been aborted. ABRQ can abort a wakeup request and can also be used in UART mode.
8 HTRQ	**Header Transmission Request** HTRQ is set by software to request the transmission of a LIN header. HTRQ is cleared by hardware when the request has been completed or on abort request. HTRQ has no effect in UART mode. **NOTE**. In master mode, if both HTRQ and ABRQ are set at the same time then ABRQ has no effect. Similarly, in slave mode after header reception, if DTRQ and ABRQ are simultaneously set then ABRQ has no effect.

21.3.1.14 LINFlexD BIDR Register

The BIDR register contains bits that provide information about the identifier of the transaction and other related information.

NOTE. All the fields (ID, CSS, DIR, DFL) of the BIDR register must be updated when an ID filter (enabled) in Slave mode (Tx or Rx) matches the ID received.

Offset: 34h

Yuntu Microelectronics

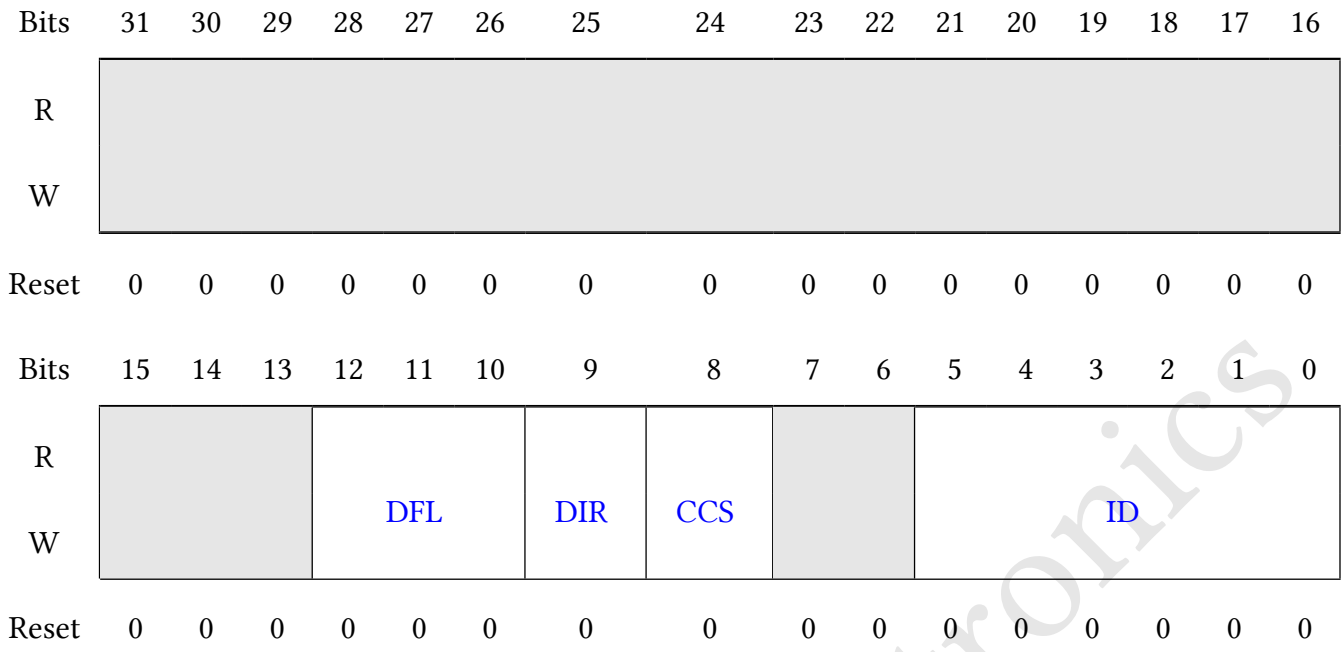


Table 21.19: LINFlexD BIDR Register Description

Field	Function
12 - 10 DFL	**Data Field Length** Number of data bytes in the response part of the frame. DFL = Number of data bytes - 1.
9 DIR	**Direction** Controls the direction of the data field. 0b - LINFlexD receives the data and copies it to the DATA registers. 1b - LINFlexD transmits the data from the DATA registers.
8 CCS	**Classic Checksum** Controls the type of checksum applied on the current message. 0b - Enhanced checksum covering identifier and data fields. This is compatible with LIN specification rev. 2.0 and higher. 1b - Classic checksum covering data field only. This is compatible with LIN specification rev. 1.3 and lower.
5 - 0 ID	**Identifier** Identifier part of the identifier field without the identifier parity. The ID field can be written only in master mode (LINCR1.MME = 1).

Yuntu Microelectronics

21.3.1.15 LINFlexD DATA_DATA8 Register

Offset: 38h + n * 1h

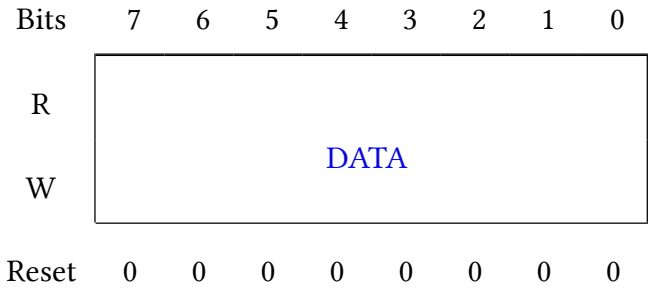


Table 21.20: LINFlexD DATA_DATA8 Register Description

Field	Function
7 - 0 DATA	**Data Byte** Byte of the data field.

21.3.1.16 LINFlexD DATA_DATA16 Register

Offset: 38h + n * 1h

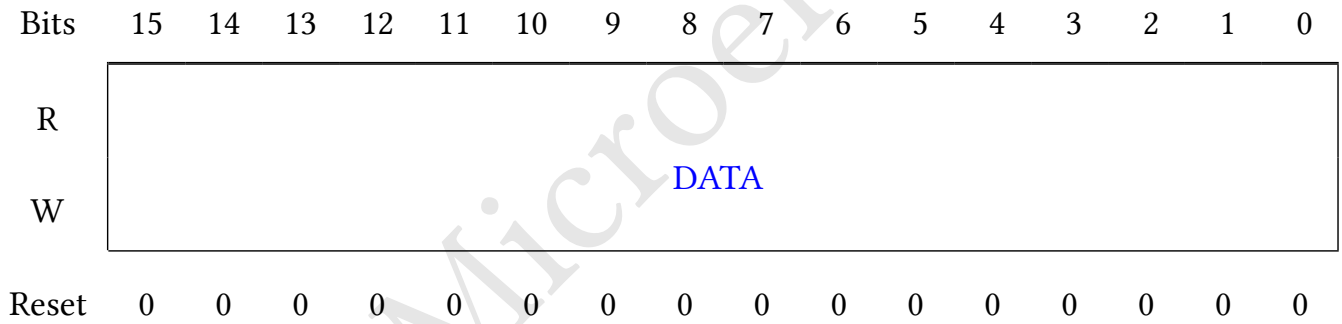


Table 21.21: LINFlexD DATA_DATA16 Register Description

Field	Function
15 - 0 DATA	**Data Byte** Byte of the data field.

21.3.1.17 LINFlexD DATA_DATA32 Register

Offset: 38h + n * 1h

Yuntu Microelectronics

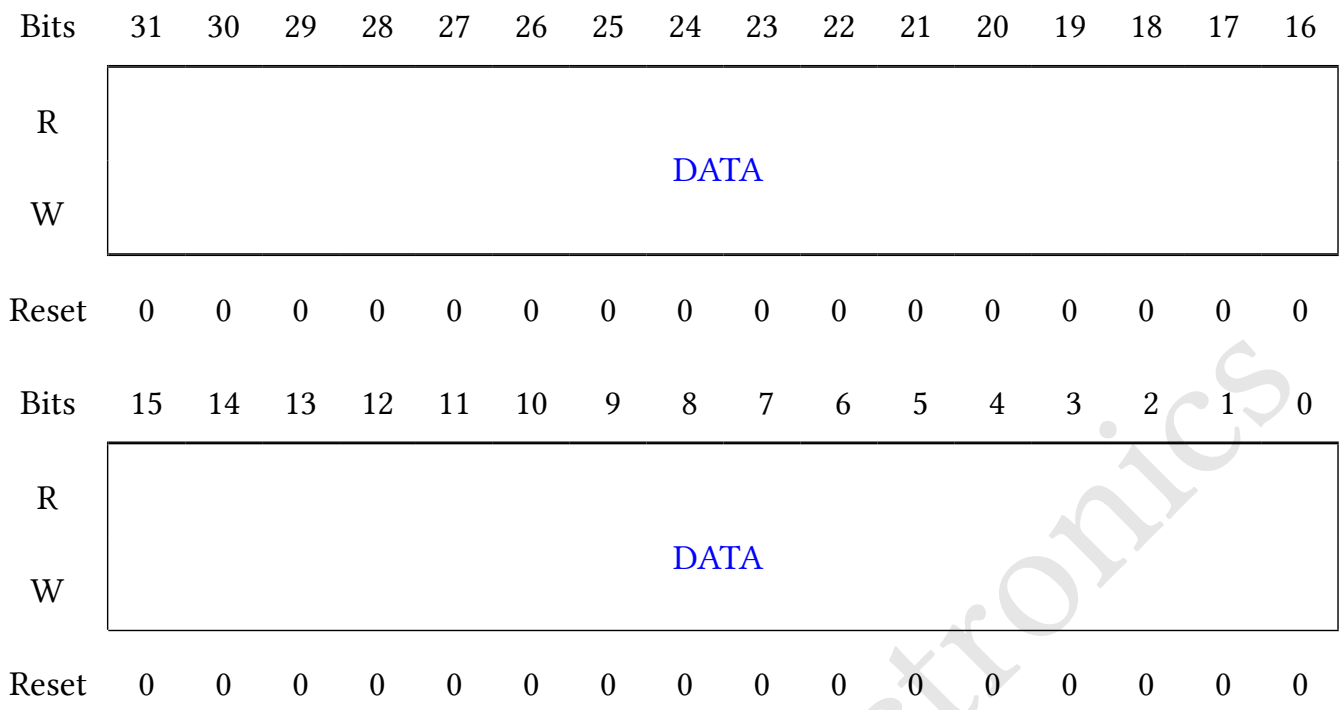


Table 21.22: LINFlexD DATA_DATA32 Register Description

Field	Function
31 - 0 DATA	**Data Byte** Byte of the data field.

Yuntu Microelectronics

21.3.1.18 LINFlexD IFER Register

The IFER register enables/disables the identifier filters. There is one filter active (FACT) bit for each filter.

Offset: 40h

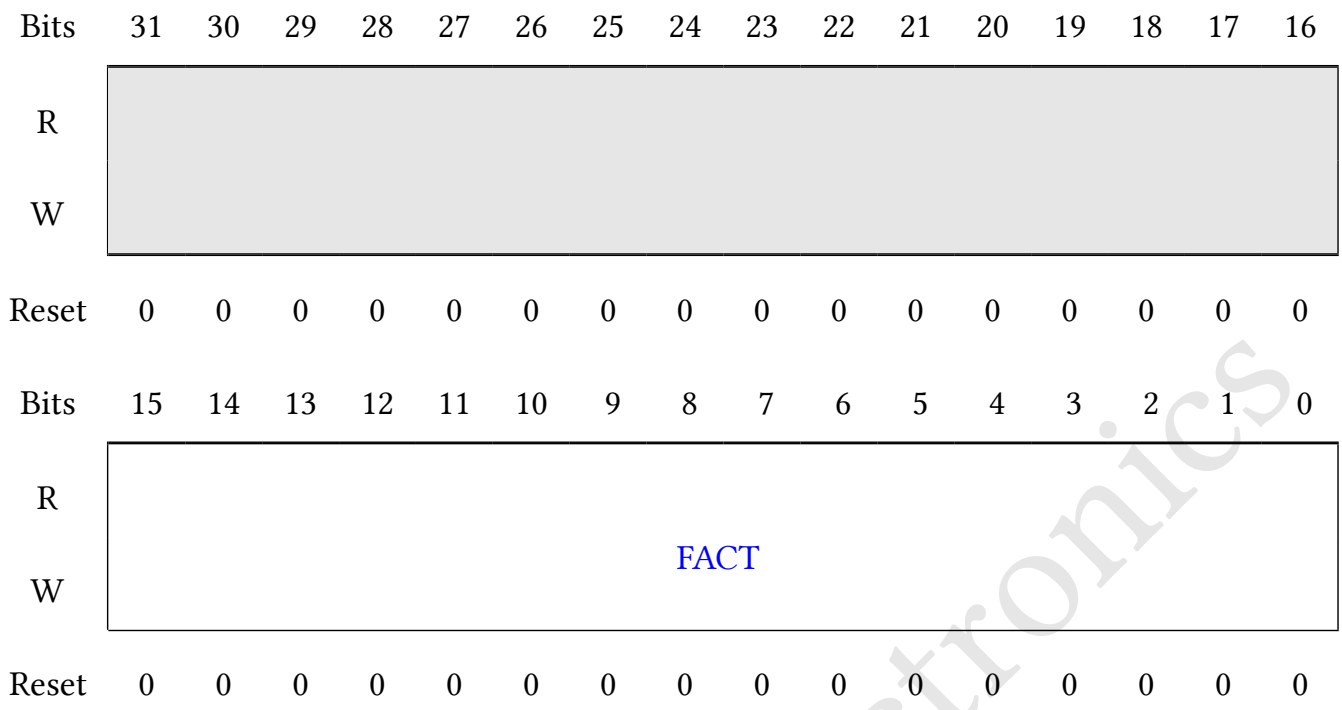


Table 21.23: LINFlexD IFER Register Description

Field	Function
15 - 0 FACT	**Filter Active** One bit for each identifier filter. Software sets FACT[n] to activate the filter n in identifier list mode. In identifier mask mode, FACT[2n+1] have no effect on the corresponding filters as they act as mask for identifier 2n. FACT[n] can be read in any mode, but can be written only during Initialization mode.

Yuntu Microelectronics

21.3.1.19 LINFlexD IFMI Register

The IFMI register contains the index corresponding to the received identifier. It can be used to read or write the data directly in RAM.

Offset: 44h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R												IFMI				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 21.24: LINFlexD IFMI Register Description

Field	Function
4 - 0 IFMI	**Filter Match Index** Contains the index corresponding to the received identifier. Upon a filter match with filter x, IFMI[N:0] = x+1. On no match, IFMI is equal to 0x00. IFMI can be used to directly write or read the data in RAM (see "Slave Mode" for details).

Yuntu Microelectronics

21.3.1.20 LINFlexD IFMR Register

The IFMR register configures the modes of filters. There is one IFM bit for each pair of filters.

Offset: 48h

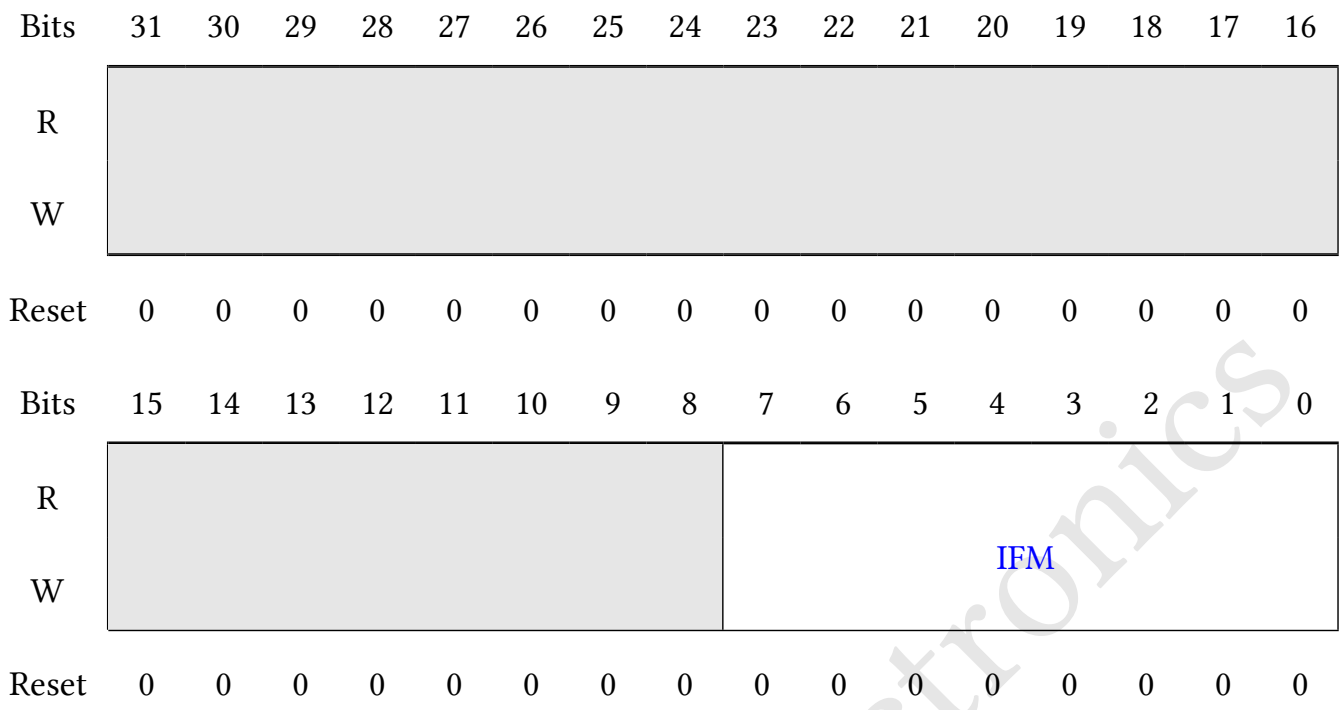


Table 21.25: LINFlexD IFMR Register Description

Field	Function
7 - 0 IFM	**Filter Mode** • 0: Filters 2n and 2n+1 are in identifier list mode. • 1: Filters 2n and 2n+1 are in mask mode. Filter 2n+1 is the mask for filter 2n. IFM[n] can be read in any mode, but can be written only during Initialization mode.

21.3.1.21 LINFlexD IFCR Register

There is an Identifier Filter Control Register (IFCRn) for each filter.
Even-numbered instances of IFCRn (IFCR0, IFCR2, IFCR4, and so on) act as filters in both identifier list and identifier mask modes.
Odd-numbered instances of IFCRn (IFCR1, IFCR3, IFCR5, and so on) act as:

- Filters in identifier list mode.
- A mask for the preceding-numbered register in identifier mask mode. For example, if filters 0 and 1 are configured for identifier mask mode, IFCR1 acts as a mask for IFCR0.

The IFCRn registers are read-only in Normal mode and can be written only in Initialization mode.

Offset: 4ch + n * 4h

Yuntu Microelectronics

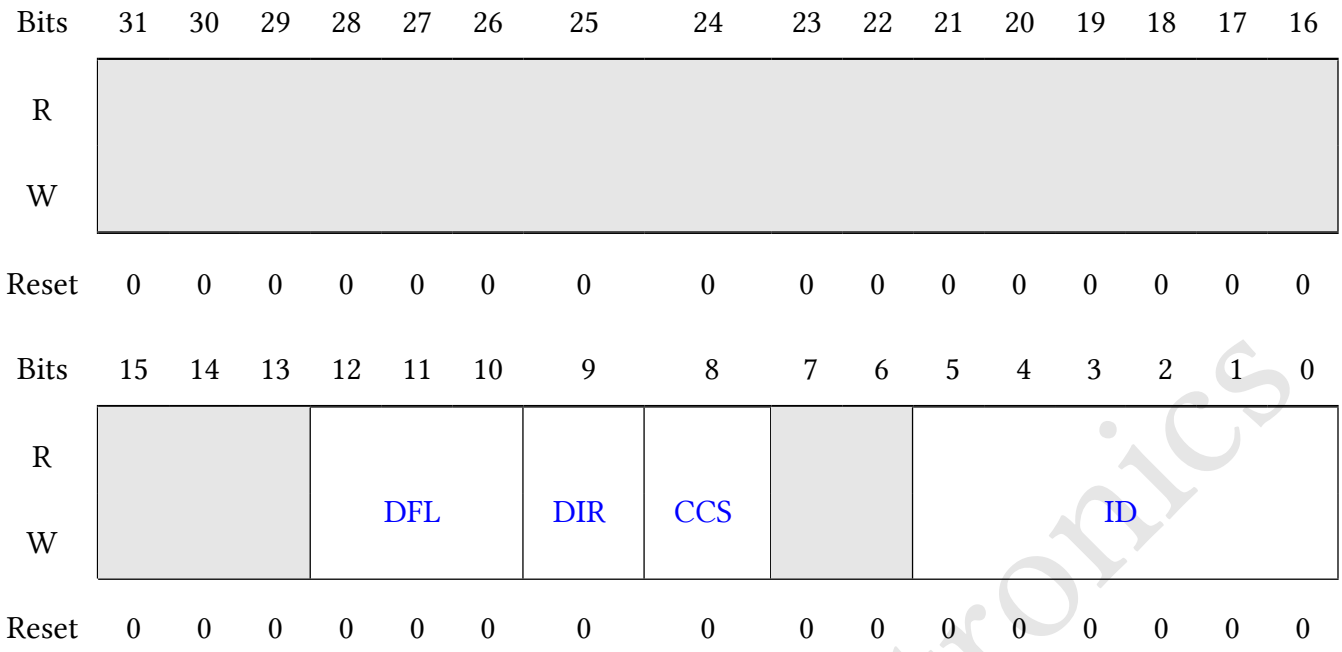


Table 21.26: LINFlexD IFCR Register Description

Field	Function
12 - 10 DFL	**Data Field Length** Number of data bytes in the response part of the frame. DFL = Number of data bytes - 1.
9 DIR	**Direction** Controls the direction of the data field. 0b - LINFlexD receives the data and copy it to the DATA registers. 1b - LINFlexD transmits the data from the DATA register.
8 CCS	**Classic Checksum** Controls the type of checksum applied on the current message. 0b - Enhanced checksum covering identifier and data fields. This is compatible with LIN specification rev. 2.0 and higher. 1b - Classic checksum covering data field only. This is compatible with LIN specification rev. 1.3 and lower.
5 - 0 ID	**Identifier** Identifier part of the identifier field without the identifier parity.

21.3.1.22 LINFlexD GCR Register

The GCR register contains control bits that apply to both LIN and UART modes.
The GCR register is read-only in Normal mode and can be written only in Initialization mode.

Offset: 8ch

Yuntu Microelectronics

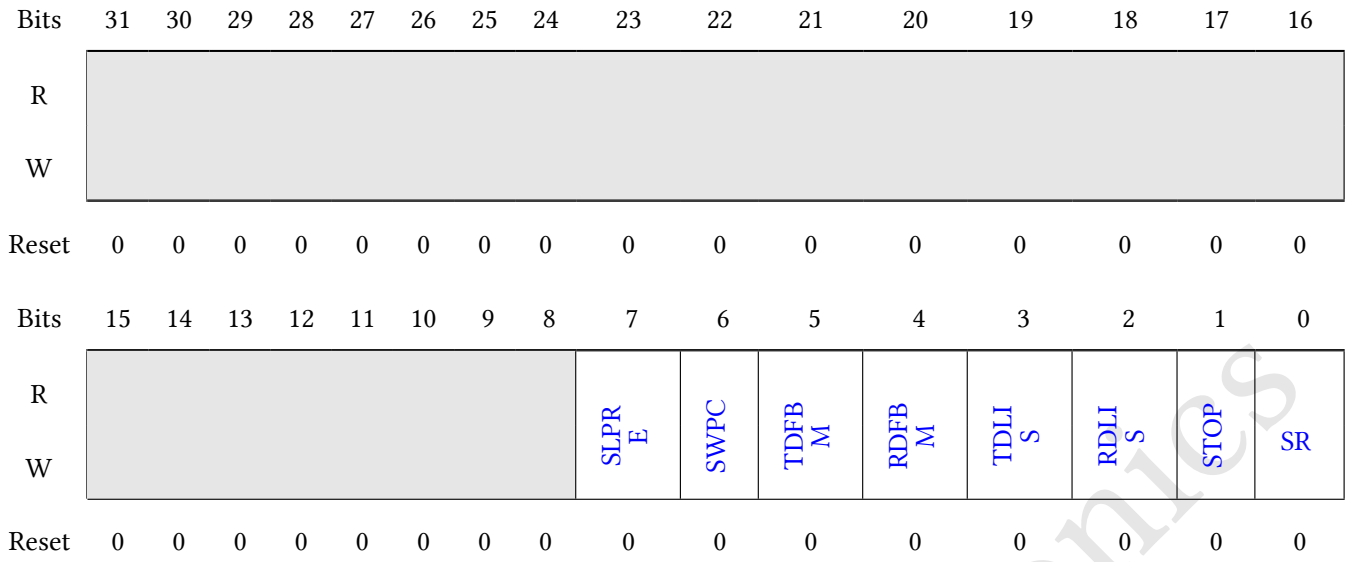


Table 21.27: LINFlexD GCR Register Description

Field	Function
7 SLPRE	**Sleep Receive Enable** When this bit is set, the receiving data will be stored in LINFlexD buffer while the system is in sleep/deepsleep mode. It is valid only if LINFlexD is running (Normal mode) when system is in sleep/deepsleep mode.
6 SWPC	**Swap Control** Controls the direction switch between ipp_ind_linrx and ipp_do_lintx. 0b - ipp_ind_linrx is working for reception, ipp_do_lintx is working for transmission. 1b - ipp_ind_linrx is working for transmission, ipp_do_lintx is working for reception. SWPC can be read in any mode, but can be written only during Initialization mode.
5 TDFBM	**Transmit Data First Bit MSB** Controls whether the first bit of transmit data (payload only) is the most or least significant bit from the respective buffer data register. 0b - The first bit of transmitted data is the least significant bit of the buffer data register (DATA0[0], DATA1[0], DATA2[0], DATA3[0], DATA4[0], DATA5[0], DATA6[0], DATA7[0]). 1b - The first bit of transmitted data is the most significant bit of the buffer data register (DATA0[7], DATA1[7], DATA2[7], DATA3[7], DATA4[7], DATA5[7], DATA6[7], DATA7[7]).
4	**Receive Data First Bit MSB**

Yuntu Microelectronics

Field	Function
RDFBM	Controls whether the first bit of received data (payload only) is mapped to the most or least significant bit from the respective buffer data register. 0b - The first bit of received data is mapped to least significant bit of the buffer data register (DATA0[0], DATA1[0], DATA2[0], DATA3[0], DATA4[0], DATA5[0], DATA6[0], DATA7[0]). 1b - The first bit of received data is mapped to the most significant bit of the buffer data register (DATA0[7], DATA1[7], DATA2[7], DATA3[7], DATA4[7], DATA5[7], DATA6[7], DATA7[7]).
3 TDLIS	**Transmit Data Level Inversion Selection** Controls the data inversion of transmitted data (payload only). 0b - Transmitted data is not inverted. 1b - Transmitted data is inverted.
2 RDLIS	**Received Data Level Inversion Selection** Controls the data inversion of received data (payload only). 0b - Received data is not inverted. 1b - Received data is inverted.
1 STOP	**One or Two Stop Bit Configuration** Controls the number of stop bits transmitted for all fields (delimiter, sync, ID, checksum, payload). 0b - One stop bit. 1b - Two stop bits.
0 SR	**Soft Reset** Writing ‘1’ to SR executes a soft reset of the LINFlexD (FSMs, FIFO pointers, counters, timers, status and error registers) without modifying the configuration registers. SR should be cleared by software to perform further operations (SR is not cleared by hardware). The SR bit can be written only be software in Initialization mode. The read value of SR is always ‘0’. When writing ‘1’ to SR, below register fields are reset:

Register	Comment
LINSR	All fields except RXBUSY and AUTOSYNC_COMP are reset.
LINESR	All fields are reset.
LINTCSR	Only CNT[7:0] is reset.
UARTSR	All fields except RFE and TFF are reset.
UARTCR	Only TFC and RFC are reset.
UARTCTO	All fields are reset.

21.3.1.23 LINFlexD UARTPTO Register

The UARTPTO register contains the preset value of the timeout register in UART mode and is programmed according to the number of bits to be received or to monitor the idle state of the reception line. The UARTPTO register can be written by software at any time.

Yuntu Microelectronics

Offset: 90h

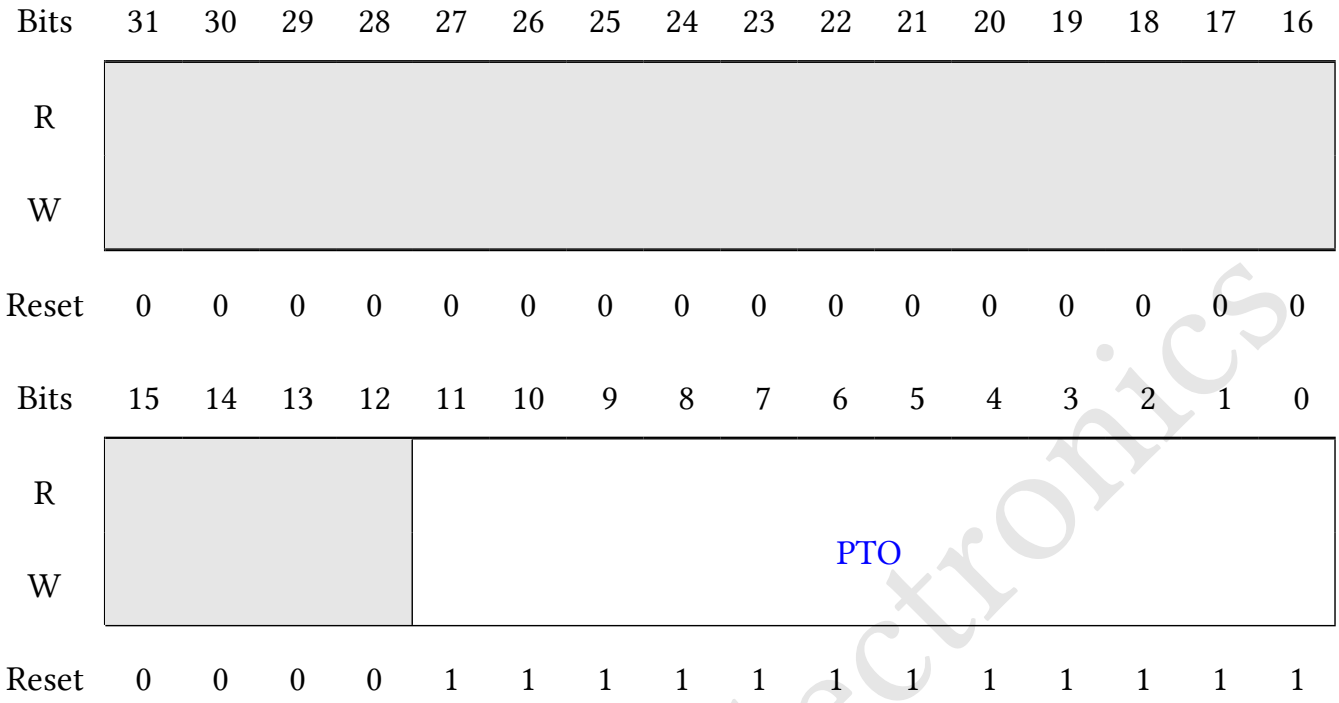


Table 21.28: LINFlexD UARTPTO Register Description

Field	Function
11 - 0 PTO	**Preset Timeout** PTO defines the preset value of timeout counter. A zero value is forbidden; otherwise, the UARTSR[TO] status bit is immediately set.

21.3.1.24 LINFlexD UARTCTO Register

The UARTCTO register contains the current timeout value in UART mode. UARTCTO is used with the UARTPTO register to monitor the number of bits received by UART or to monitor the idle state of the reception line.
UART timeout works in both CPU and DMA modes.
The timeout counter:

- Starts at zero and counts upward.
- Is clocked with ipg_baud_clk / (16 * LDIV) synchronized to bus clock (when UARTCR.ROSE = 0).
- Is clocked with ipg_baud_clk / (UARTCR.OSR * LINIBRR.IBR) synchronized to bus clock (when UARTCR.ROSE = 1).
- Is automatically enabled when UARTCR.RXEN = 1.

NOTE. Due to synchronization, there is a delay between 4 to 6 clock cycles of bus clock for the internal value (which is clocked with ipg_baud_clk/16 * LDIV) to reflect on UARTCTO.
The timeout counter is reset when one of the following conditions occurs:

Yuntu Microelectronics

- The number of frames received is equal to UARTCR.NEF.
- UARTCTO becomes equal to UARTPTO.
- UARTPTO is written.
- UARTSR.DRF is set and UARTCR.DTU = 1.

Offset: 94h

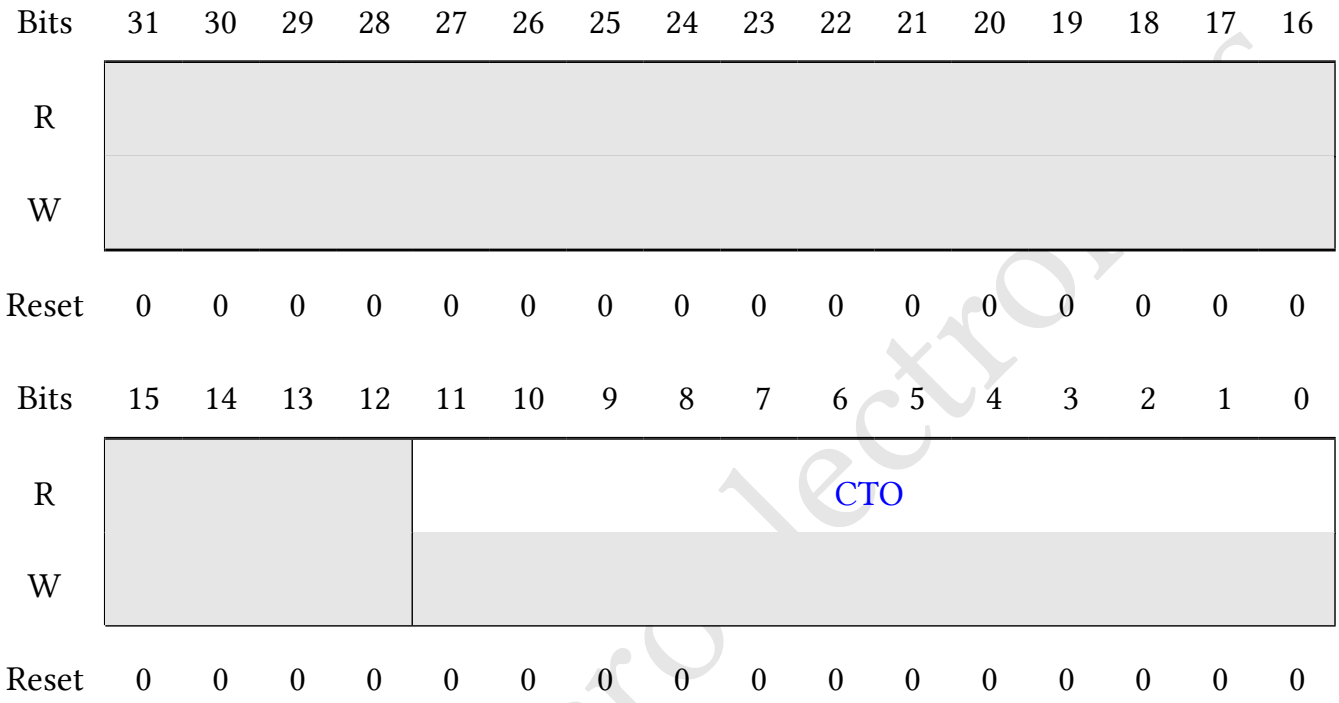


Table 21.29: LINFlexD UARTCTO Register Description

Field	Function
11 - 0 CTO	**Preset Timeout** CTO defines the current value of the timeout counter. CTO is a read-only field. CTO is reset every time UARTPTO is written, or UARTCTO = UARTPTO, or by hard/soft reset. When the CTO value matches the preset value (UARTPTO.PTO), the status bit UARTSR.TO is set.

21.3.1.25 LINFlexD DMATXE Register

The DMATXE register enables the DMA Tx interface. The DMATXE register can be written by software at any time. When DMATXE = 0x0000_0000, the DMA Tx interface FSM is forced (by soft reset) into the Idle state.

NOTE. When configuration parameter DMA_SUPPORTED = 0, the LINFlexD does not support DMA operation. The DMATXE register is invalid and must be written only with value 0x0000_0000.

Offset: 98h

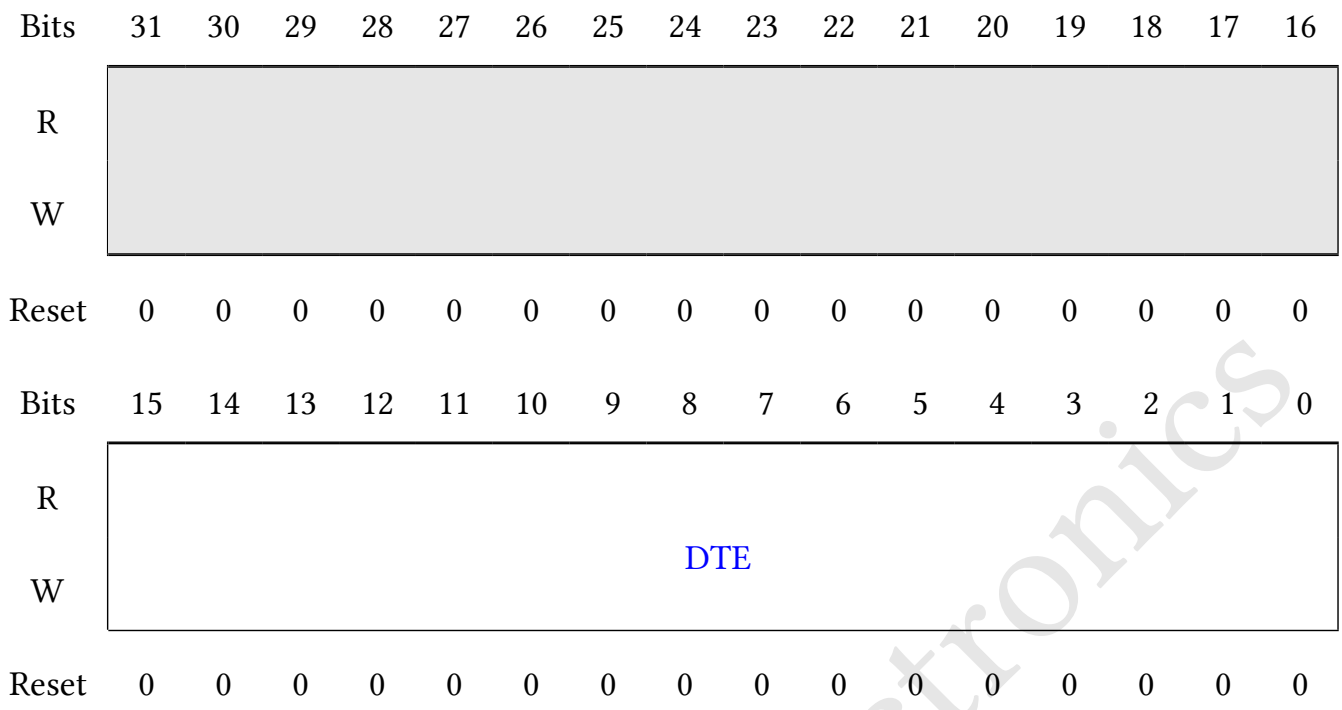


Table 21.30: LINFlexD DMATXE Register Description

Field	Function
15 - 0 DTE	**DMA Tx Channel n Enable** • 0 = DMA Tx channel n is disabled. • 1 = DMA Tx channel n is enabled.

21.3.1.26 LINFlexD DMARXE Register

The DMARXE register enables the DMA Rx interface. The DMARXE register can be written by software at any time. When DMARXE = 0x0000_0000, the DMA Rx interface FSM is forced (by soft reset) into the Idle state.

NOTE. When configuration parameter DMA_SUPPORTED = 0, the LINFlexD does not support DMA operation. The DMARXE register is invalid and must be written only with value 0x0000_0000.

Offset: 9ch

Yuntu Microelectronics

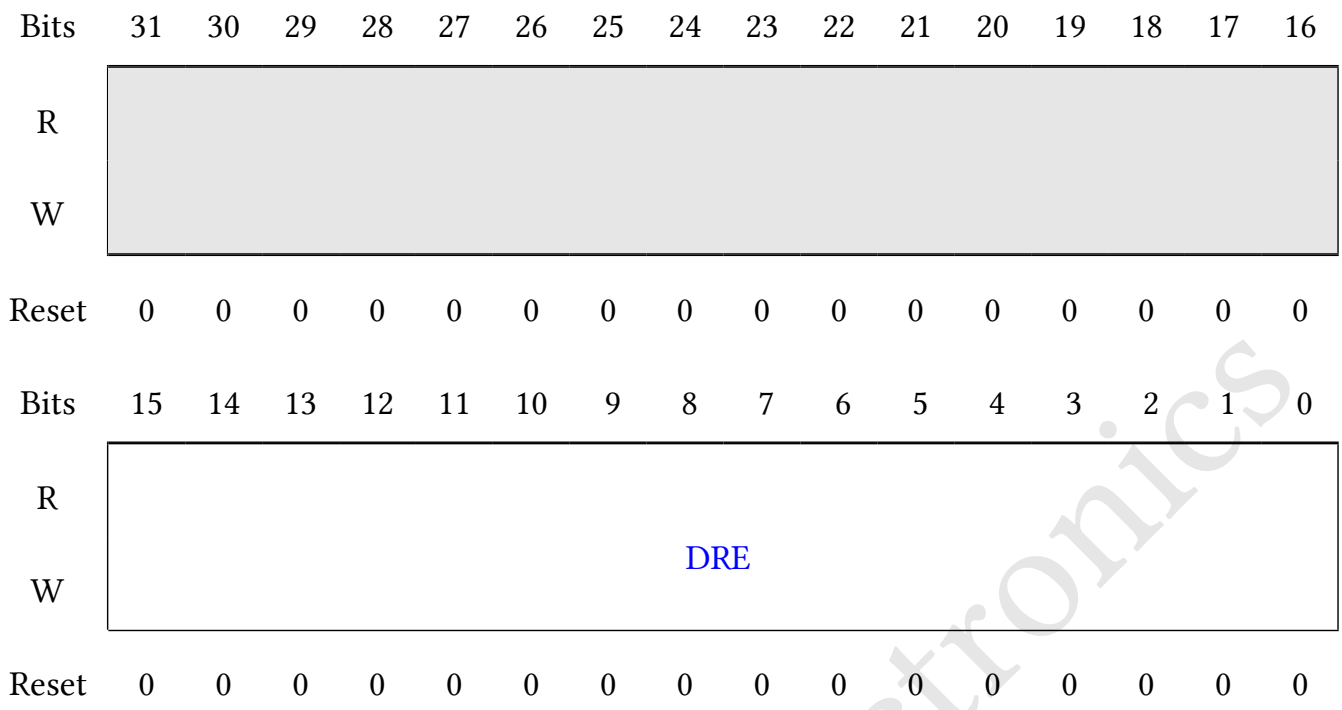


Table 21.31: LINFlexD DMARXE Register Description

Field	Function
15 - 0 DRE	**DMA Rx Channel n Enable** • 0 = DMA Rx channel n is disabled. • 1 = DMA Rx channel n is enabled.

21.4 Functional Description

21.4.1 LIN Protocol

LIN is a serial communication protocol. A LIN cluster consists of one master task and several slave tasks. A master node contains the master task as well as a slave task. All other nodes contain a slave task only. The master node decides when and which frame is transferred on the bus. The slave task provides the data to be transported by the frame.

21.4.1.1 Frames

A frame consists of a header provided by the master task and a response provided by the slave task. The header consists of a break, followed by a sync pattern, followed by an identifier. The slave task associated with the identifier provides the response. The response consists of a data field and checksum. The slave task designated to receive the data associated with the identifier receives the response and verifies the checksum.

LIN frames are transmitted/received as shown in following figures.

Yuntu Microelectronics

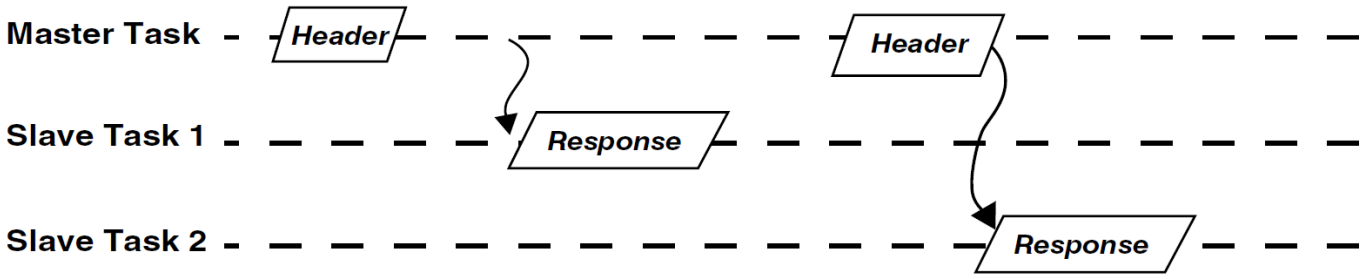


Figure 21.2: Frames

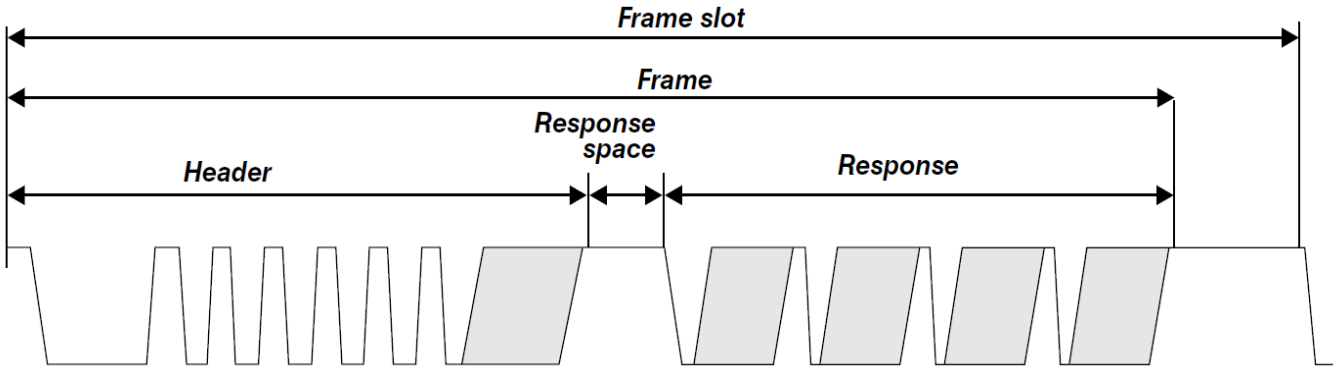


Figure 21.3: Structure of LIN Frame

21.4.1.2 Data Field

Each byte is transmitted as shown in below. The LSB of the data is sent first and the MSB is sent last. The start bit is encoded as a bit with value zero (dominant) and the stop bit is encoded as bit value one (recessive).

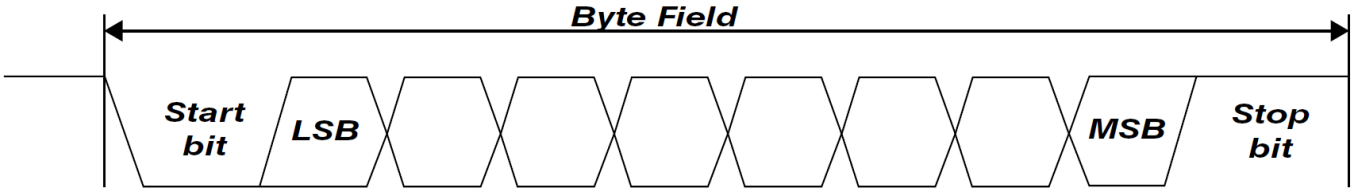


Figure 21.4: Structure of Byte Field

21.4.1.3 Break

The break symbol signals the beginning of a new frame. It is the only field that does not comply with the byte field structure. The break is always generated by the master and shall be at least 13 bits of dominant value, followed by a break delimiter as shown in the break field. The break delimiter must be at least one nominal bit time in length.

Figure 21.5: Break Field

21.4.1.4 Sync Byte

The sync byte is a byte field with the data value of 0x55.

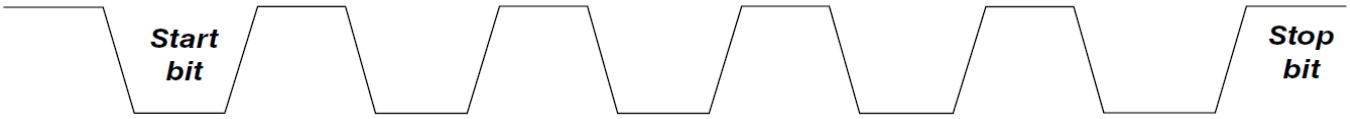


Figure 21.6: Sync Byte Field

21.4.1.5 Identifier

The identifier field consists of two sub-fields, the identifier and the identifier parity. Bits 0 to 5 are the identifier. Bits 6 and 7 indicate the parity:

- P0 = ID0 XOR ID1 XOR ID2 XOR ID4
- P1 = NOT (ID1 XOR ID3 XOR ID4 XOR ID5)

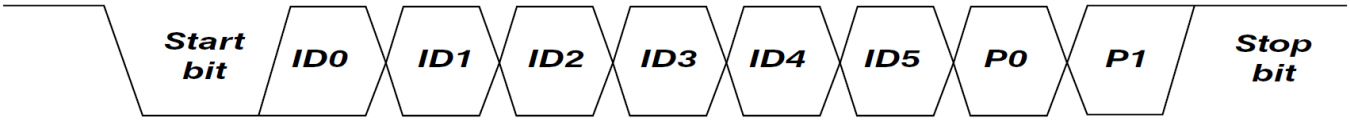


Figure 21.7: Identifier Field

21.4.1.6 Checksum

The last field of a frame is the checksum. The checksum contains the inverted 8-bit sum with carryover of all data bytes and the identifier. Checksum calculation over the data bytes only is called classic checksum and is used for communication with LIN 1.3 (and lower) slaves. Checksum calculation over the data bytes and the protected identifier byte is called enhanced checksum and is used for communication with LIN 2.0 (and higher) slaves.

21.4.2 LINFlexD Features

21.4.2.1 Operating Modes

The LINFlexD has three operating modes: Initialization, Normal, and Sleep. After hardware reset, the LINFlexD enters Sleep mode to reduce power consumption. The figure below shows the state diagram of operating modes.

Yuntu Microelectronics

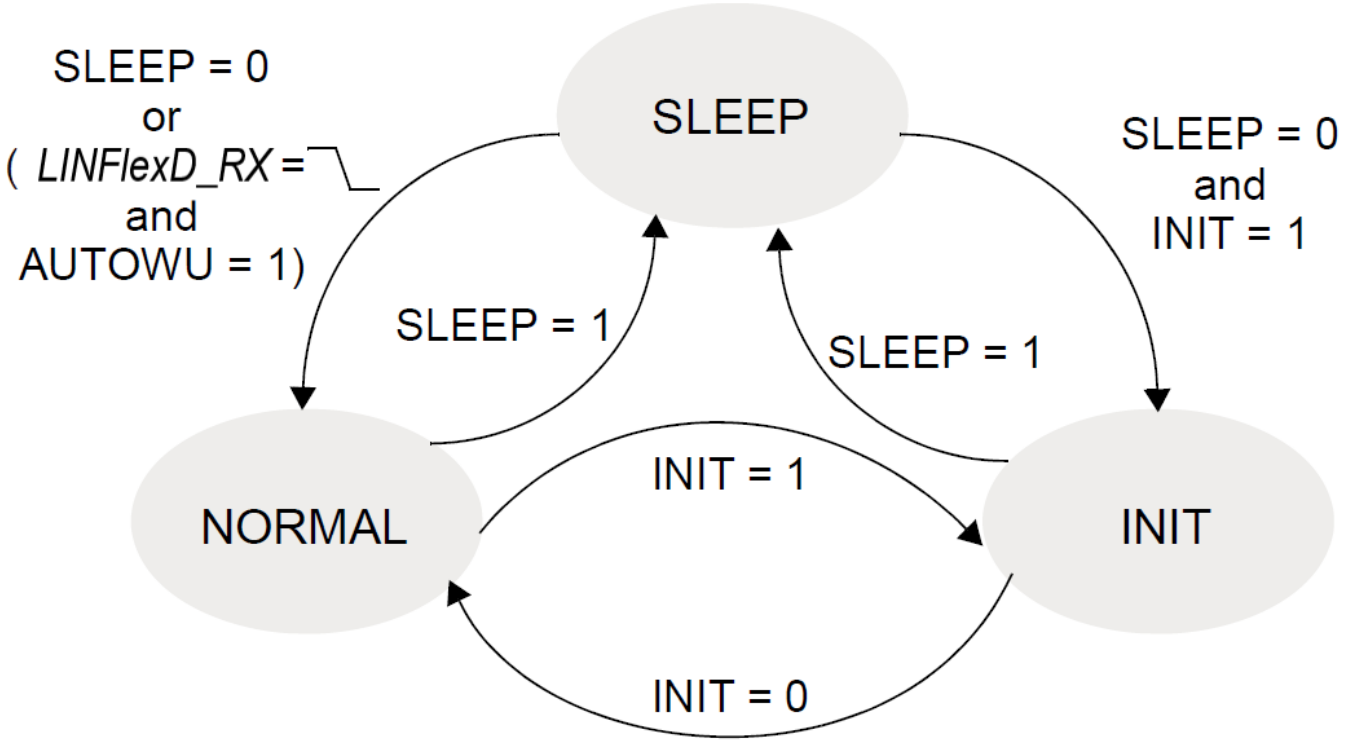


Figure 21.8: Operating Modes

Yuntu Microelec

Initialization Mode (INIT)

To enter Initialization mode, software sets the INIT bit in LINCR1. To exit Initialization mode, software should reset the INIT bit.

When the LINFlexD is in Initialization mode, all message transfers to and from the LIN bus are stopped and the status of LIN bus output ipp_do_lintx is recessive (high). If software invokes Initialization mode when a bus transfer is in progress, the transfer is aborted. Software should therefore check the LIN state before setting the INIT bit.

To initialize the LINFlexD Controller, software must:

1. Set up the baud rate registers.
2. Reset the UART bit.
3. Select master or slave mode.
4. Configure checksum control bit.
5. Initialize the identifier list (slave mode).

Normal Mode (NM)

After initializing the LINFlexD, software can set the LINFlexD to Normal mode by clearing the INIT bit.

Sleep Mode (SM)

Sleep mode in the LINFlexD reduces power consumption. Sleep mode is entered by setting the SLEEP bit in LINCR1. The LINFlexD can be awakened from Sleep mode by clearing the SLEEP bit.

If software detects a wakeup pulse of 150 µs on the LIN Bus, it can request the LINFlexD to wake up from Sleep mode. See “Wakeup Management” .

21.4.2.2 Loop Back Mode

Loop Back mode is entered by setting LINCR1.LBKM. In Loop Back mode, the LINFlexD receives the identifier and data transmitted by itself and writes the same in the BIDR register and data buffers respectively. Loop Back mode is provided for self-test functions. Bit error is checked in Loop Back mode.

While in Loop Back mode, the LINFlexD ignores the LINFlexD_RX signal. There is an internal feedback from the Tx output to the Rx input. The ipp_do_lintx signal can be disconnected from the chip-level LINTX pin by a user-implemented GPIO function.

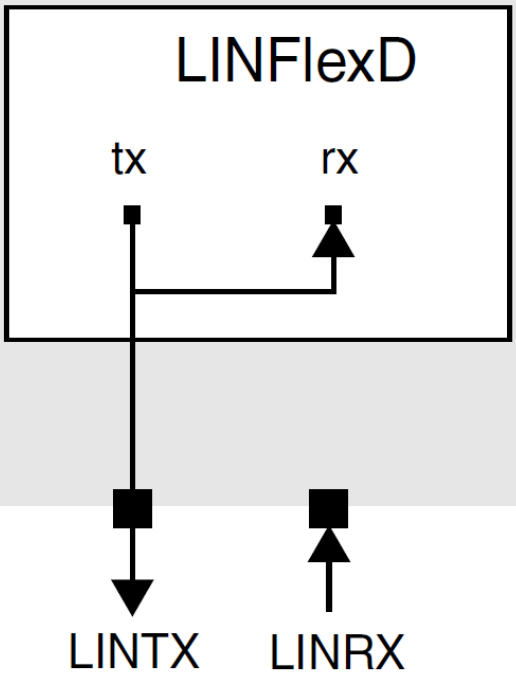


Figure 21.9: LINFlexD in Loop Back Mode

21.4.2.3 Master Mode

Master mode is selected by setting the LINCR1.MME bit.

Header Transmission

According to the LIN protocol, any communication on the LIN bus is triggered by the master sending a header. The header is transmitted by the master task of a node, while the response is transmitted by the slave task of a node.

To transmit the header, first program the identifier (ID), data field length (DFL), message direction (DIR), and classic checksum enable (CCS) in the BIDR register; then set the HTRQ bit in LINCR2. After header transmission starts, do not modify the BIDR register bits until the current frame is complete. The transmitted ID is also received by the master node and copied to BIDR.

Data Transmission

When the master node is the publisher of the data corresponding to the identifier sent by the master, then the slave task of the node should send the data in the response part of the frame. Therefore, software must provide the data to the LINFlexD before the header transmission is requested. The data to be transmitted is stored in the message buffers DATA. The number of bytes to be transmitted depends on

BIDR.DFL. Software uses the BIDR.CCS bit to configure the checksum type (classic or enhanced) for each message.

When the response is sent successfully, LINSR.DTF is set. In the case of an error, LINSR.DTF is not set and the corresponding error flag is set in LINESR. (See "Errors" for more details.)

The direction of the message buffer is controlled by the BIDR.DIR bit. The transmitted data is also received by the same node and copied to the buffer.

Data Reception

To receive data from a slave node, the master sends a header with the corresponding identifier. The data received from the slave is stored in the message buffer and the status of the message is stored in LINSR.

If the response is received successfully, LINSR.DRF bit is set. In the case of an error, the LINSR.DRF flag is not set and the corresponding error flag is set in LINESR. (See "Errors" for more details.)

Data Discard

To discard the data after header transmission, set the LINCR2.DDRQ bit.

21.4.2.4 Slave Mode

Slave mode is selected when the LINCR1.MME bit is cleared.

Data Transmission

On header reception, the LINSR.HRF bit is set and an Rx interrupt is generated. Software must then:

1. Read the received ID in the BIDR register.
2. Fill the DATA registers.
3. Program the CCS and DIR bits in the BIDR register.
4. Specify the data field length in the BIDR.DFL bits.

The LINSR.HRF bit should be reset only after the LINCR2.DTRQ bit is set. For the DTRQ to be effective, the HRF bit must be set. This is to ensure that DTRQ is not set randomly, but only after a header reception. The DIR and DTRQ bits cannot be set after LINSR.RXBUSY is asserted.

Alternately, one or more identifier filters are configured for transmission by setting the DIR bit, and activated by setting the enable bits in IFER. When at least one identifier filter is active and configured for transmission, and the received ID matches the filter, a Tx interrupt is generated. Software can use the index in the IFMI register to point directly to the corresponding data array in the RAM and copy the data to DATA.

The use of a filter saves software the processing time required to read the ID value in the BIDR, match it, and configure the data field length and checksum type.

If the number of filters provided by the LINFlexD is not sufficient for the application, mask mode can be used for the filters.

The transmitted data is also received by the same node and copied to the buffer.

Data Reception

When the LINFlexD is the subscriber of the data of the received identifier, on header reception, the HRF flag is set and an Rx interrupt is generated. Software must read the received ID from the BIDR register and specify the data field length before the reception of the stop bit of the first data byte. When the checksum is received, an Rx interrupt is generated for software to read the received data from DATA and LINSR.RMB is set. Software must then release the data buffer by clearing LINSR.RMB.

Yuntu Microelectronics

When at least one identifier filter is active and configured for reception, an Rx interrupt is generated only after the checksum reception. No interrupt request is generated on reception of the ID.

If the identifier is filtered by software, software can discard the data by setting LINCR2.DDRQ while HRF is set.

NOTE. For software filtering, software must set the type of checksum (configure BIDR.CCS) before reception of data. Otherwise, the previous value of the CCS bit is considered while calculating checksum.

21.4.2.5 Errors

Header Error

A header error is an error during the header reception. The error types are:

- Break delimiter error (SDEF)
- Sync field error (SFEF)
- Identifier parity error (IDPEF)

Break Delimiter Error (SDEF)

The break delimiter must be 1 for at least one bit time; otherwise, it is considered short and the receiver discards synchronization on the current header. Consequently, the frame is discarded. An interrupt is generated if LINIER.HEIE is set.

Sync Field Error (SFEF)

The SFEF error condition is monitored differently depending on whether autosynchronization is enabled, as determined by LINCR1.LASE.

Case 1: Autosynchronization enabled (LINCR1.LASE = 1)

When autosynchronization is enabled, the SFEF flag indicates:

- The deviation error on the sync field is outside the LIN specification, which allows up to 14% of period deviation between the slave and master oscillators, or
- An overflow has occurred during the sync field measurement, which leads to an overflow of the divider registers.

NOTE. SFEF does not indicate that an inconsistent sync field value (other than 0x55) was received by the slave.

Deviation Error on the Sync Field

The deviation error is checked by comparing the current baud rate (relative to the slave oscillator) with the received LIN sync field (relative to the master oscillator).

This check is based on a measurement between the first falling edge and the last falling edge of the sync field. Referring to this period deviation as D:

- If SFEF is asserted, D > 14.0625%
- If there is no error, D < 14.84375%
- If 14.0625% < D < 14.84375%, then the sync field could be either consistent or inconsistent depending on de-phasing between the signal on LINFlexD_RX and ipg_baud_clk.

Overflow During Sync Field Measurement

Yuntu Microelectronics

The overflow check is based on the measurement of each bit time between both edges of the sync field. This checks that each of these bit times is large enough (more than 12 samples) compared to the bit time of the current baud rate.

Case 2: Autosynchronization Disabled (LINCR1.LASE = 0)

When autosynchronization is disabled, the sync character is received as a normal character. If the received character is 0x55, the sync field is OK.

On any occurrence of an inconsistent sync field, the receiver immediately exits from Sync_Field state, the frame is discarded, and LINESR.SFEF is set.

Identifier Parity Error (IDPEF)

There are two parity bits in the identifier field. The parity bits are checked against the hardware-calculated parity over the other six bits of identifier, according to the formula:

P0 = ID0 XOR ID1 XOR ID2 XOR ID4

P1 = NOT (ID1 XOR ID3 XOR ID4 XOR ID5)

The identifier parity is checked after the ID is transferred to the BIDR register (after stop of ID has been detected properly) and, upon parity mismatch, the receiver state machine exits from the identifier state immediately if LINCR2.IOPE is set.

Bit Error

A bit error is flagged in transmission mode when the value read back from the bus is different from the value transmitted. Bit error checking on each bit is ensured if transceiver delay is less than one bit time minus 6 ipg_baud_clk cycles, for example:

- One bit time at 20 Kbit/s = 50 µs
- 6 ipg_baud_clk cycles at 80 MHz = 75 ns
- Thus, (1-bit time) - (6 ipg_baud_clk cycles) = 49.925 µs

Bit error is not checked during break field transmission.

Transmission of the frame is stopped after the corrupted bit if LINCR2.IOBE = 1. If LINCR2.IOBE = 0, the transmitter continues to transmit despite the bit error. An interrupt is generated if LINIER.BEIE = 1.

NOTE. If the break delimiter is not detected by the master within one bit time after delimiter transmission because of transceiver delay or error on the bus, then a train of bit error interrupts is generated. In this case, the identifier may not be replaced in BIDR. Similarly, if the start of a falling edge of data is not detected by the transmitter node within one bit time after start bit transmission, then a train of bit error interrupts is generated. In this case, data and checksum replacements in DATA and LINCFR respectively are not ensured.

Framing Error

A framing error is flagged when a dominant state is sampled on the stop bit of the current received character (sync field, identifier field, data field, checksum field). The LINFlexD discards the current frame and returns to the Idle state. An interrupt is generated LINIER.FEIE is set.

The byte that caused framing error is shifted to the buffer, but LINSR.DRF is not set.

Checksum Error

A checksum error is flagged when the checksum computed by hardware does not match the received checksum.

Yuntu Microelectronics

The LINFlexD discards the received frame and returns to the Idle state. An interrupt is generated if LINIER.CEIE is set.

Overrun Error

After the message buffer is full (LINSR.RMB is set), the next valid message reception leads to an overrun and the message is lost. The LINFlexD hardware signals the overrun condition by setting the LINESR.BOF bit. Which message is lost depends on the buffer lock function control bit LINCR1.RBLM:

- If RBLM = 0, the old message in the buffer is overwritten by the most recent message.
- If RBLM = 1, the most recent message is discarded and the oldest message is available to the software. In slave mode, if the buffer is not released (RMB is not reset) before reception of next identifier and RBLM = 1, then the ID along with the data is discarded.

Timeout Error

A timeout error occurs in the case of incomplete response or no response within the timeout response period.

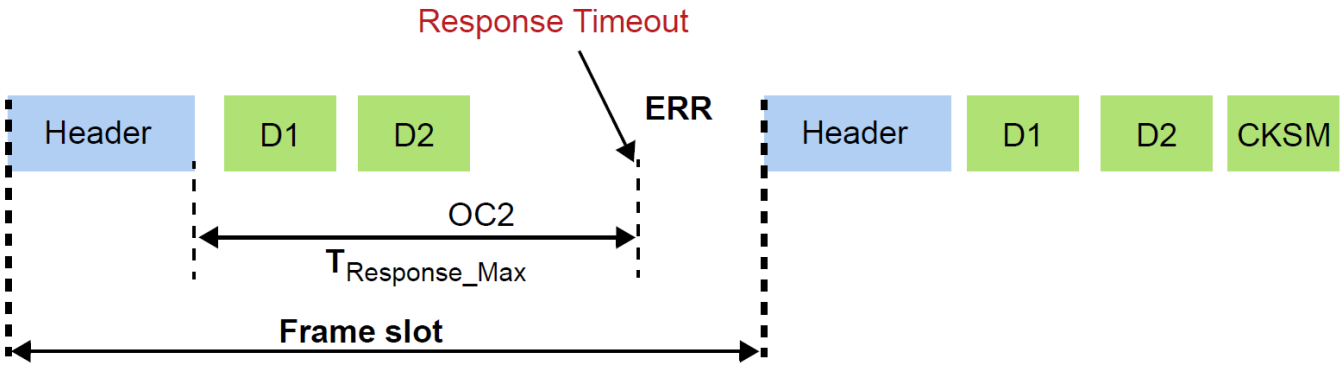


Figure 21.10: Incomplete Response (For Example, Missing Checksum)

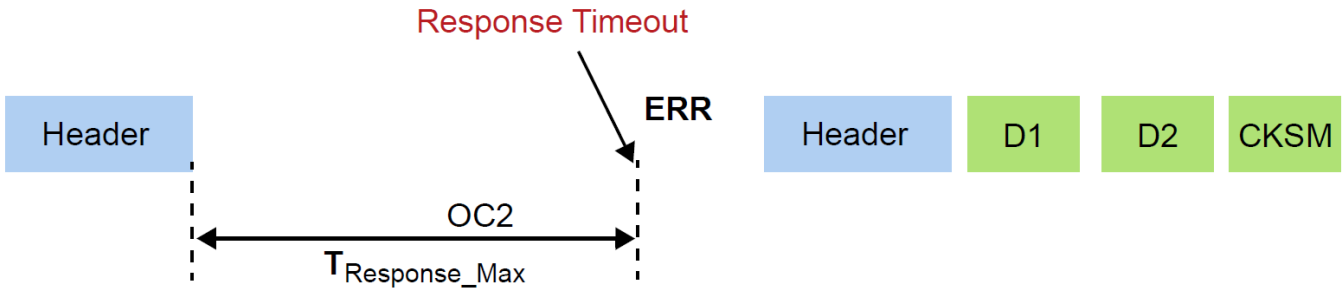


Figure 21.11: No Response

Response Timeout Mechanism

- Master mode:
  - OC2 is loaded with nominal_time_out + LINTCSR.CNT at the end of the stop bit of the identifier field (where nominal_time_out = 1.4 × ((DFL + 2) × 10 bit time). This loading takes place at the end of identifier field (not at the beginning of data field). Moreover, the correct value, which depends on DFL, is loaded immediately. No further OC2 update is required.
  - In case of no response at all within this time (in other words, start of data is not received), timeout takes place when the counter reaches the OC2 value.

  - In case of an incomplete response, timeout also occurs when the counter reaches the OC2 value.
- Slave mode:
  - **Case 1:** The received identifier is managed by a filter. The implementation can be similar to the master mode, because the DFL value is loaded by hardware. Thus, OC2 can be loaded with nominal_time_out + LINTCSR.CNT at the end of the stop bit of the identifier field (where nominal_time_out = 1.4 × ((DFL + 2) × 10 bit time)).
  - **Case 2:** The received identifier is not managed by a filter. As the DFL value needs to be updated by software after the identifier field has been received, the implementation is:
    * At the end of the ID, OC2 is loaded with 36 (maximum possible response space) + LINTCSR.CNT.
    * At the end of the first data byte, it is reloaded again according to DFL.
    * Before reloading, the LINFlexD checks the count_val. If count value is higher than the value to be reloaded, timeout takes place immediately and no reloading occurs.

NOTE. The value of nominal_time_out (1.4 × ((DFL +2) × 10 bit time)) shown is as per the reset value of LINTOCR.RTO and LINCR1.CFD. It changes with change in value of LINTOCR.RTO and LINCR1.CFD. Here, 1.4 corresponds to (LINTOCR.RTO)/10 and 2 corresponds to (2 - LINCR1.CFD)

Header Timeout Mechanism

- Master mode – As the header is generated by the LINFlexD, there are only two cases:
  - No error occurs on the bus and timing is correct (nominal header length).
  - An error occurs on the bus and the LINFlexD flags it in LINESR (typically a bit error).

  Therefore, there is no meaning of header timeout in master mode, so it is disabled.
- Slave mode:
  - header_nominal = 13 + 2 + 10 + 10 = 35 T_{bit}
  - header_max = 1.4 × header_nominal = 49 T_{bit}
  - Taking into account a possible 14% clock deviation, header_max seen by the LINFlexD is 49 × 1.14 = 55.86 T_{bit}
  - If the counter starts after 11 T_{bit} (break duration), the LINTOCR.HTO value is 55.86 - 11 = 44.86 T_{bit}

  LINTOCR.HTO can only be programmed in slave mode and has a reset value of 44. Counter restarts after break duration and LINOCR.OC1 is loaded with the value of LINTOCR.HTO.

NOTE. The timeout counter can be used to detect other timeouts. In this case, the LINTCSR.MODE bit must be reset and the output compare value can be updated in LINTOCR by software.

Stuck at Zero Timeout Error

If a dominant pulse lasts for a time of at least 100 bits, LINESR.SZF is set. If the same dominant pulse continues, the subsequent SZF settings occur 87 bit times apart (instead of 100 bit times).

Noise

During reception, each bit is sampled 16 times and the value of the bit is obtained by taking the majority value of the 8th, 9th, and 10th samples. If any one of these three samples has a value different from the other two, a noise error is flagged.

When UARTCR.OSR = 8, majority of 2nd, 3rd, and 4th samples are taken into account to determine noise. Noise checking is disabled for OSR values less than eight.

Yuntu Microelectronics

21.4.2.6 Identifier Filtering

In the LIN protocol, the identifier of a message is not associated with the address of a node; it is related to the content of the message. A transmitter broadcasts its message to all the receivers. Based on the header received, the receiver decides whether to receive or transmit a response (depending on the identifier value). If the message does not target the node, the message should be discarded.

To fulfill this requirement, the LINFlexD provides configurable filters in order to eliminate software intervention. Hardware filtering saves CPU resources that would otherwise be required to perform filtering by software.

There are a maximum of 16 filters (depending on configuration parameter no_of_filters) in the LINFlexD. The filters can be programmed only during Initialization mode. To activate a filter, the corresponding FACT bit must be set in IFER. There are two modes possible for each identifier, depending on the corresponding IFMR.IFM bit.

Identifier List Mode

If IFMR.IFM[n] = 0, filters 2n and 2n+1 are in identifier list mode. The maximum number of filters that can be configured for transmission/reception is no_of_filters, depending on the value of IFER.FACT. The identifier received must match bit-by-bit to the ID field of IFCR2n or IFCR2n+1 (if the corresponding filter is enabled in IFER.FACT).

Identifier Mask Mode

If the number of filters required is more than no_of_filters, then filters should be configured in mask mode. If IFMR.IFM[n] = 1, filters 2n and 2n+1 are in mask mode. Filter 2n is the filter and filter 2n+1 acts as a mask for it. The IFER.FACT bit for filter 2n+1 has no effect in this case. If bit x of the mask is set, then bit x of the received identifier must match bit x of filter.

In either filter mode, if there is a match of identifier with filter number m, value m+1 is loaded into the IFMI register by hardware. IFMI = 0 indicates a no-match condition.

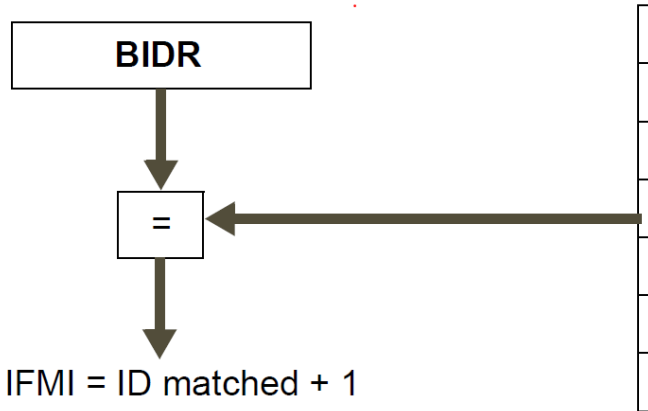
On a matching ID, the LINFlexD hardware copies the DFL[2:0], CCS, and DIR values from the matching filter to the BIDR register; from then on, the BIDR register is read-only until the end of the frame. BIDR.DIR determines the transfer direction:

- If BIDR.DIR = 1, the direction is transmit. A Tx interrupt (ipi_int_tx) is generated if LINIER.HRIE is set. Software can read the filter match index from the IFMI register, then transfer the relevant data from the RAM area to the DATA registers. After transferring the transmit data to the DATA registers, set LINCR2.DTRQ to start the transmission.
- If BIDR.DIR = 0, the direction is receive. If LINIER.DRIE is set, an Rx interrupt (ipi_int_rx) is generated when the checksum has been received and there is no checksum error.

In the case of no filter match condition (IFMI = 0), if LINCR1.BF = 1, an Rx interrupt (ipi_int_rx) is generated. It is the responsibility of software to configure BIDR and start transmission (by loading the DATA registers and setting LINCR2.DTRQ) or discard reception (by setting LINCR2.DDRQ). If LINCR1.BF = 0, the receiver discards the received identifier and turns to Idle state in search of a new break.

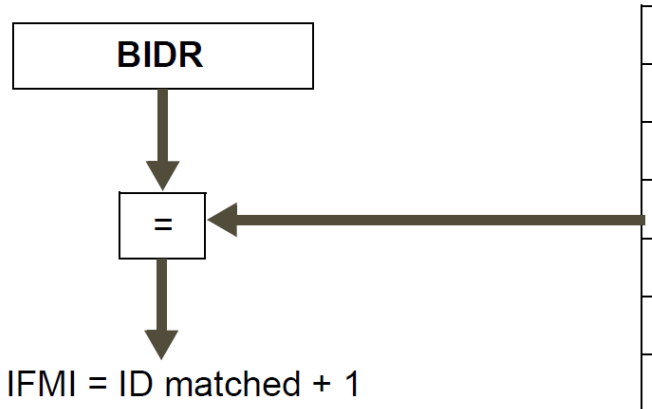
NOTE. If one identifier matches with two filters (one in list mode and the other in mask mode), then list mode prevails over mask mode. In mask mode, if two filters match with the identifier, the filter with the lower number prevails.

Yuntu Microelectronics



IFCR0	ID5	ID4	ID3	ID2	ID1	ID0
IFCR1	ID5	ID4	ID3	ID2	ID1	ID0
IFCR2	ID5	ID4	ID3	ID2	ID1	ID0
IFCR3	ID5	ID4	ID3	ID2	ID1	ID0
IFCR4	ID5	ID4	ID3	ID2	ID1	ID0
IFCR5	ID5	ID4	ID3	ID2	ID1	ID0
IFCR6	ID5	ID4	ID3	ID2	ID1	ID0
IFCR7	ID5	ID4	ID3	ID2	ID1	ID0
...	ID5	ID4	ID3	ID2	ID1	ID0
IFCR*n*	ID5	ID4	ID3	ID2	ID1	ID0

Figure 21.12: Identifier List Mode



IFCR0	ID5	ID4	ID3	ID2	ID1	ID0
IFCR1	0	0	0	1	1	1
IFCR2	ID5	ID4	ID3	ID2	ID1	ID0
IFCR3	1	1	1	1	0	0
IFCR4	ID5	ID4	ID3	ID2	ID1	ID0
IFCR5	1	1	1	1	0	1
IFCR6	ID5	ID4	ID3	ID2	ID1	ID0
IFCR7	1	1	1	1	1	0
...	...	...	...	...	...	...
IFCR*n*	...	...	...	...	...	...

$IFCR_{2n+1}$ = Mask for $IFCR_{2n}$

Figure 21.13: Identifier Mask Mode

21.4.2.7 Start Detection and Break Delimiter Detection in Receiver

There is a 10-bit-shift sample register in the receiver that shifts signal data to the next least significant bit on each incoming sample.

A start is detected as soon as it is qualified with 1110 in the sample register and verified with at least two out of three predefined verification samples being zero. Similarly, for delimiter detection, qualification samples are 0001 and at least two of the three verification samples are one. The Noise Flag is set if the start is verified with only two valid samples.

Table 21.32: Start detection and delimiter detection in receiver

9	8	7	6	5	4	3	2	1	0	Sample register
−−	−−	−−	−−	−−	−−	−−	−−	−−	RT1	counter = 0
−−	−−	−−	−−	−−	−−	−−	−−	RT1	RT2	counter = 1
−−	−−	−−	−−	−−	−−	−−	RT1	RT2	RT3	counter = 2
−−	−−	−−	−−	−−	−−	RT1	RT2	RT3	RT4	counter = 3
−−	−−	−−	−−	−−	RT1	RT2	RT3	RT4	RT5	counter = 4
−−	−−	−−	−−	RT1	RT2	RT3	RT4	RT5	RT6	counter = 5
−−	−−	−−	RT1	RT2	RT3	RT4	RT5	RT6	RT7	counter = 6
−−	−−	RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	counter = 7
−−	RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	counter = 8
RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	counter = 9
RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	counter = 10
RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	counter = 11
RT4	RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	RT13	counter = 12
RT5	RT6	RT7	RT8	RT9	RT10	RT11	RT12	RT13	RT14	counter = 13
RT6	RT7	RT8	RT9	RT10	RT11	RT12	RT13	RT14	RT15	counter = 14
RT7	RT8	RT9	RT10	RT11	RT12	RT13	RT14	RT15	RT16	counter = 15
1	1	1	0	x	0	x	0	x	0	start detect
1	1	1	0	x	0	x	0	x	1	
1	1	1	0	x	0	x	1	x	0	
1	1	1	0	x	1	x	0	x	0	
0	0	0	1	x	1	x	1	x	1	delimiter detect
0	0	0	1	x	1	x	1	x	0	
0	0	0	1	x	1	x	0	x	1	
0	0	0	1	x	0	x	1	x	1	

Start Detection Mechanism

The process for detecting a start bit is:

1. See the table above for the status of the sample register (shift register) when count = 6. If RT1 (the first incoming sample) = 0 and the previous three samples already received are all ones, then it might be a start bit.
2. To make sure it is a start bit, the incoming data samples in sample register (4), (2), and (0), which correspond to RT3, RT5, and RT7 are verified using the steps mentioned below.
3. If the majority (2 or more) out of these 3 samples are equal to '0', a start bit is detected. These three samples RT3, RT5, and RT7 are called the "verification samples" which are checked at count = 6.
4. Sample register bits 9, 8, 7, and 6 are called "qualification samples" which are checked at count = 6.
5. At count = 9, if majority value of RT8, RT9, and RT10 is not equal to '0', the Noise Flag is set.

Break Delimiter Detection Mechanism

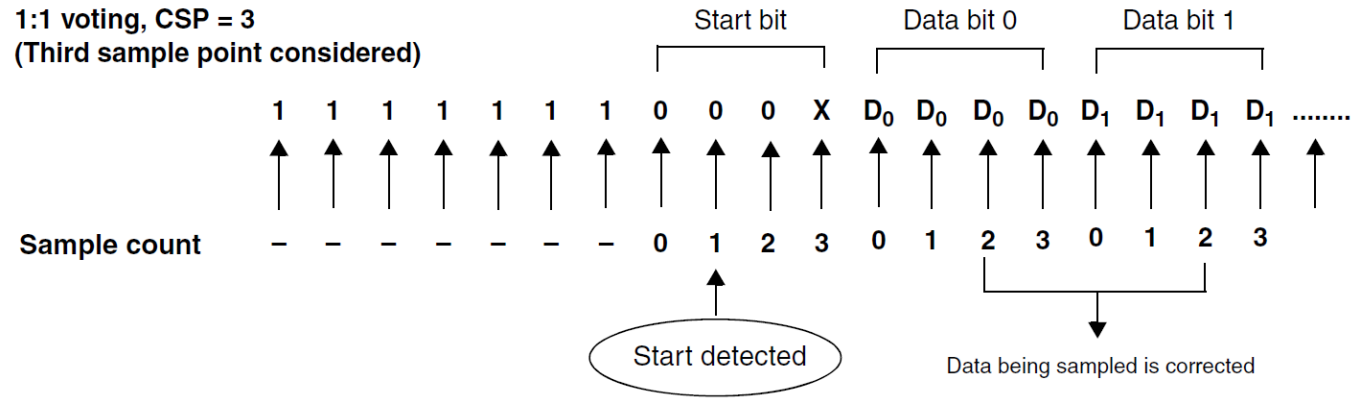
The process for detecting a break delimiter is:

1. See the table above for the status of the sample register (shift register) when count = 6. If RT1 (the first incoming sample) = 1 and the previous three samples already received are all zeros, then it might be a delimiter bit.

Yuntu Microelectronics

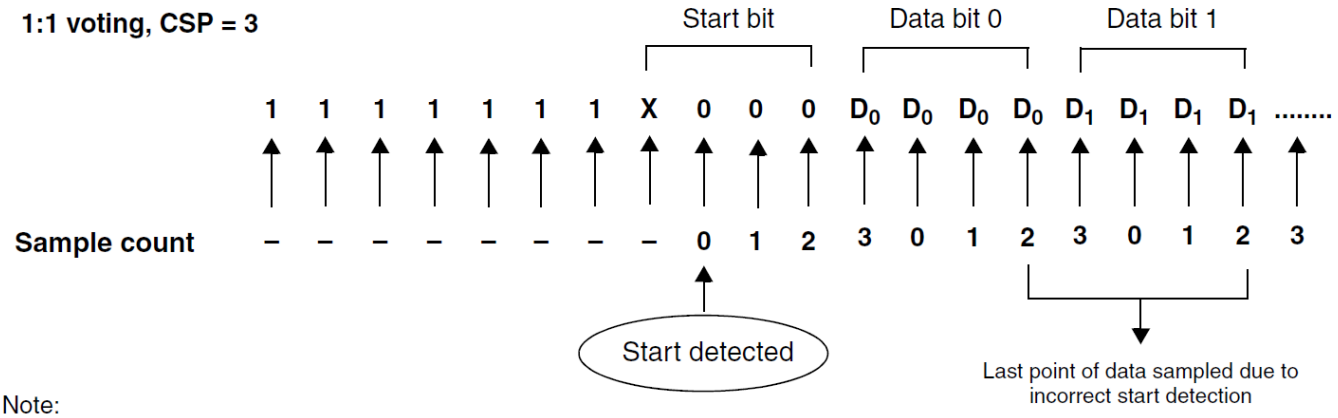
2. To make sure it is a break delimiter, the incoming data samples in the sample register (4), (2), and (0), which correspond to RT3, RT5, and RT7 are verified using the steps mentioned below.
3. If the majority (2 or more) out of these 3 samples are equal to '1', then a break delimiter bit is detected. These three samples RT3, RT5, and RT7 are called the "verification samples" which are checked at count = 6.
4. The sample register bits 9, 8, 7, and 6 are called "qualification samples" which are checked at count = 6.

Therefore, a start is detected as soon as it is qualified with 1110 in the sample register and verified with at least two out of three predefined verification samples being zero. Similarly for delimiter detection, qualification samples are 0001 and at least two out of the three verification samples are one. The Noise Flag is set if the start is verified with only two valid samples.



Note:
1) D_0, D_1: Databits
2) X: Due to sampling during transition of the bit, value might be sampled as a '0' or '1' depending on the rise/fall time. Therefore 'X' refers to a indeterministic value.

Figure 21.14: Start Detection and Sampling for Oversampling Rate = 4 (Case 1)

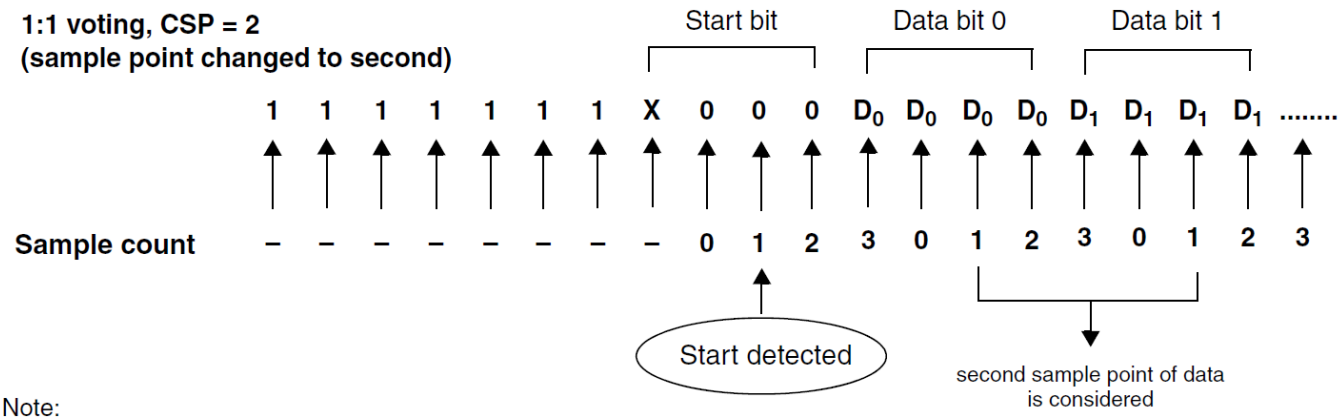


Note:
1) D_0, D_1: Databits
2) X: Due to sampling during transition of the bit, value might be sampled as a '0' or '1' depending on the rise/fall time. Therefore 'X' refers to a indeterministic value.

Figure 21.15: Start Detection and Sampling for Oversampling Rate = 4 (Case 2a)

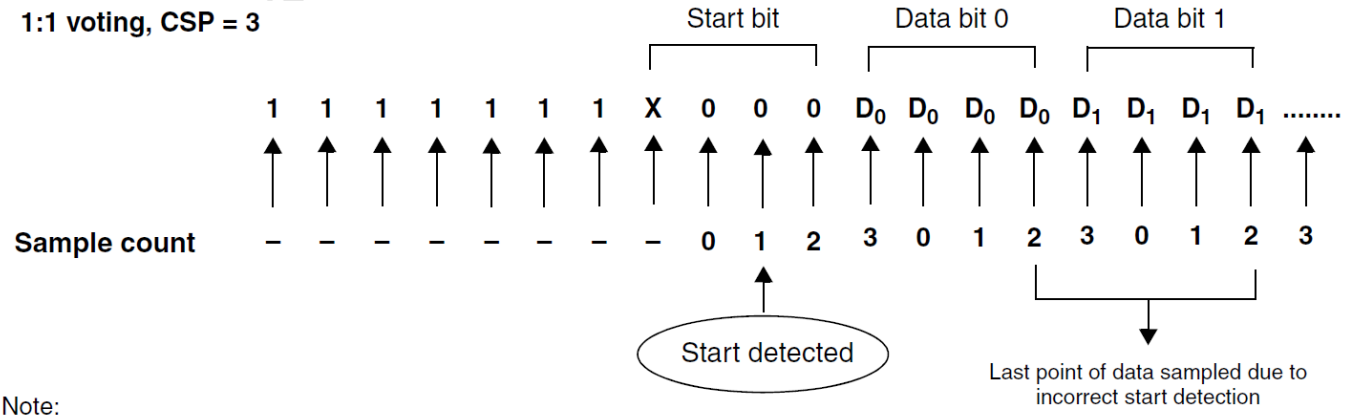
The choice of the correct sample points depends on the external Rx signal quality. During the application development process, the error information indicated by the parity error can help to find the best setting for an application-specific hardware signal.

To sample at the correct point therefore, CSP has to be changed to two; resulting in the following situation:



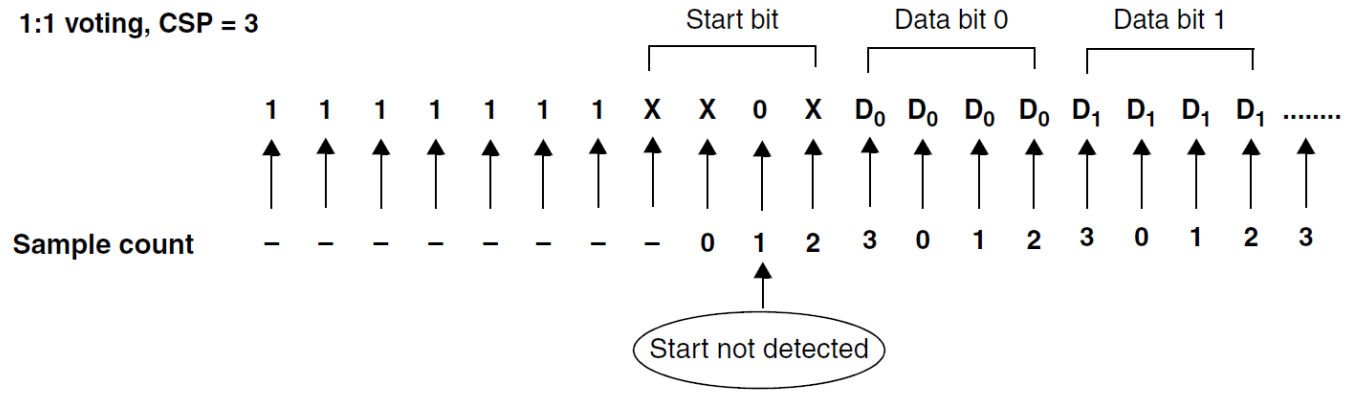
Note:
1) D_0, D_1: Databits
2) X: Due to sampling during transition of the bit, value might be sampled as a '0' or '1' depending on the rise/fall time. Therefore 'X' refers to a non deterministic value.

Figure 21.16: Start Detection and Sampling for Oversampling Rate = 4 (Case 2b)



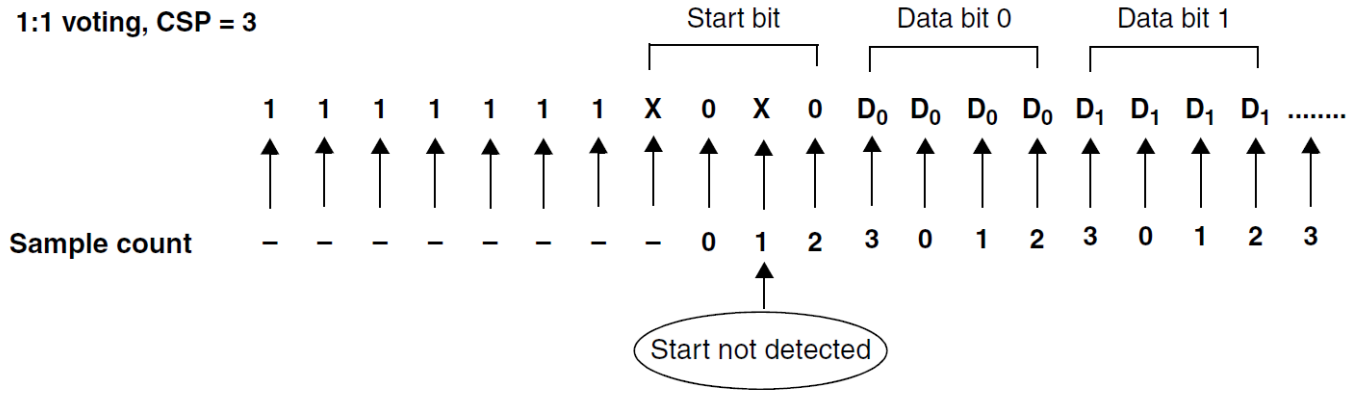
Note:
1) D_0, D_1: Databits
2) X: Due to sampling during transition of the bit, value might be sampled as a '0' or '1' depending on the rise/fall time. Therefore 'X' refers to a non deterministic value.

Figure 21.17: Start Detection and Sampling for Oversampling Rate = 4 (Case 3)



Note:
1) D_0, D_1: Databits
2) X: Due to sampling during transition of the bit, value might be sampled as a '0' or '1' depending on the rise/fall time. Therefore 'X' refers to a non deterministic value.

Figure 21.18: Start Detection and Sampling for Oversampling Rate = 4 (Case 4)



Note:
1) D_0, D_1: Databits
2) X: Due to sampling during transition of the bit, value might be sampled as a '0' or '1' depending on the rise/fall time. Therefore 'X' refers to a non deterministic value.

Figure 21.19: Start Detection and Sampling for Oversampling Rate = 4 (Case 5)

21.4.2.8 Baud Rate Generation

The LIN baud rate is programmed in two registers: the LIN Integer Baud Rate Register (LINIBRR) and the LIN Fractional Baud Rate Register (LINFBRR). These registers can be programmed only during Initialization mode.

Baud rate is calculated with the following formula for both receiver and transmitter:

- When UARTCR.ROSE = 1, Tx = Rx = ipg_baud_clk / (OSR × LDIV)
- When UARTCR.ROSE = 0, Tx = Rx = ipg_baud_clk / (16 × LDIV)

Where ipg_baud_clk is the frequency of the baud clock, ipg_baud_clk.

LDIV is an unsigned fixed point number. The mantissa is coded into 20 bits of LINIBRR and the fraction is coded on 4 bits of LINFBRR.

When reduced oversampling is enabled, LINFBRR should not be used and programmed to zero and LDIV contains only the integer part of LINIBRR.

For example: When ROSE = 0 (for LIN and UART mode): LDIV = 468.75 d, ipg_baud_clk = 36 MHz, LINIBRR = 468 d, LINFBRR = 12, Baud rate = 36 MHz / (16 × 468.75) = 4.8 Kbit/s **For example:** When ROSE = 1 (only for UART mode): LDIV = 5 d, ipg_baud_clk = 80 MHz, LINIBRR = 5 d, OSR = 4, Baud rate = LIN_CLK / (OSR × LDIV) = 80 MHz / (4 × 5) = 4 Mbit/s

21.4.2.9 Automatic Resynchronization

To automatically adjust the baud rate based on measurement of the LIN sync field, write the nominal prescalar value (nominal baud rate) in LINIBRR and LINFBRR, then set LINCR1.LASE to enable automatic synchronization.

When auto-synchronization is enabled, after each break delimiter, the time duration between five falling edges on LINFlexD_RX is sampled on ipg_baud_clk.

21.4.2.10 Wakeup Management

Any node in a sleeping LIN cluster may request a wakeup. The wakeup request is issued by forcing the bus to the dominant state for 250 µs to 5 ms. Every slave node should detect the wakeup request (a dominant pulse longer than 150 µs) and be ready to listen to bus commands within 100 ms, measured from the ending edge of the dominant pulse. The master also wakes on detecting a wakeup request and when the slaves are ready, starts sending frame headers to find out the cause of the wakeup. If the master does not issue a frame header within 150 ms from the wakeup request, then the node issuing a request may try issuing a new wakeup request.

In the LINFlexD, a wakeup request can be generated by writing the wakeup character in DATA0 and setting LINCR2.WURQ. On setting the WURQ bit, the character in DATA0 is transmitted.

In the LINFlexD, wakeup can be detected in two ways:

- AUTOWU = 1: On detecting a falling edge in sleep mode, the LINCR1.SLEEP bit is cleared by hardware, LINCR1.WUF is set, and an interrupt is generated if LINIER.WUIE = 1. The LINFlexD is now in normal mode and ready to receive frames.
- AUTOWU = 0: On detecting a falling edge, LINCR1.WUF is set and an interrupt is generated (if LINIER.WUIE = 1). It is then up to the software to clear LINCR1.SLEEP.

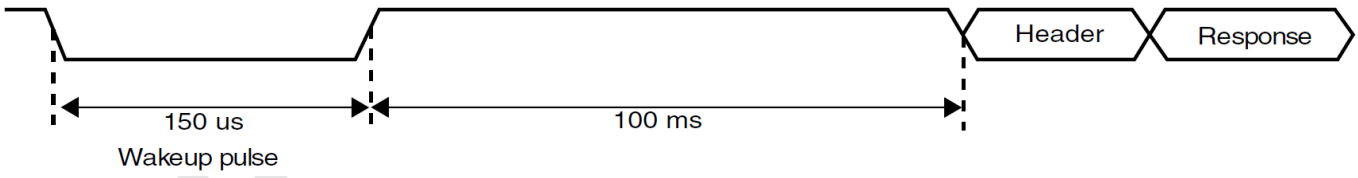


Figure 21.20: Wakeup Sequence

21.4.3 Timer

The LINFlexD includes an 8-bit counter, which has different behavior in different modes:

- When the counter is in output compare mode (LINTCSR.MODE = 1), the counter runs even in Sleep and Initialization modes. When the counter value matches LINOCR.OC1 or LINOCR.OC2, and LINTCSR.TOCE is set, the output compare flag (LINESR.OCF) is set. An error interrupt (ipi_int_err) is generated if LINIER.OCIE is set.
- When the counter is in LIN mode (LINTCSR.MODE = 0), software has no control over the TOCE bit and output compare registers are utilized for generation of LIN timeout interrupts (header,

Yuntu Microelectronics

frame, and response timeouts). If the LINFlexD goes to Sleep mode or Initialization mode, the counter remains in the reset state. When UART mode is enabled, LIN mode for the counter has no meaning; the counter remains in the reset state.

21.4.4 UART Mode

Main features in the UART mode are:

- Full-duplex communication
- 8-bit frames, 9-bit frames, 13-bit frames, 16-bit frames, 17-bit frames
- Even/odd/0/1 parity
- User programmable oversampling rate to obtain the baud rate of up to ipg_baud_clk/4 Mbit/s

21.4.4.1 8-Bit Data Frames

The eighth bit can be a data bit or a parity bit. Even/odd/0/1 parity can be selected by UARTCR.PC[1:0]. An even parity is set if the modulo-2 sum of the seven data bits is one. An odd parity is cleared in this case.

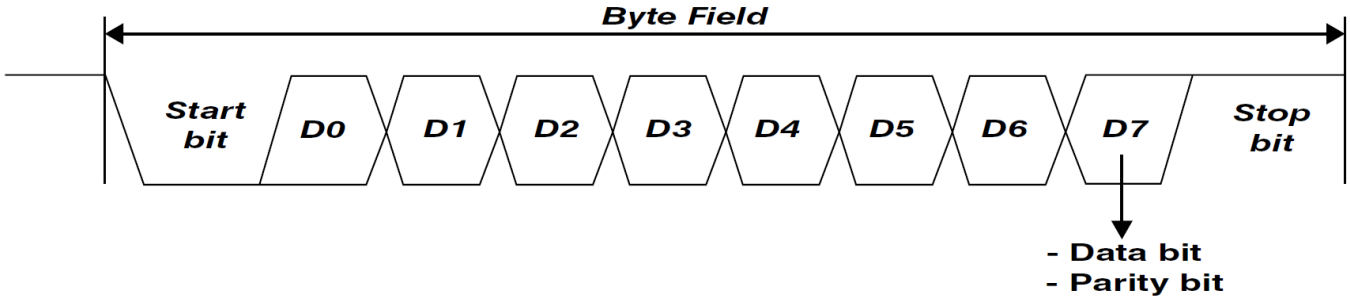


Figure 21.21: UART Mode 8-Bit Data Frame

21.4.4.2 9-Bit Frames

The ninth bit should be a parity bit. Even/odd/0/1 parity can be selected by the UARTCR.PC[1:0]. An even parity is set if the modulo-2 sum of the eight data bits is one.

An odd parity is cleared in this case. Parity 0 forces a zero logical value. Parity 1 forces a high logical value.

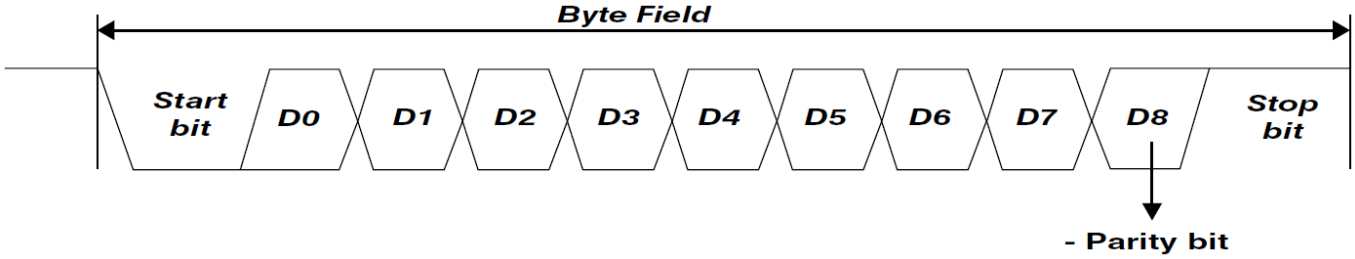


Figure 21.22: UART Mode 9-Bit Data Frame

21.4.4.3 13-Bit Frames

When UARTCR.WLS = 1, special word length is selected in UART mode. Special word length enables 12-data-bit + parity bit reception only in FIFO mode.

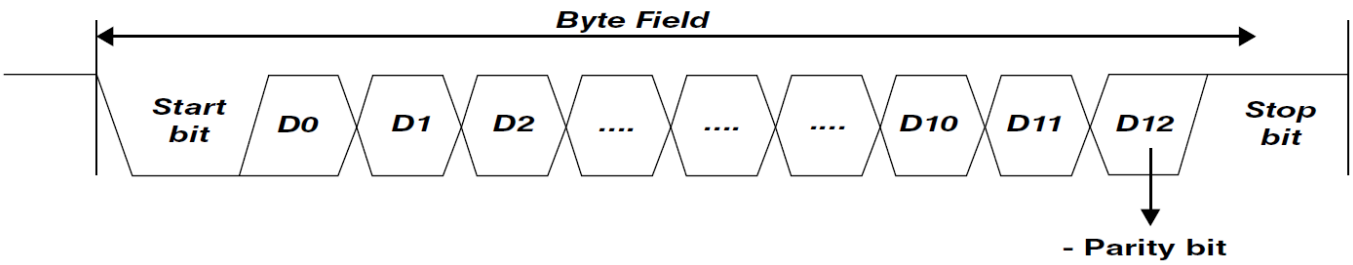


Figure 21.23: UART Mode 13-Bit Data Frame

21.4.4.4 Buffer in UART Mode

The 8-byte buffer is divided into two parts: one for receiver and one for transmitter, shown as below.

Tx0	BDR0 (BDRL.DATA0)
Tx1	BDR1 (BDRL.DATA1)
Tx2	BDR2 (BDRL.DATA2)
Tx3	BDR3 (BDRL.DATA3)
Rx0	BDR4 (BDRM.DATA4)
Rx1	BDR5 (BDRM.DATA5)
Rx2	BDR6 (BDRM.DATA6)
Rx3	BDR7 (BDRM.DATA7)

Figure 21.24: Structure of 8 Byte Buffer

21.4.4.5 UART Transmitter

To start transmission in the UART mode, set UARTCR.UART and UARTCR.TxEN. Transmission starts when the BDR0 (least significant data byte) is programmed and continues until the number of bytes/halfwords transmitted is equal to the value in the UARTCR.TDFL bits.

The transmit buffer is four bytes (when UARTCR.WL1 = 0) or two halfwords (when UARTCR.WL1 = 1). Therefore, a maximum of four bytes (two halfwords) can be triggered for transmission. Once the programmed number of bytes or halfwords is transmitted, the UARTSR.DTF flag is set. If the UARTCR.TxEN bit is reset in the middle of a transmission, the current transmission is completed and no further transmissions are invoked.

The buffer can be configured in FIFO mode (mandatory when the DMA Tx is enabled) by setting control bit UARTCR.TFBM.

NOTE. If UARTSR.TFF = 1 and a write is performed to the FIFO, the data transmitted may be erroneous.

Table 21.33: DATA0 DATA3 access in UART mode

BUS operation	Register	Mode (UARTCR.TFBM)	Word length (UARTCR.WL1)	Peripheral Bus Operation Result
Write Byte0	DATA0~DATA3	FIFO	Byte	OK
Write Byte1-2-3	DATA0~DATA3	FIFO	Byte	Transfer error
Write Half-word0-1	DATA0~DATA3	FIFO	Byte	Transfer error
Write Word	DATA0~DATA3	FIFO	Byte	Transfer error
Write Byte0-1-2-3	DATA0~DATA3	FIFO	Half-word	Transfer error
Write Half-word0	DATA0~DATA3	FIFO	Half-word	OK
Write Half-word1	DATA0~DATA3	FIFO	Half-word	Transfer error
Write Word	DATA0~DATA3	FIFO	Half-word	Transfer error
Read Half-word0-1	DATA0~DATA3	FIFO	Byte/Half-word	Transfer error
Read Byte0-1-2-3	DATA0~DATA3	FIFO	Byte/Half-word	Transfer error
Read Word	DATA0~DATA3	FIFO	Byte/Half-word	Transfer error
Write Byte0-1-2-3	DATA0~DATA3	BUFFER	Byte/Half-word	OK
Write Half-word0-1	DATA0~DATA3	BUFFER	Byte/Half-word	OK
Write Word	DATA0~DATA3	BUFFER	Byte/Half-word	OK
Read Byte0-1-2-3	DATA0~DATA3	BUFFER	Byte/Half-word	OK
Read Half-word0-1	DATA0~DATA3	BUFFER	Byte/Half-word	OK
Read Word	DATA0~DATA3	BUFFER	Byte/Half-word	OK

In UART FIFO mode (UARTCR.TFBM = 1), any read operation causes a peripheral bus transfer error (ipx_xfr_err).

21.4.4.6 UART Receiver

The reception of a data byte is started as soon as software exits Initialization mode (by clearing LINCR1.INIT), sets UARTCR.RxEn, and detects a start bit. There is a dedicated 4-byte (if UARTCR.WL1 = 0) or 2-halfword (if UARTCR.WL1 = 1) data buffer for received data. Once the programmed number (UARTCR.RDFL) of bytes have been received, the UARTSR.DRF flag is set, and the current reception is completed. UARTCR.RxEn needs to be set only to start the reception. The reception is automatically completed as soon as the programmed number of bytes are received.

The buffer can be configured in FIFO mode (mandatory when the DMA Rx is enabled) by setting the control bit UARTCR.RFBM.

Table 21.34: DATA4 DATA7 access in UART mode

BUS operation	Register	Mode (UARTCR.RFBM)	Word length (UARTCR.WL1)	Peripheral Bus Operation Result
Read Byte4	DATA4~DATA7	FIFO	Byte	OK

Yuntu Microelectronics

Table 21.34 continued from previous page

BUS operation	Register	Mode (UARTCR.RFBM)	Word length (UARTCR.WL1)	Peripheral Bus Operation Result
Read Byte5-6-7	DATA4~DATA7	FIFO	Byte	Transfer error
Read Half-word2-3	DATA4~DATA7	FIFO	Byte	Transfer error
Read Word	DATA4~DATA7	FIFO	Byte	Transfer error
Read Byte4-5-6-7	DATA4~DATA7	FIFO	Half-word	Transfer error
Read Half-word2	DATA4~DATA7	FIFO	Half-word	OK
Read Half-word3	DATA4~DATA7	FIFO	Half-word	Transfer error
Read Word	DATA4~DATA7	FIFO	Half-word	Transfer error
Write Byte4-5-6-7	DATA4~DATA7	FIFO	Byte/Half-word	Transfer error
Write Half-word2-3	DATA4~DATA7	FIFO	Byte/Half-word	Transfer error
Write Word	DATA4~DATA7	FIFO	Byte/Half-word	Transfer error
Read Byte4-5-6-7	DATA4~DATA7	BUFFER	Byte/Half-word	OK
Read Half-word2-3	DATA4~DATA7	BUFFER	Byte/Half-word	OK
Read Word	DATA4~DATA7	BUFFER	Byte/Half-word	OK
Write Byte4-5-6-7	DATA4~DATA7	BUFFER	Byte/Half-word	Transfer error
Write Half-word2-3	DATA4~DATA7	BUFFER	Byte/Half-word	Transfer error
Write Word	DATA4~DATA7	BUFFER	Byte/Half-word	Transfer error

NOTE. Refer to the layout of the registers DATA to identify the mapping between byte x and data bits of registers DATA.

Additional Notes:

- If it is not known in advance how many bytes are to be received, RDFL should not be programmed in advance. The reset value of RDFL is zero, which ensures that reception occurs byte-by-byte. The state machine goes to the Idle state after each received byte.
- If RDFL is programmed for a certain value but that number of bytes are not received, reception hangs. UARTSR.TO is set when a reception timeout occurs.
- If ipg_stop is asserted in the middle of a reception, it is only acknowledged after all the programmed number of data bytes have received. If the programmed number of data bytes are not received and a reception timeout occurs, the ipg_stop request is acknowledged when the receiver FSM is reset to the Idle state.
- If a parity error occurs during reception of any byte, the corresponding UARTSR.PE[n] bit is set. No interrupt is generated in this case. If a framing error occurs in an byte (UARTSR.FEF is set), an interrupt is generated if LINIER.FEIE =1. As there is only one register bit for framing error, this interrupt helps to identify which byte has a framing error.
- If the last received frame has not been read from the buffer (in other words, RMB is not reset by software), then upon reception of the next byte, an overrun error occurs (UARTSR.BOF is set). An interrupt is generated if LINIER.BOIE = 1. The new message is either discarded or overwrites the previous message in the buffer.
- When ipg_stop is requested when the LINFlexD is in UART FIFO mode, software must set LINCR.INIT bit to set the receiver FSM to Idle, at which point the LINFlexD asserts ipg_stop_ack.

Yuntu Microelectronics

21.4.5 Interrupts

The table below summarizes the LINFlexD interrupt events and corresponding interrupt enables.

Table 21.35: Interrupts

Interrupt Event	Event Flag in LINSR/LINESR/UARTSR	Interrupt Enable Bit in LINIER	Interrupt Vector
Stuck at zero	SZF	SZIE	Error
Output compare	OCF	OCIE	Error
Bit error	BEF	BEIE	Error
Checksum error	CEF	CEIE	Error
Header error	SFEF or SDEF or IDPEF	HEIE	Error
Frame error	FEF	FEIE	Error
Buffer Overrun error	BOF	BOIE	Error
UART Timeout error	TO	TOIE	Error
LIN state	Sync Del, Sync Field, Identifier field, or Checksum Field	LSIE	Rx
Wakeup	WUF	WUIE	Rx
Data Reception Complete	DRF	DRIE	Rx
Data transmitted	DTF	DTIE	Tx
Header received	HRF	HRIE	Rx
Header received	HRF	HRIE	Tx (if there is a filter match for Tx identifier)

Yuntu Microelectronics

The figure below shows the mapping interrupt flags to the LINFlexD interrupt output signals.

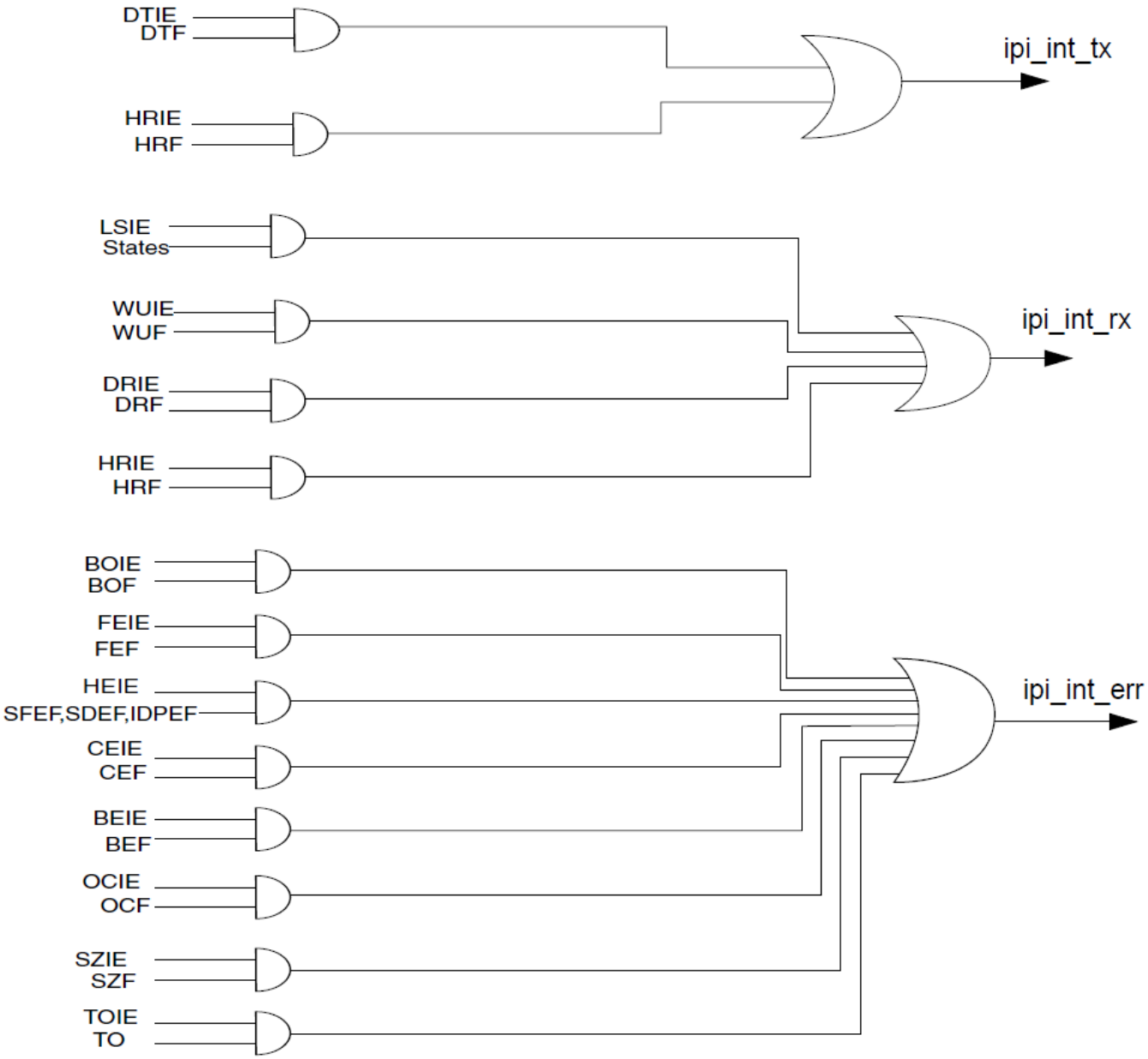


Figure 21.25: Interrupt Diagram

NOTE. Rx interrupt due to LIN state event is cleared by writing LINSR.LINS field with 0xF when LSIE is enabled.

NOTE. In Loop Back mode, Rx interrupt is set earlier than Tx interrupt.

21.4.6 DMA Operations

When configuration parameter DMA_SUPPORTED = 1, the LINFlexD supports 16 DMA channels each for transmit and receive. Use of the DMA feature depends on the LINFlexD operating mode (LIN master, LIN slave, or UART) and whether the LINFlexD is the transmitter or receiver of data.

21.4.6.1 LIN Master Node: Transmit DMA Channel

As a LIN master node, the LINFlexD transmits the LIN frame header and, if the LINFlexD is the publisher of the data, the LINFlexD also transmits the response (data) portion of the LIN frame.

The LINFlexD uses a single transmit DMA channel (0) for both the header and transmit data. To enable DMA in LIN master mode, both DMATXE[0] and input signal dma_enb_req_tx[0] must be asserted.

The table below shows the data to be transferred from RAM to LINFlexD registers in response to assertion of ipd_req_tx[0] for a single LIN frame depending on frame type:

- Master to slave: Master transmits frame header and data.
- Slave to master: Master transmits frame header. Use receive DMA channel 0 to receive data.
- Slave to slave: Master transmits header.

In the cycle after the data phase of the last write for the transfer, the DMA controller must assert both dma_done_lw_tx[0] and dma_ipd_complete_tx[0] to acknowledge the DMA request.

After the DMA transfer is complete to load the header (and data for a master to slave transfer), write '1' to LINCR2.HTRQ or assert in_trig_htrq to start the header transmission.

Table 21.36: LIN master node transmit DMA transfers

Frame Type	DMA Transfer			Data Values
	From RAM	→	To LINFlexD Register	
Master to slave	LINCR2 data	→	LINCR2	DDRQ = 1 DTRQ = 0 HTRQ = 0
	BIDR data	→	BIDR	DFL = payload size ID = identifier CCS = checksum type DIR = 1
	Payload data bytes 0-3	→	DATA0-3	1-4 data bytes depending on DFL
	Payload data bytes 4-7	→	DATA4-7	0-4 data bytes depending on DFL
Slave to master	LINCR2 data	→	LINCR2	DDRQ = 0 DTRQ = 0 HTRQ = 0
	BIDR data	→	BIDR	DFL = payload size ID = identifier CCS = checksum type DIR = 0
Slave to slave	LINCR2 data	→	LINCR2	DDRQ = 0 DTRQ = 0 HTRQ = 0
	BIDR data	→	BIDR	DFL = payload size ID = identifier CCS = checksum type DIR = 0

21.4.6.2 LIN Master Node: Receive DMA Channel

As a LIN master node, the LINFlexD uses a single receive DMA channel (0) to receive data for a slave-to-master frame.

Yuntu Microelectronics

To enable the DMA receive channel in LIN master mode, both DMARXE[0] and input signal dma_enb_req_rx[0] must be asserted.

The table below shows the data to be transferred to RAM from LINFlexD registers in response to assertion of ipd_req_rx[0] for a single LIN frame.

In the cycle after the data phase of the last read for the transfer, the DMA controller must assert both dma_done_lw_rx[0] and dma_ipd_complete_rx[0] to acknowledge the DMA request.

Table 21.37: LIN master node receive DMA transfers

Frame Type	DMA Transfer			Data Values
	From LINFlexD Register	→	To RAM	
Slave to master	BIDR (optional)	→	BIDR data	DFL = payload size ID = identifier
	DATA0-3	→	Payload data bytes 0–3	1–4 data bytes depending on DFL
	DATA4-7	→	Payload data bytes 4–7	0–4 data bytes depending on DFL

21.4.6.3 LIN Slave Node: Transmit DMA Channel

As a LIN slave node, the LINFlexD transmits the response (data) portion of the LIN frame.

The LINFlexD uses one transmit DMA channel for each ID filter that is programmed in transmit mode. To enable transmit DMA channel n in LIN slave mode, both DMATXE[n] and input signal dma_enb_req_tx[n] must be asserted.

The table below shows the data to be transferred from RAM to LINFlexD registers in response to assertion of ipd_req_tx[n] for a single LIN frame.

In the cycle after the data phase of the last write for the transfer, the DMA controller must assert both dma_done_lw_tx[n] and dma_ipd_complete_tx[n] to acknowledge the DMA request.

Table 21.38: LIN slave node transmit DMA transfers

Frame Type	DMA Transfer			Data Values
	From RAM	→	To LINFlexD Register	
Slave to master or slave to slave	Payload data bytes 0–3	→	DATA0-3	1–4 data bytes depending on IFCRn.DFL
	Payload data bytes 4–7	→	DATA4-7	0–4 data bytes depending on IFCRn.DFL

21.4.6.4 LIN Slave Node: Receive DMA Channel

As a LIN slave node, the LINFlexD uses one receive DMA channel for each ID filter that is programmed in receive mode. To enable receive DMA channel n in LIN slave mode, both DMARXE[n] and input signal dma_enb_req_rx[n] must be asserted.

Yuntu Microelectronics

The table below shows the data to be transferred to RAM from LINFlexD registers in response to assertion of ipd_req_rx[n] for a single LIN frame.

In the cycle after the data phase of the last read for the transfer, the DMA controller must assert both dma_done_lw_rx[0] and dma_ipd_complete_rx[0] to acknowledge the DMA request.

Table 21.39: LIN slave node receive DMA transfers

Frame Type	DMA Transfer			Data Values
	From LINFlexD Register	→	To RAM	
Master to slave or slave to slave	BIDR	→	BIDR data	DFL = payload size ID = identifier
	DATA0-3	→	Payload data bytes 0–3	1–4 data bytes depending on DFL
	DATA4-7	→	Payload data bytes 4–7	0–4 data bytes depending on DFL

21.4.6.5 UART Mode: Transmit DMA Channel

To use DMA when the LINFlexD is operating in UART mode, the LINFlexD UART transmit/receive buffers must be configured as FIFOs.

The LINFlexD uses a single transmit DMA channel (0) to move data from RAM to the UART transmit buffer. To enable DMA for UART transmit mode, both DMATXE[0] and input signal dma_enb_req_tx[0] must be asserted.

In response to assertion of ipd_req_tx[0], the DMA controller transfers up to 4 bytes or 2 halfwords from RAM to the UART transmit FIFO through successive writes to the DATA0~DATA3 registers.

In the cycle after the data phase of the last write for the transfer, the DMA controller must assert both dma_done_lw_tx[n] and dma_ipd_complete_tx[n] to acknowledge the DMA request.

21.4.6.6 UART Mode: Receive DMA Channel

To use DMA when the LINFlexD is operating in UART mode, the LINFlexD UART transmit/receive buffers must be configured as FIFOs.

The LINFlexD uses a single receive DMA channel (0) to move data from the UART receive buffer to RAM. To enable DMA for UART receive mode, both DMATRE[0] and input signal dma_enb_req_rx[0] must be asserted.

In response to assertion of ipd_req_rx[0], the DMA controller performs successive reads from the DATA4~DATA7 register to transfer up to 4 bytes or 2 halfwords from the UART receive FIFO to RAM.

In the cycle after the data phase of the last read for the transfer, the DMA controller must assert both dma_done_lw_rx[0] and dma_ipd_complete_rx[0] to acknowledge the DMA request.

21.4.6.7 DMA Usage Notes and Limitations

- In LIN slave mode, the DMA capability can be used only if the ID filtering mode is activated. The number of ID filters enabled must be equal to the number of DMA channels enabled. The

Yuntu Microelectronics

correspondence between channel number and ID filter is based on Identifier Filter Match Index (IFMI).

- To use the DMA capability in LIN master mode, both the transmit and receive DMA channels must be enabled.
- In UART mode the DMA capability can be used only if the UART Tx/Rx buffers are configured as FIFOs.
- DMA and CPU operating modes are mutually exclusive for the data/frame transfer on a UART or LIN node. Once a DMA transfer is completed, the CPU can manage subsequent accesses.
- DMA enable input signals (dma_enb_req_rx[15:0] and dma_enb_req_rx[15:0]) may be tied high (enabling DMA for the respective channels) or low (disabling DMA for the respective channels). The DMA enable inputs do not have to be actively toggled to enable or disable DMA for the respective channels.

21.4.7 LINFlexD Clock Tolerance

21.4.7.1 Faster Receiver Tolerance

If the receiver has a higher baud rate than the transmitter, stop bit sampling may start during the last transmitted payload bit. To ensure the correct noise and framing error free reception, samples RS8, RS9, and RS10 must be located in the transmitted stop bit as shown in the figure below.

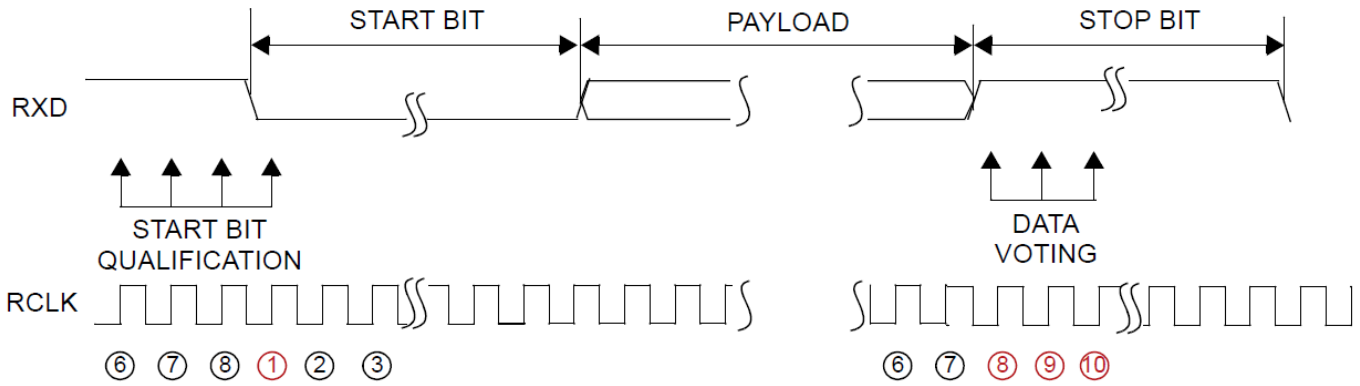


Figure 21.26: Faster Receiver

21.4.7.2 Slower Receiver Tolerance

If the receiver has a slower baud rate than the transmitter, stop bit sampling may still be running while the next start bit is transmitted. To ensure the correct noise and framing error free reception, samples RS8, RS9, and RS10 must be located in the transmitted stop bit as shown in figure below.

Yuntu Microelectronics

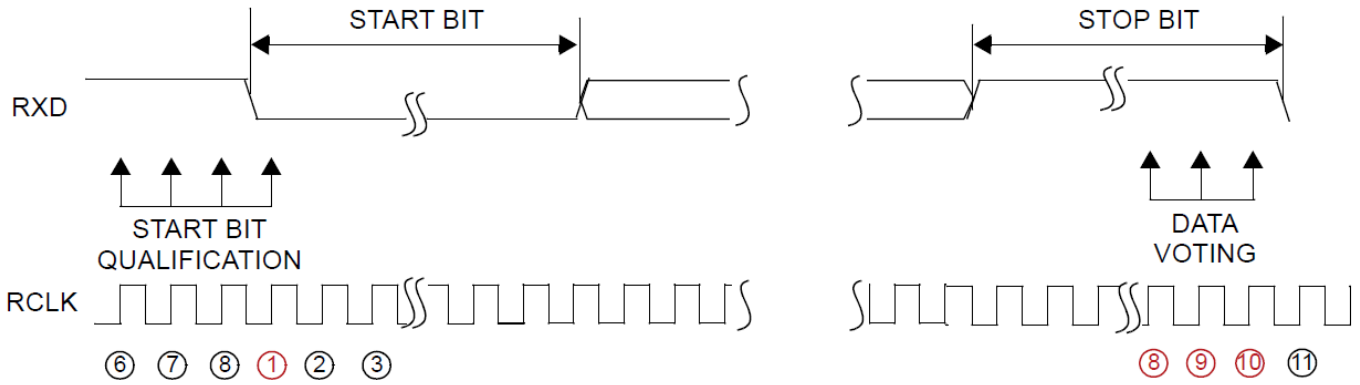


Figure 21.27: Slower Receiver

21.5 Glossary

LIN	Local Interconnect Network
UART	Universal Asynchronous Receiver/Tansmitter
DMA	Direct Memory Access
FIFO	First In First Out

Yuntu Microelectronics

Chapter 22

Serial Port Interface (SPI)

22.1 Chip-specific SPI information

22.1.1 Instantiation Information

YTM32B1HA0x contains 8 SPI instances.

The FIFO depth of SPI0 and SPI1 is 3, and the others is 2.

22.2 Introduction

22.2.1 Features

- Support clock polarity and phase configuration
- Configurable frame size
- Transmit/Receive FIFO
- Support single line mode
- Support Master and slave mode
- Support Transmit/Receive via DMA

22.2.2 Signal Description

Table 22.1: SPI Module Signal Description

Signal	Name	Description	Type
SCK	Serial Clock	• Clock output in mater mode • Clock input in slave mode	IO
SOUT	Serial data output	• Serial data output • Could be configured as data input • Used as DATA[0] in parallel data transfer	IO
SIN	Serial data input	• Serial data input • Could be configured as data output • Used as DATA[1] in parallel data transfer	IO

Yuntu Microelectronics

Signal	Name	Description	Type
PCS[7:0]	Peripheral Chip Selection	• Peripheral chip selection in master and slave mode • The PCS number of SPI0 is 8, SPI1 is 6 and the others is 4.	IO

22.3 Register Definition

22.3.1 SPI Memory Map

SPI0 base address 400a8000h
SPI1 base address 400ac000h
SPI2 base address 400b0000h
SPI3 base address 400b4000h
SPI4 base address 400b8000h
SPI5 base address 400bc000h
SPI6 base address 400c0000h
SPI7 base address 400c4000h

Table 22.2: SPI Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Transmit Configuration Register (TXCFG)	32	RW	0000001Fh
8h	Clock Configuration Register (CLK)	32	RW	00000000h
ch	Status Register (STS)	32	RW	00020002h
10h	Interrupt Enable Register (INTE)	32	RW	00000000h
14h	Data Match Register 0 (MATCH0)	32	RW	00000000h
18h	Data Match Register 1 (MATCH1)	32	RW	00000000h
1ch	TX FIFO Configuration Register (TXFIFO)	32	RW	see description
20h	RX FIFO Configuration Register (RXFIFO)	32	RW	see description
24h	Data Register (DATA)	32	RW	00000000h

Yuntu Microelectronics

22.3.1.1 SPI CTRL Register

Offset: 0h

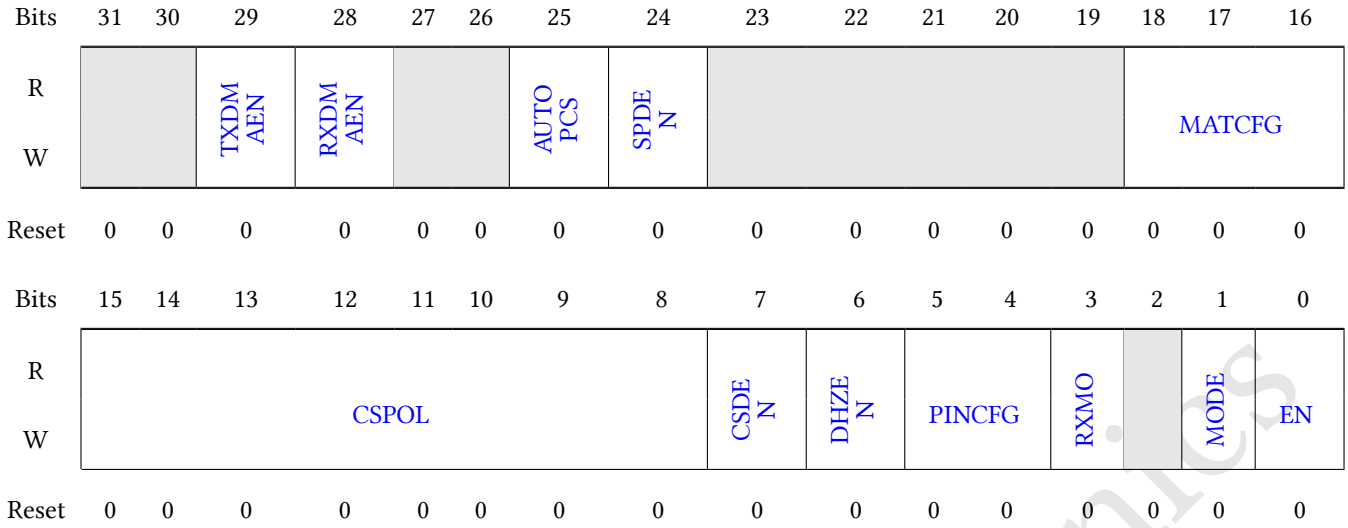


Table 22.3: SPI CTRL Register Description

Field	Function
29 TXDMAEN	**TX Data DMA Enable** TX DATA request DMA enable when TX fifo less than watermark or buff empty 0b - DMA request disabled 1b - DMA request enabled
28 RXDMAEN	**RX Data DMA Enable** RX DATA request DMA enable when RX fifo great than watermark or buff full 0b - DMA request disabled 1b - DMA request enabled
25 AUTOPCS	**Automatic PCS** When SPI in slave mode, setting this bit will cause SPI slave to automatically generate an internal PCS signal when TXCFG[CPHA]=1 0b - SPI slave use PCS pin directly 1b - SPI slave will regenerate internal PCS signal
24 SPDEN	**Sample Point Delay Enable** Setting this bit will enable sample delay logic before data input under master mode. 0b - SPI sample data on SCK edge 1b - SPI sample data on delayed SCK edge
18 - 16	**Match Configuration**

Yuntu Microelectronics

Field	Function
MATCFG	SPI match feature configuration 000b - Match disabled 010b - First data match MATCH0 or MATCH1 011b - Any data match MATCH0 or MATCH1 100b - First data match MATCH0 and second data match MATCH1 101b - Any data match MATCH0 and next data match MATCH1 110b - First data bit AND with MATCH1 match MATCH0 AND MATCH1 111b - Any data bit AND with MATCH1 match MATCH0 AND MATCH1
15 - 8 CSPOL	**SPI Pin Selection Polarity Selection** Pin selection polarity selection of pcs pin. 00000000b (0x00) - set PCS[n] to active low 00000001b (0x01) - set PCS[n] to active high
7 CSDEN	**SPI PCS[3:2] Pin as Data Line Enable** Setting this bit will change SPI PCS[3:2] pin as data line to support 4 data lines transfer 0b - PCS pin works as normal chip selection function 1b - SPI works in half_duplex 4 data lines mode and PCS[3:2] is DATA[3:2]
6 DHZEN	**SPI Data Highz After Transfer Enable** Setting this bit will set output data line to high z when chip select is negated 0b - SPI data output line keeps at last value when chip select is negated 1b - SPI data output line in high z mode when chip select is negated
5 - 4 PINCFG	**Pin Configuration** Control the data input source and data output target 00b - Data input from SIN and data output to SOUT 01b - Data input from SIN and data output to SIN, spi works under half-duplex serial mode 10b - Data input from SOUT and data output to SOUT, spi works under half-duplex serial mode 11b - Data input from SOUT and data output to SIN
3 RXMO	**Receive Data Match Only** Setting this bit will set SPI only stores data after received matched match configuration 0b - SPI will store all received data 1b - SPI will only store data after data matched
1 MODE	**SPI Mode Selection** SPI Master/slave mode selection 0b - SPI slave mode 1b - SPI master mode
0	**SPI Module Enable**

Yuntu Microelectronics

Field	Function
EN	Write to this field will enable/disable SPI module Read this field will reflect current SPI module state **NOTE**. This bit only update when SPI logic is enabled or disabled successfully 0b - SPI module is disabled 1b - SPI module is enabled

22.3.1.2 SPI TXCFG Register

Any data written to TXCFG will be push to TX FIFO, TXCFG will only support 32 bit access, and the configuration in TXCFG will only affect following Data transmissions.

NOTE. TXCFG should only update when SPI is IDLE

NOTE. Read TXCFG should read twice check or read when SPI TX FIFO empty

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CPOL	CPHA	PRESCALE			PCS			LSBF	BSW	CONT	CONTC	MSKRX	MSKTX	WIDTH	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W					FRAMESZ											
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table 22.4: SPI TXCFG Register Description

Field	Function
31 CPOL	**Clock Polarity** Clock polarity of following data transfer 0b - Clock idles low, leading edge is rising, trailing edge is falling 1b - Clock idles high, leading edge is falling, trailing edge is rising
30 CPHA	**Clock Phase** Clock phase fo following data transfer 0b - Data captured on clock leading edge, Data changed on clock trailing edge 1b - Data captured on clock trailing edge, Data changed on clock leading edge

Yuntu Microelectronics

Field	Function
29 - 27 PRESCALE	**Prescaler Value** Clock prescaler of following SPI clock 000b - SPI clock divided by 1 001b - SPI clock divided by 2 010b - SPI clock divided by 4 011b - SPI clock divided by 8 100b - SPI clock divided by 16 101b - SPI clock divided by 32 110b - SPI clock divided by 64 111b - SPI clock divided by 128
26 - 24 PCS	**Peripheral Chip Selection** Changing this field will change following peripheral selection 000b - Peripheral 0 selected 001b - Peripheral 1 selected 010b - Peripheral 2 selected 011b - Peripheral 3 selected 100b - Peripheral 4 selected 101b - Peripheral 5 selected 110b - Peripheral 6 selected 111b - Peripheral 7 selected
23 LSBF	**LSB First Mode** Setting this bit will change following data transfer shift order 0b - MSB is shifted first in data transfer 1b - LSB is shifted first in data transfer
22 BSW	**Byte Swap Mode** Setting this bit will change 32-bit data byte order, this bit controls both TX and RX FIFO 0b - Byte swap mode not enabled 1b - Byte[3] swap with Byte[0], Byte[2] swap with Byte[1]
21 CONT	**Continuous Transfer** In master mode, continuous transfer will keep PCS selected, not space will be inserted between data frame. 0b - Continuous transfer mode is disabled 1b - Continuous transfer mode is enabled
20 CONTC	**Continuous Command** Setting this bit will allow insert command during continuous transfer 0b - Continuous command mode is disabled 1b - Continuous command mode is enabled
19	**Mask RX Data**

Yuntu Microelectronics

Field	Function
MSKRX	Setting this bit will disable write received data to RX FIFO 0b - Received data will write to RX FIFO 1b - Received data won't write to RX FIFO
18 MSKTX	**Mask TX Data** In master mode, setting this bit will SPI start a new transfer, SPI data out will be tristated This bit is used to receive data from slave in half-duplex mode. **NOTE**. This bit is cleared by hardware when data transfer ends. 0b - Normal data transfer 1b - Mask TX data and start a data transfer
17 - 16 WIDTH	**Data Width Control** Changing this field allows SPI working in 2 line or 4 line mode **NOTE**. MSKTX or MSKRX must be set to let SPI working in half_duplex mode 00b - 1 data line, normal mode 01b - 2 data line [SDI, SDO], half-duplex mode 10b - 4 data line [PCS[3:2], SDI, SDO], half-duplex mode
11 - 0 FRAMESZ	**Frame Size Control** One frame include FRAMESZ + 1 bits

22.3.1.3 SPI CLK Register

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SCKPCS								PCSSCK							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FMDLY								DIV							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22.5: SPI CLK Register Description

Field	Function
31 - 24	**SCK to PCS Delay**

Yuntu Microelectronics

Field	Function
SCKPCS	Control the delay from last SCK to the PCS negation, only valid in SPI master mode The delay is SCKPCS + 1 divided SPI clock divided by TXCFG[PRESCALE] field
23 - 16 PCSSCK	**PCS to SCK Delay** CS to SCK delay
15 - 8 FMDLY	**Frame Delay** Control SCK delay between frames in continuous mode or the time of PCS negation, only valid in master mode • Delay between frames in continuous mode is FMDLY + 1 divided SPI clock divided by TXCFG[PRESCALE] field • PCS negation time is FMDLY + 1 divided SPI clock divided by TXCFG[PRESCALE] field
7 - 0 DIV	**SPI Clock Divider** SPI clock period is DIV + 2 divided SPI clock divided by TXCFG[PRESCALE] field

22.3.1.4 SPI STS Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R														BUSY	RXEPT	SOF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									MATIF	RXOVIF	TXUNIF	TCIF	FCIF		TXIF	RXIF
W									W1C	W1C	W1C	W1C	W1C			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Table 22.6: SPI STS Register Description

Field	Function
18	**SPI Module Busy**

Field	Function
BUSY	SPI module is transmitting or receiving data 0b - SPI module is idle 1b - SPI module is busy
17 RXEPT	**SPI RX FIFO Empty Flag** 0b - SPI RX FIFO is not empty 1b - SPI RX FIFO is empty
16 SOF	**Start of Frame** Indicates SPI received first data frame 0b - Received data after first frame or idle 1b - Received first frame data
7 MATIF	**Data Match Interrupt Flag** Indicates SPI received data matched data match configuration 0b - Received data not matched 1b - Received data matched
6 RXOVIF	**RX FIFO Overflow Interrupt Flag** Indicates SPI RX FIFO is overflowed 0b - RX FIFO is not overflowed 1b - RX FIFO is overflowed
5 TXUNIF	**TX FIFO Underflow Interrupt Flag** Indicates SPI TX FIFO is underflow 0b - TX FIFO is not underflow 1b - TX FIFO is underflow
4 TCIF	**Transmit Complete Interrupt Flag** Indicates SPI has finished data transfer and return to idle 0b - All data transfer not completed 1b - All data transfer completed
3 FCIF	**Frame Transmit Complete Interrupt Flag** Indicates SPI has finished one frame data transfer 0b - One frame data transfer not completed 1b - One frame data transfer completed
1 TXIF	**TX Request Data Flag** Indicates words in TX FIFO is equal or less than TX FIFO watermark setting 0b - Word in TX FIFO is greater than watermark 1b - Word in TX FIFO is equal or less than watermark, request to write new data
0	**RX Request Read Data Flag**

Yuntu Microelectronics

Field	Function
RXIF	Indicates words in RX FIFO is greater than RX FIFO watermark setting 0b - Word in RX FIFO is equal or less than watermark 1b - Word in RX FIFO is greater than watermark, should read data register

22.3.1.5 SPI INTE Register

Offset: 10h

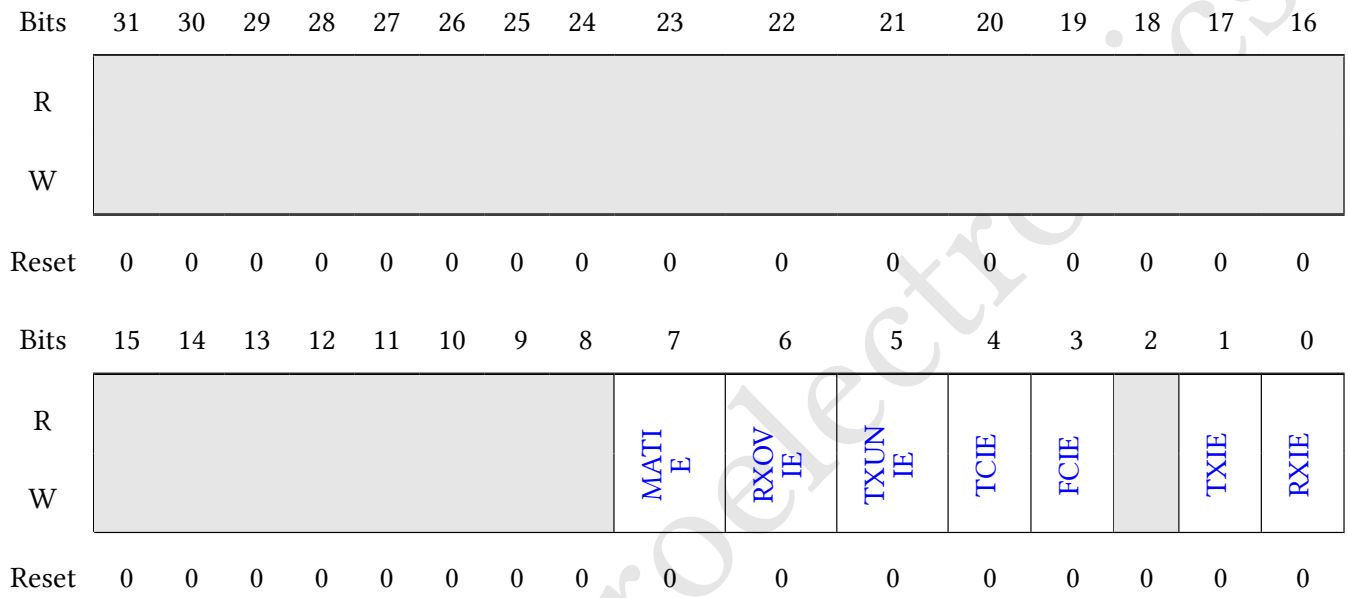


Table 22.7: SPI INTE Register Description

Field	Function
7 MATIE	**Data Match Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
6 RXOVIE	**RX FIFO Overflow Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
5 TXUNIE	**TX FIFO Underflow Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
4 TCIE	**SPI TX Complete Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
3	**SPI Frame Complete Interrupt Enable**

Yuntu Microelectronics

Field	Function
FCIE	0b - Interrupt disabled 1b - Interrupt enabled
1 TXIE	**TX DATA Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
0 RXIE	**RX DATA Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled

22.3.1.6 SPI MATCH0 Register

Offset: 14h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22.8: SPI MATCH0 Register Description

Field	Function
31 - 0 MATCH0	**SPI Data Match Register 0**

22.3.1.7 SPI MATCH1 Register

Offset: 18h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MATCH1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22.9: SPI MATCH1 Register Description

Field	Function
31 - 0 MATCH1	**SPI Data Match Register 1**

22.3.1.8 SPI TXFIFO Register

The reset values of SPI0 and SPI1 are 0x03000000, and that of others are 0x02000000.

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R					SIZE											
W																
Reset	0	0	0	0	u	u	u	u	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					COUNT								WATER			
W	RESET															
Reset	u	0	0	0	u	u	u	u	0	0	0	0	u	u	u	u

Table 22.10: SPI TXFIFO Register Description

Field	Function
27 - 24 SIZE	**SPI TX FIFO Size** Available TX FIFO Size is 2^SIZE words
15 RESET	**SPI TX FIFO Reset** Reset TX FIFO address pointer

Yuntu Microelectronics

Field	Function
11 - 8 COUNT	**Word Count in TX FIFO** SPI TX FIFO current data count
3 - 0 WATER	**TX FIFO Watermark** TXIF set when words count in TX FIFO equal or less than watermark

22.3.1.9 SPI RXFIFO Register

The reset values of SPI0 and SPI1 are 0x03000000, and that of others are 0x02000000.

Offset: 20h

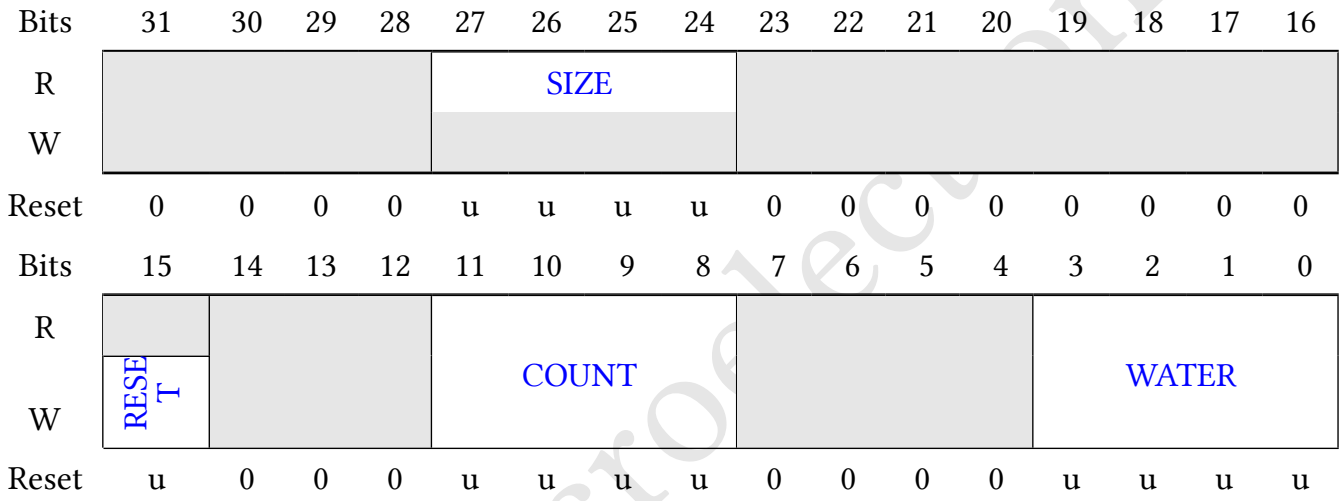


Table 22.11: SPI RXFIFO Register Description

Field	Function
27 - 24 SIZE	**SPI RX FIFO Size** Available RX FIFO Size is 2^SIZE words
15 RESET	**SPI RX FIFO Reset** Reset SPI RX FIFO address pointer
11 - 8 COUNT	**Word Count in TX FIFO** SPI RX FIFO current data count
3 - 0 WATER	**RX FIFO Watermark** RXIF set when words count in RX FIFO greater than watermark

22.3.1.10 SPI DATA Register

Offset: 24h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 22.12: SPI DATA Register Description

Field	Function
31 - 0 DATA	**RX TX Data Register** Read DATA will return SPI received data in RX FIFO Write DATA will push new data into TX FIFO **NOTE**. This register could only access in word

22.4 Functional description

The module supports full-duplex synchronous serial communications between chips. The SPI module can operate as master or slave. As master, the SPI initiates and controls all data transactions. The SPI has Tx/CMD FIFO and Rx FIFO for data and command buffering.

22.4.1 SPI Functional Clock

There are a few of clock sources for the SPI functional clock, and it can be configured by IPC. It is asynchronous to the bus clock. Whether SPI is able to work in the low power modes depends on whether the SPI functional clock remains enabled. The clock used for timing control in the register CLK is the SPI functional clock.

22.4.2 Master Mode

22.4.2.1 Tx and CMD FIFO

In master mode, the module initiates the serial transfers by controlling the SCK and PCS signals, the Tx FIFO stores the Tx data and command words. A command word should be written between frames and always takes control in the next frame.

22.4.2.2 Rx FIFO

The Rx FIFO stores the received data during SPI bus transfer. The received data is stored in the least significant bits of a 32-bit word and then is written to the Rx FIFO at the end of the frame. During a continuous transfer, the received data is only written to the Rx FIFO when the Tx FIFO is not empty or the Transmit Configuration Register(TXCFG) is written to end the frame. When the TXCFG[MSKRX] bit(Mask RX Data) is set, the received data is discarded instead of being stored in the Rx FIFO.

Yuntu Microelectronics

22.4.2.3 Data Match

Receive data supports a received data match function.

- Received data that is discarded due to the Mask RX Data TXCFG[MSKRX] bit cannot cause the data match, and will delay the received data match on the first received data word until all discarded data is received.
- The receive data match function can also be configured to discard all received data until a data match is detected, when the Receive Data Match Only CTRL[RXMO] bit is set.
- After a receive data match, to allow all subsequent data to be received, should clear the Receive Data Match Only CTRL[RXMO] bit and the DATA Match Interrupt Flag STS[MATIF] bit first.

22.4.2.4 Pin Configuration

- To swap direction or support half-duplex transfers on the same pin, the SPI_SIN and SPI_SOUT pins can be configured using the Pin Configuration CTRL[PINCFG].
- When performing quad-data transfers, the Chip Selection Polarity CTRL[CSPOL] must be enabled. The Peripheral Chip Selection Polarity CTRL[CSPOL] is also used to disable SPI_PCS[3:2] functions.

22.4.2.5 Clock LoopBack

The SPI master can use the SPI_SCK or the delayed version of the SPI_SCK to sample the input data. The delayed version of the SPI_SCK is delayed by the SPI_SCK pin output delay plus the SPI_SCK pin input delay, and is configured by setting CTRL[SPDEN].

22.4.2.6 Continuous Transfer

When TXCFG[CONT] bit is set, the PCS will keep asserted at the end of the frame size transfer and the SPI will start a new frame size transfer.

22.4.2.6.1 Continuous Command

When TXCFG[CONTC] bit is set, it allows to change the command word during a continuous transfer. When TXCFG[CONTC] bit is cleared, the new command word will terminate the current frame at the the end of the existing word.

22.4.3 Slave Mode

In SPI slave mode, the module responds to transfers initiated by SPI bus master. SPI slave mode uses the same shift register and logic that master mode uses, but does not use the Clock Configuration Register(CLK). During the SPI transfer, the Transmit Configuration Register (TXCFG) should be unchanging.

22.4.3.1 Tx and CMD FIFO

Before enabling SPI slave mode, the Transmit Configuration Register (TXCFG) should be initialized. If SPI is enabled, the Transmit Configuration Register(TXCFG) should only be updated when SPI is idle. Before the SPI_PCS input asserts, the Tx FIFO must be filled with transmit data, or the Tx FIFO Underflow Interrupt Flag STS[TXUNIF] will set.

Yuntu Microelectronics

22.4.3.2 Receive FIFO

The Rx FIFO is used to store received data during SPI bus transfers. When the Mask RX Data TXCFG[MSKRX] is set, the receive data is discarded instead of being stored in the receive FIFO.

22.4.3.3 Data Match

Receive data supports a receive data match function. - Received data that is discarded due to the Mask RX Data TXCFG[MSKRX] bit cannot cause the data match, and will delay the receive data match on the first received data word until all discarded data is received. - The receive data match function can also be configured to discard all received data until a data match is detected, when the Receive Data Match Only CTRL[RXMO] bit is set. - After a receive data match, to allow all subsequent data to be received, should clear the Receive Data Match Only CTRL[RXMO] bit and the DATA Match Interrupt Flag STS[MATIF] bit first.

22.4.3.4 Internal PCS

When SPI in slave mode, SPI support to generate an internal PCS signal. It requires: - The Clock Phase TXCFG[CPHA] bit is set - The Clock Polity TXCFG[CPOL] bit is set - The Regenerate PCS Enable CTRL[REPCS] bit is set.

22.4.3.5 Continuous Transfer

When TXCFG[CONT] bit is set, the SPI will transmits the received data after the first frame size bits are transmitted until the PCS is inactive.

22.4.3.5.1 Continuous Mode

When TXCFG[CONTC] bit is set, the internal configuration RXMSK and TXMSK of the command word will take effect after the first frame size bits are transmitted until the PCS is inactive.

22.5 Glossary

PCS	Peripheral Chip Select
SCK	Serial Clock
SDI	Slave Data In
SDO	Slave Data Out

Yuntu Microelectronics

Chapter 23

Quad Serial Peripheral Interface (QSPI)

23.1 Chip-specific Information

23.1.1 Instantiation Information

YTM32B1HA0x contains 1 QSPI module.

23.1.2 Signal Description

Table 23.1: QSPI Signal Table

Signal Name	Description	Direction
PCSFA	Peripheral Chip Select Flash A	O
SCKFA	Serial Clock Flash A	O
IOFA[7:0]	Input/output data Flash A	IO
DQSA	Data Strobe Flash A	IO
PCSFB	Peripheral Chip Select Flash B	O
SCKFB	Serial Clock Flash B	O
IOFB[7:0]	Input/output data Flash B	IO
DQSB	Data Strobe Flash B	IO

23.1.3 Clock Description

Table 23.2: QSPI Clock Table

Clock Name	Description
AHB clock	AHB bus clock used for accessing SRAM
APB Clock	APB bus clock used for accessing registers
Functional Clock	Functional Clock used for communication with serial flash devices

23.1.4 DMA Description

Yuntu Microelectronics

Table 23.3: QSPI DMA Table

No.	Request Source	Description
106	RxFIFO drain request	RxFIFO watermark valid will generate request
107	TxFIFO fill request	TxFIFO watermark valid will generate request

23.1.5 Interrupt Description

Table 23.4: QSPI Interrupt Table

Vector	IRQn	Request Source	Description
67	51	All interrupts ORed	Error detected, FIFO status changed and command transfer finished

23.2 Introduction

23.2.1 Overview

The QSPI module acts as a communication interface to the external serial flash devices.

23.2.2 Features

- Flexible programmable sequence engine to support various vendor devices
  - Serial NOR Flash
  - Serial NAND Flash
  - HyperBus device (HyperFlash)
- Support max to 4 LUT sequences, each LUT sequence contains 10 instructions
- Support single, dual, quad, and octal access modes for serial flash memories
- Support two sets of independent flash interfaces
- Support individual/parallel mode
- Support for XIP (Execute in Place)
- Support for SDR (100MHz) and DDR (50MHz) mode
- Support multi-master accesses
- Support delay chain for input clock sampling data
- 32*32 TxFIFO for data writing with DMA support
- 32*32 RxFIFO for data reading with DMA support
- 64*32 buffer for AHB access
- Support 128MB AHB access size (0x68000000~0x6FFFFFFF)
- Support serial flash clock stopped when RxFIFO is full

23.2.3 Block Diagram

The following figure shows the QSPI block diagram.

Yuntu Microelectronics

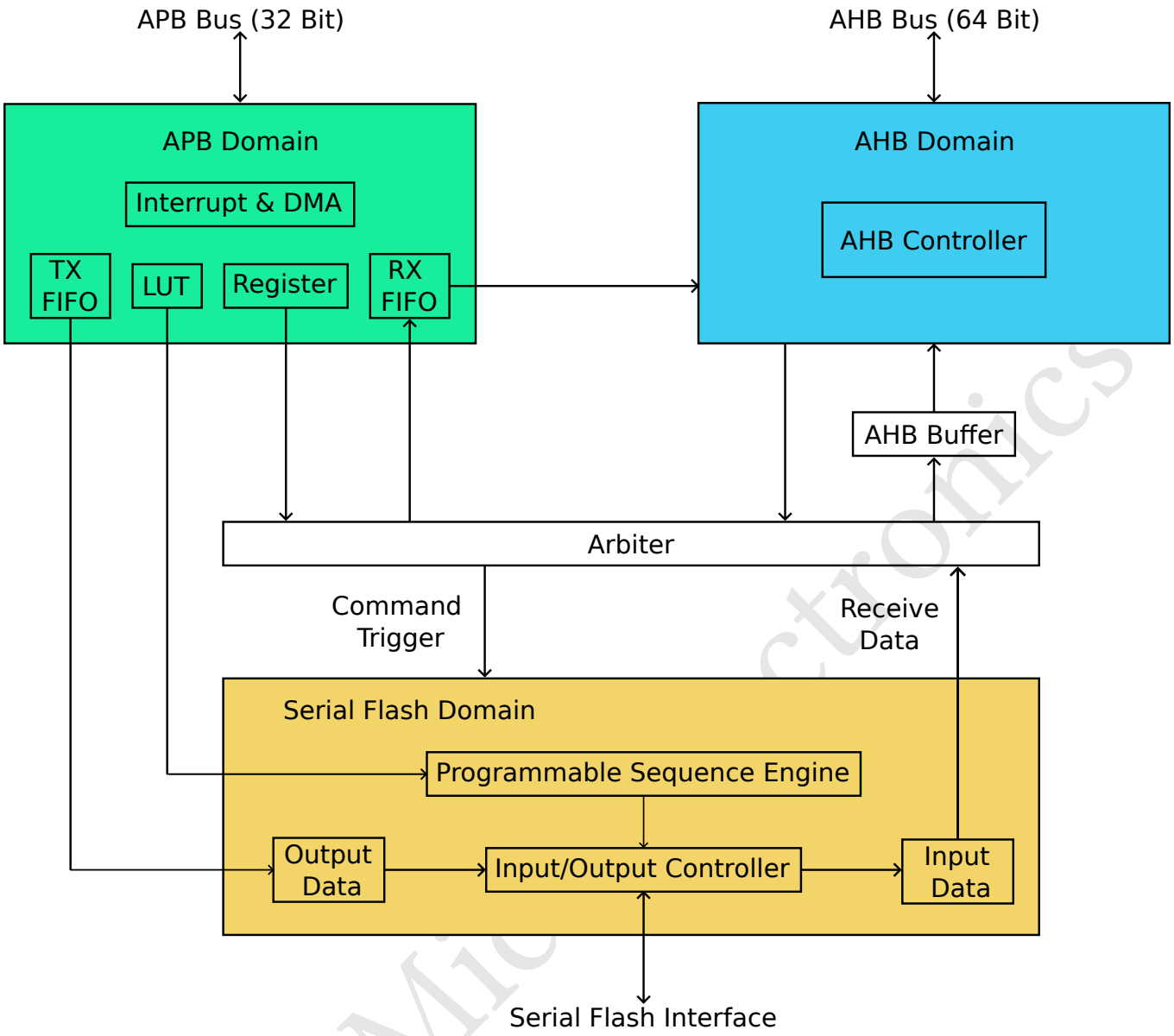


Figure 23.1: Block Diagram of QSPI

23.3 Register Definition

Registers of this module allow 32/16/8-bit access.

The values of reserved register bits are 0 when reading, write will have no influence.

All registers that are reset by IPC[SWR]. Accessing a reserved address will result in a bus error.

Register_address = Base_address + Address_offset
For example: as RXDAT1
Register_address = 0x401a0000 + 0x84

The register bits can be reset by different reset sources.

- If no specific description about register reset, the register bit is reset by any reset, including Power-on Reset, System Reset and IPC software reset if IPC module exists in the document.
- If the register bit is described as “reset by System Reset”, the register bit can be reset by System Reset and Power-on Reset, and cannot be reset by IPC[SWR].

- If the register bit is described as “reset by Power-on Reset”, the register bit can be reset by Power-on Reset only, and cannot be reset by System Reset and IPC[SWR].
- If the register bit is described as “not able to be reset”, the register bit can be reset by any reset, which means those register bits have no reset value after any reset.

23.3.1 QSPI Memory Map

QSPI0 base address 401a0000h

Table 23.5: QSPI Memory Map

Offset	Register	Width	Access	Reset Value
0h	Module Enable Register (MEN)	32	RW	00000000h
4h	Module Configuration Register (MCFG)	32	RW	0F00000Ch
8h	Sample Register (SMP)	32	RW	00009F00h
ch	Status Register (ST)	32	RW	00207000h
10h	Interrupt Flag Register (IF)	32	RW	00800000h
14h	Interrupt Enable Register (IE)	32	RW	00000000h
18h	Suspend Status Register (SPNDST)	32	RO	00000000h
1ch	RxFIFO Control and Status Register (RBCS)	32	RW	00000000h
20h	TxFIFO Control and Status Register (TBCS)	32	RW	00000000h
24h	APB access Configuration Register (APBCFG)	32	RW	00000000h
28h	AHB access Configuration Register (AHBCFG)	32	RW	00002000h
2ch	Buffer 0 Configuration Register (BUF0CFG)	32	RW	00000003h
30h	Buffer 1 Configuration Register (BUF1CFG)	32	RW	00000002h
34h	Buffer 2 Configuration Register (BUF2CFG)	32	RW	00000001h
38h	Buffer 3 Configuration Register (BUF3CFG)	32	RW	80000000h
3ch	Buffer 0 Index Register (BUF0IND)	32	RW	00000000h
40h	Buffer 1 Index Register (BUF1IND)	32	RW	00000000h
44h	Buffer 2 Index Register (BUF2IND)	32	RW	00000000h
48h	Flash Access Configuration Register (FACFG)	32	RW	00000303h
4ch	Serial Flash Address Register (SFA)	32	RW	00000000h
50h	Serial Flash Address Configuration Register (SFACFG)	32	RW	00000000h
54h	Serial Flash A Top Address Register (SFATA)	32	RW	6C000000h
58h	(SFA1TA)	32	RW	6C000000h
5ch	Serial Flash B Top Address Register (SFBTA)	32	RW	70000000h
60h	(SFB1TA)	32	RW	70000000h
64h	Data Test Pattern Register (DTPR)	32	RW	5A691E78h
7ch	TxFIFO Data Register (TXDAT)	32	RW	00000000h
80h	RxFIFO Data Register (RXDAT0)	32	RW	00000000h
84h	RxFIFO Data Register (RXDAT1)	32	RW	00000000h
88h	RxFIFO Data Register (RXDAT2)	32	RW	00000000h
8ch	RxFIFO Data Register (RXDAT3)	32	RW	00000000h
90h	RxFIFO Data Register (RXDAT4)	32	RW	00000000h
94h	RxFIFO Data Register (RXDAT5)	32	RW	00000000h
98h	RxFIFO Data Register (RXDAT6)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
9ch	RxFIFO Data Register (RXDAT7)	32	RW	00000000h
a0h	RxFIFO Data Register (RXDAT8)	32	RW	00000000h
a4h	RxFIFO Data Register (RXDAT9)	32	RW	00000000h
a8h	RxFIFO Data Register (RXDAT10)	32	RW	00000000h
ach	RxFIFO Data Register (RXDAT11)	32	RW	00000000h
b0h	RxFIFO Data Register (RXDAT12)	32	RW	00000000h
b4h	RxFIFO Data Register (RXDAT13)	32	RW	00000000h
b8h	RxFIFO Data Register (RXDAT14)	32	RW	00000000h
bch	RxFIFO Data Register (RXDAT15)	32	RW	00000000h
c0h	RxFIFO Data Register (RXDAT16)	32	RW	00000000h
c4h	RxFIFO Data Register (RXDAT17)	32	RW	00000000h
c8h	RxFIFO Data Register (RXDAT18)	32	RW	00000000h
cch	RxFIFO Data Register (RXDAT19)	32	RW	00000000h
d0h	RxFIFO Data Register (RXDAT20)	32	RW	00000000h
d4h	RxFIFO Data Register (RXDAT21)	32	RW	00000000h
d8h	RxFIFO Data Register (RXDAT22)	32	RW	00000000h
dch	RxFIFO Data Register (RXDAT23)	32	RW	00000000h
e0h	RxFIFO Data Register (RXDAT24)	32	RW	00000000h
e4h	RxFIFO Data Register (RXDAT25)	32	RW	00000000h
e8h	RxFIFO Data Register (RXDAT26)	32	RW	00000000h
ech	RxFIFO Data Register (RXDAT27)	32	RW	00000000h
f0h	RxFIFO Data Register (RXDAT28)	32	RW	00000000h
f4h	RxFIFO Data Register (RXDAT29)	32	RW	00000000h
f8h	RxFIFO Data Register (RXDAT30)	32	RW	00000000h
fch	RxFIFO Data Register (RXDAT31)	32	RW	00000000h
100h	LUT Key Register (LUTKEY)	32	RW	5AF05AF0h
104h	Lock Register (LOCK)	32	RW	00000002h
108h	Look Up Table Register (LUT0)	32	RW	see description
10ch	Look Up Table Register (LUT1)	32	RW	see description
110h	Look Up Table Register (LUT2)	32	RW	see description
114h	Look Up Table Register (LUT3)	32	RW	see description
118h	Look Up Table Register (LUT4)	32	RW	see description
11ch	Look Up Table Register (LUT5)	32	RW	see description
120h	Look Up Table Register (LUT6)	32	RW	see description
124h	Look Up Table Register (LUT7)	32	RW	see description
128h	Look Up Table Register (LUT8)	32	RW	see description
12ch	Look Up Table Register (LUT9)	32	RW	see description
130h	Look Up Table Register (LUT10)	32	RW	see description
134h	Look Up Table Register (LUT11)	32	RW	see description
138h	Look Up Table Register (LUT12)	32	RW	see description
13ch	Look Up Table Register (LUT13)	32	RW	see description
140h	Look Up Table Register (LUT14)	32	RW	see description
144h	Look Up Table Register (LUT15)	32	RW	see description

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
148h	Look Up Table Register (LUT16)	32	RW	see description
14ch	Look Up Table Register (LUT17)	32	RW	see description
150h	Look Up Table Register (LUT18)	32	RW	see description
154h	Look Up Table Register (LUT19)	32	RW	see description

23.3.1.1 QSPI MEN Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W																EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23.6: QSPI MEN Register Description

Field	Function
0 EN	**Enable** Module enable/disable control. 0b - Module is disabled or disabled after current operation done. 1b - Module is enabled.

23.3.1.2 QSPI MCFG Register

Offset: 4h

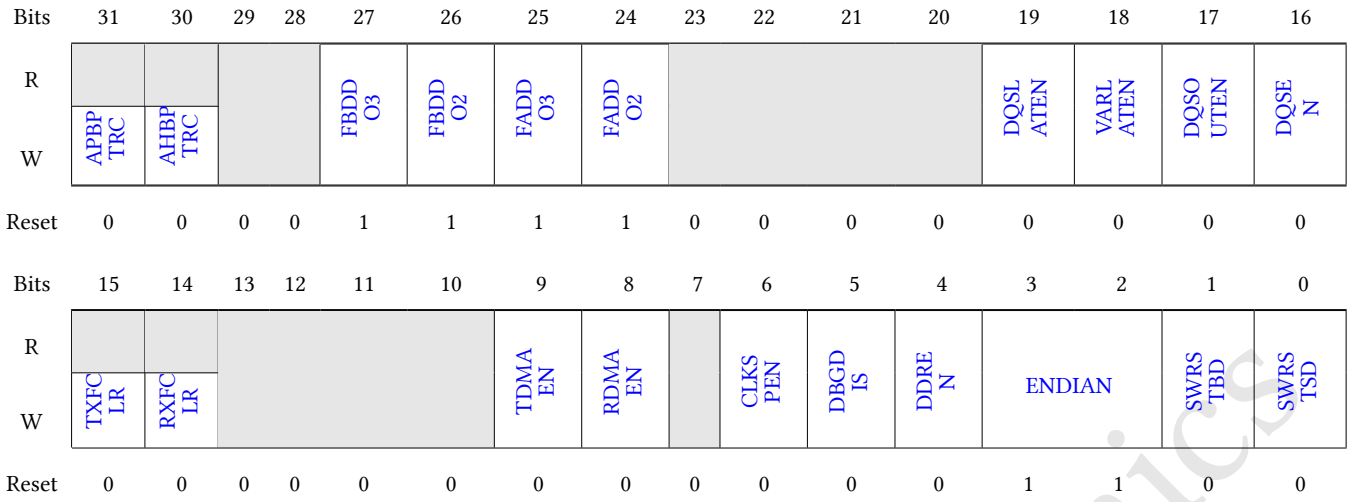


Table 23.7: QSPI MCFG Register Description

Field	Function
31 APBPTRC	**APB Pointer Clear** Write 1 to clear LUT sequence point for APB access. 0b - No action. 1b - LUT sequence point is reset to 0.
30 AHBPTRC	**AHB Pointer Clear** Write 1 to clear LUT sequence point for AHB access. 0b - No action. 1b - LUT sequence point is reset to 0.
27 FBDDO3	**Flash B Default Data Output 3** Define the data 3 output default value of flash B. It's effective in 1/2 line mode. This bit can only be changed when MEN[EN]=0. 0b - Data 3 output low in 1/2 line mode. 1b - Data 3 output high in 1/2 line mode.
26 FBDDO2	**Flash B Default Data Output 2** Define the data 2 output default value of flash B. It's effective in 1/2 line mode. This bit can only be changed when MEN[EN]=0. 0b - Data 2 output low in 1/2 line mode. 1b - Data 2 output high in 1/2 line mode.
25 FADDO3	**Flash A Default Data Output 3** Define the data 3 output default value of flash A. It's effective in 1/2 line mode. This bit can only be changed when MEN[EN]=0. 0b - Data 3 output low in 1/2 line mode. 1b - Data 3 output high in 1/2 line mode.
24	**Flash A Default Data Output 2**

Yuntu Microelectronics

Field	Function
FADDO2	Define the data 2 output default value of flash A. It's effective in 1/2 line mode. This bit can only be changed when MEN[EN]=0. 0b - Data 2 output low in 1/2 line mode. 1b - Data 2 output high in 1/2 line mode.
19 DQSLATEN	**DQS Latency Enable** Select sample data store way in DQS sample mode. 0b - No latency store, sample data stored each time is not greater than 64 bits, it's up to configured data size. 1b - Latency store, sample data stored each time is every 64 bits.
18 VARLATEN	**Variable Latency Enable** Configure this bit to adapt additional latency insertion. Only used in HyperRAM device. 0b - Variable latency is disabled. 1b - Variable latency is enabled.
17 DQSOUTEN	**DQS Output Enable** Enable/disable DQS output during write operation. DQS can only output low. 0b - DQS output is disabled. 1b - DQS output is enabled.
16 DQSEN	**DQS Enable** Enable/disable DQS write/read mode. Only used in HyperFlash. 0b - DQS write/read is disabled. 1b - DQS write/read is enabled.
15 TXFCLR	**TxFIFO Clear** Write 1 to clear TxFIFO data. 0b - No action. 1b - Clear TxFIFO data.
14 RXFCLR	**RxFIFO Clear** Write 1 to clear RxFIFO data. 0b - No action. 1b - Clear RxFIFO data.
9 TDMAEN	**Transmit DMA Enable** Enable/disable transmit DMA. 0b - Transmit DMA is disabled. 1b - Transmit DMA is enabled.
8 RDMAEN	**Receive DMA Enable** Enable/disable receive DMA. 0b - Receive DMA is disabled. 1b - Receive DMA is enabled.
6	**Clock Stop Enable**

Yuntu Microelectronics

Field	Function
CLKSPEN	Enable/disable clock output stop when RxFIFO full. 0b - Clock output stop is disabled. 1b - Clock output stop is enabled.
5 DBGDIS	**Debug Disable** Module disable control under debug mode. 0b - No action when the chip enters debug mode. 1b - Disable the module after current operation done when the chip enters debug mode.
4 DDREN	**DDR Enable** DDR mode write/read enable. 0b - DDR mode is disabled. 1b - DDR mode is enabled.
3 - 2 ENDIAN	**Endian** Define endian format of reading data. 00b - No swapping. 01b - Byte swapping in 32 bit. 10b - Word swapping in 64 bit. 11b - Word swapping in 64 bit and byte swapping in 32 bit.
1 SWRSTBD	**Software Reset Bus Domain** Reset all bus domain logics. Clear by manual. 0b - Stop reset action. 1b - Start reset action.
0 SWRSTSD	**Software Reset Serial Domain** Reset all serial flash domain logics. Clear by manual. 0b - Stop reset action. 1b - Start reset action.

Yuntu Microelectronics

23.3.1.3 QSPI SMP Register

This register can only be changed when MEN[EN]=0.

Offset: 8h

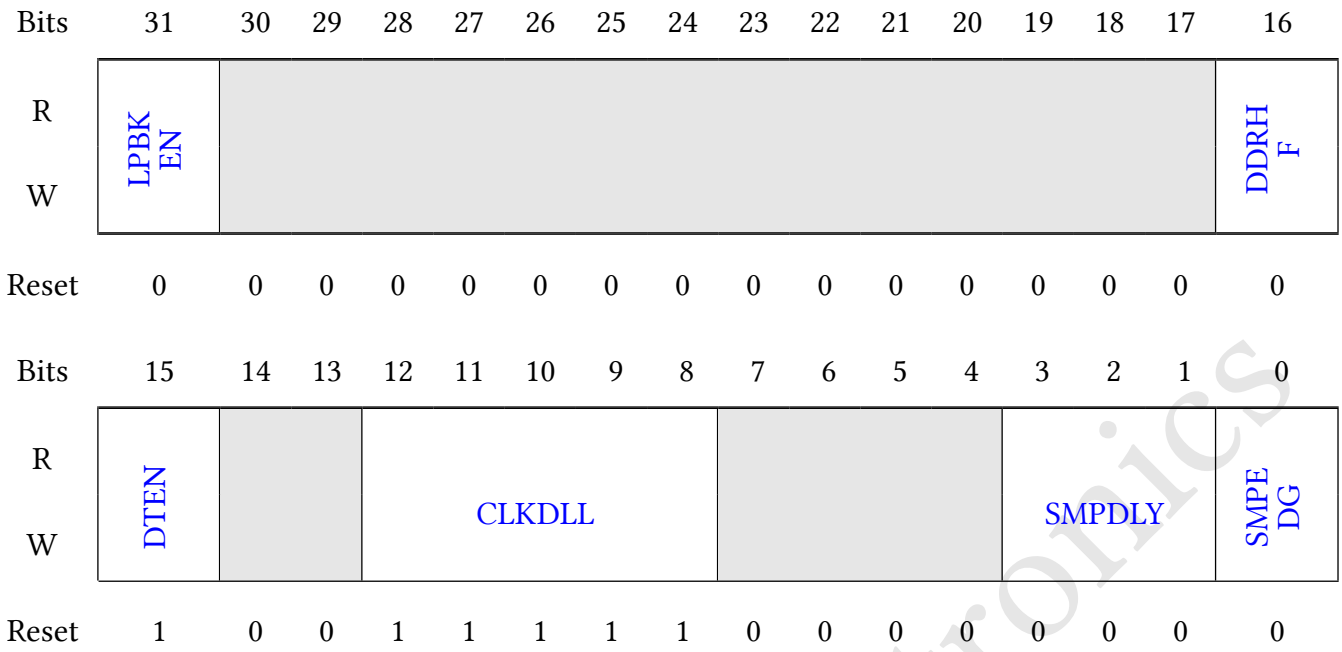


Table 23.8: QSPI SMP Register Description

Field	Function
31 LPBKEN	**Loopback Enable** Enable/disable loopback sample. 0b - Loopback sample is disabled. 1b - Loopback sample is enabled.
16 DDRHF	**DDR Half** Half cycle sample offset in DDR mode. This field only works when SMP[DTEN]=0. 0b - Sampling without offset, the first sample edge is rising edge in DDR mode. 1b - Sampling with half cycle offset, the first sample edge is falling edge in DDR mode.
15 DTEN	**Data Test Enable** Enable/disable data test mode. 0b - Data test is disabled, sampling data with configuration of registers. 1b - Data test is enabled, sampling data with the result of data test.
12 - 8 CLKDLL	**Clock Delay Length** Configure the delay cell number in loopback clock or DQS clock input path. Each delay cell is about 180~540ps. Delay cell number is the field value, value 0 means no delay. This field only works in loopback sample mode or DQS sample mode.
3 - 1	**Sample Delay**

Yuntu Microelectronics

Field	Function
SMPDLY	Configure the sample delay point, offset on the serial flash clock cycle. Value 0 is no offset, value 7 is the max offset. This field only works when SMP[DTEN]=0.
0 SMPEDG	**Sample Edge** Select the sample edge for input data. This field only works when SMP[DTEN]=0. 0b - Sampling on rising edge in SDR mode, and the first sample edge is rising edge in DQS mode. 1b - Sampling on falling edge in SDR mode, and the first sample edge is falling edge in DQS mode.

23.3.1.4 QSPI ST Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DTRTY	DTFAIL		SMPEDG				DDRHLF	TXFUL	TXDMA	TXWA	TXNE	RXFUL	RXDMA		
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BUF3FUL	BUF2FUL	BUF1FUL	BUF0FUL	BUF3NE	BUF2NE	BUF1NE	BUF0NE					AHBTRN	AHBACC	APBACC	BUSY
W																
Reset	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Table 23.9: QSPI ST Register Description

Field	Function
31 DTRTY	**Data Test Retry** Data test retry indication. 0b - Don't care. 1b - Need retry the DATA TEST instruction.
30 DTFAIL	**Data Test Fail** Data test fail indicator. 0b - Don't care. 1b - DATA TEST is failed, no valid data is sampled.

Yuntu Microelectronics

Field	Function
28 SMPEDG	**Sample Edge** Sample edge indicator. 0b - SDR mode sample edge is rising edge, or the first sample edge is rising edge in DQS mode. 1b - SDR mode sample edge is falling edge, or the first sample edge is falling edge in DQS mode.
24 DDRHLF	**DDR Half** Half cycle sample offset indicator in DDR mode. 0b - Sampling without offset, the first sample edge is rising edge in DDR mode. 1b - Sampling with half cycle offset, the first sample edge is falling edge in DDR mode.
23 TXFUL	**TxFIFO Full** TxFIFO full indicator. 0b - TxFIFO is not full. 1b - TxFIFO is full.
22 TXDMA	**TxFIFO DMA** TxFIFO DMA busy indicator. 0b - TxFIFO is not under DMA access. 1b - TxFIFO is under DMA access.
21 TXWA	**TxFIFO Watermark Available** TxFIFO watermark available indicator. 0b - TxFIFO empty space is not greater than the watermark. 1b - TxFIFO empty space is greater than the watermark.
20 TXNE	**TxFIFO Not Empty** TxFIFO not empty indicator. 0b - TxFIFO data number is less than 4. 1b - TxFIFO data number is not less than 4.
19 RXFUL	**RxFIFO Full** RxFIFO full indicator. 0b - RxFIFO is not full. 1b - RxFIFO is full.
18 RXDMA	**RxFIFO DMA** RxFIFO DMA busy indicator. 0b - RxFIFO is not under DMA access. 1b - RxFIFO is under DMA access.
15	**Buffer 3 Full**

Yuntu Microelectronics

Field	Function
BUF3FUL	Buffer 3 full indicator. 0b - Buffer 3 is not full. 1b - Buffer 3 is full.
14 BUF2FUL	**Buffer 2 Full** Buffer 2 full indicator. 0b - Buffer 2 is not full. 1b - Buffer 2 is full.
13 BUF1FUL	**Buffer 1 Full** Buffer 1 full indicator. 0b - Buffer 1 is not full. 1b - Buffer 1 is full.
12 BUF0FUL	**Buffer 0 Full** Buffer 0 full indicator. 0b - Buffer 0 is not full. 1b - Buffer 0 is full.
11 BUF3NE	**Buffer 3 Not Empty** Buffer 3 not empty indicator. 0b - Buffer 3 is empty. 1b - Buffer 3 is not empty.
10 BUF2NE	**Buffer 2 Not Empty** Buffer 2 not empty indicator. 0b - Buffer 2 is empty. 1b - Buffer 2 is not empty.
9 BUF1NE	**Buffer 1 Not Empty** Buffer 1 not empty indicator. 0b - Buffer 1 is empty. 1b - Buffer 1 is not empty.
8 BUF0NE	**Buffer 0 Not Empty** Buffer 0 not empty indicator. 0b - Buffer 0 is empty. 1b - Buffer 0 is not empty.
3 AHBTRN	**AHB Transmit** AHB transmit indictor. 0b - No AHB transmit. 1b - AHB transmit is ongoing.
2	**AHB Access**

Yuntu Microelectronics

Field	Function
AHBACC	AHB access serial flash indictor. 0b - Serial flash is not under AHB access. 1b - Serial flash is under AHB access.
1 APBACC	**APB Access** APB access serial flash indictor. 0b - Serial flash is not under APB access. 1b - Serial flash is under APB access.
0 BUSY	**Busy** Serial flash busy indicator. 0b - Serial flash is idle. 1b - Serial flash is under AHB or APB access.

23.3.1.5 QSPI IF Register

Offset: 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R								IIE	TXF	TXUF			RXOF	RXWE		
W								W1C	W1C	W1C			W1C	W1C		
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R					AHBTO	AHBBSE	AHBBOF	AHBWAE	APBTE	APBAE						CMDTF
W					W1C	W1C	W1C	W1C	W1C	W1C						W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 23.10: QSPI IF Register Description

Field	Function
24 IIE	**Illegal Instruction Error** Illegal instruction indicator. 0b - No error detected. 1b - Illegal instruction detected.
23	**TxFIFO Fill**

Yuntu Microelectronics

Field	Function
TXF	Before writing to the TxFIFO, this field should be cleared. Then, it should be read back. 0b - TxFIFO can be considered as full. 1b - TxFIFO is not full and can fill in more data.
22 TXUF	**TxFIFO Underflow** TxFIFO underflow indicator. 0b - No underflow detected. 1b - Underflow detected.
19 RXOF	**RxFIFO Overflow** RxFIFO overflow indicator. 0b - No overflow detected. 1b - Overflow detected.
18 RXWE	**RxFIFO Watermark Exceed** RxFIFO data number exceed watermark indicator. 0b - RxFIFO data number is not greater than the watermark. 1b - RxFIFO data number is greater than the watermark.
11 AHBTO	**AHB Timeout** AHB access timeout indicator. If QSPI has no response to the AHB access after the time specified by AHBCFG[AHBWT], this flag will be set. 0b - No AHB access timeout detected. 1b - AHB access timeout detected.
10 AHBBSE	**AHB Burst Size Error** AHB burst size error indicator. If burst size greater than the data size, this flag will set. 0b - No error detected. 1b - AHB burst size error detected.
9 AHBBOF	**AHB Buffer Overflow** AHB current buffer overflow indicator. 0b - No buffer overflow detected. 1b - Buffer overflow detected.
8 AHBWAE	**AHB Write Access Error** AHB write access error indicator. 0b - No error detected. 1b - AHB write access detected.
7 APBTE	**APB Trigger Error** APB command trigger during AHB access error indicator. 0b - No error detected. 1b - APB command trigger detected during AHB access the serial flash.
6	**APB Access Error**

Yuntu Microelectronics

Field	Function
APBAE	Illegal register access during APB access error indicator. 0b - No error detected. 1b - Write access to the APBCFG/SFA/RBCS register detected during APB access the serial flash.
0 CMDTF	**Command Transfer Finish** LUT command transfer finish indicator. Only for APB command. 0b - LUT command transfer is ongoing or no transfer. 1b - LUT command transfer finish.

23.3.1.6 QSPI IE Register

Offset: 14h

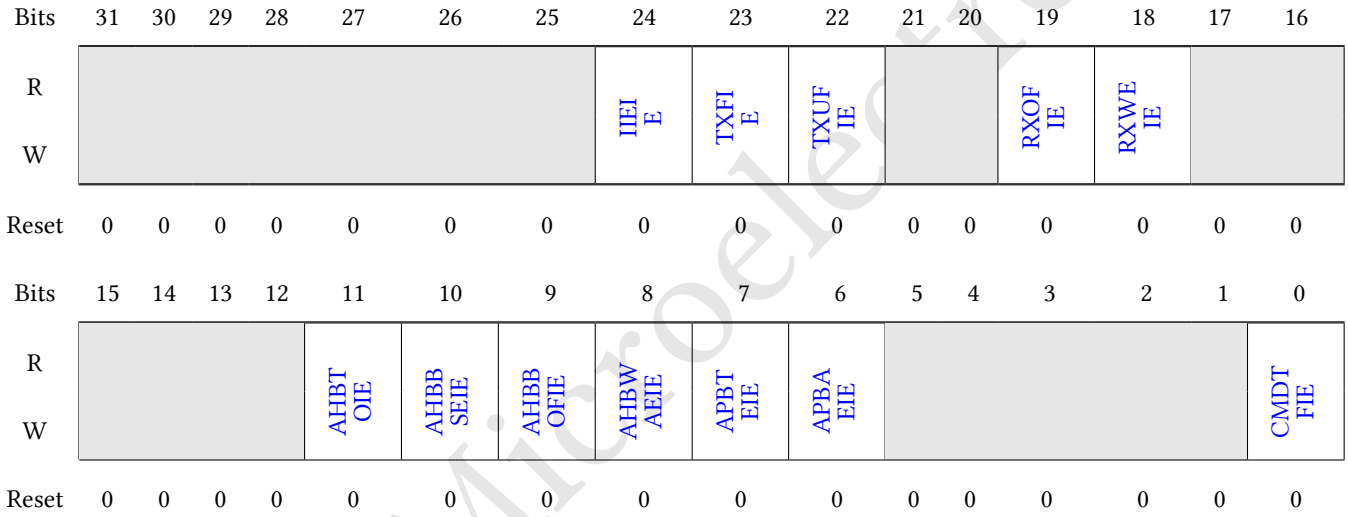


Table 23.11: QSPI IE Register Description

Field	Function
24 IIEIE	**Illegal Instruction Error Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
23 TXFIE	**TxFIFO Fill Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
22 TXUFIE	**TxFIFO Underflow Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.

Yuntu Microelectronics

Field	Function
19 RXOFIE	**RxFIFO Overflow Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
18 RXWEIE	**RxFIFO Watermark Exceed Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
11 AHBTOIE	**AHB Timeout Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
10 AHBBSEIE	**AHB Burst Size Error Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
9 AHBBOFIE	**AHB Buffer Overflow Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
8 AHBWAEIE	**AHB Write Access Error Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
7 APBTEIE	**APB Trigger Error Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
6 APBAEIE	**APB Access Error Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.
0 CMDTFIE	**Command Transfer Finish Interrupt Enable** Interrupt enable/disable control. 0b - Interrupt is disabled. 1b - Interrupt is enabled.

Yuntu Microelectronics

23.3.1.7 QSPI SPNDST Register

Offset: 18h

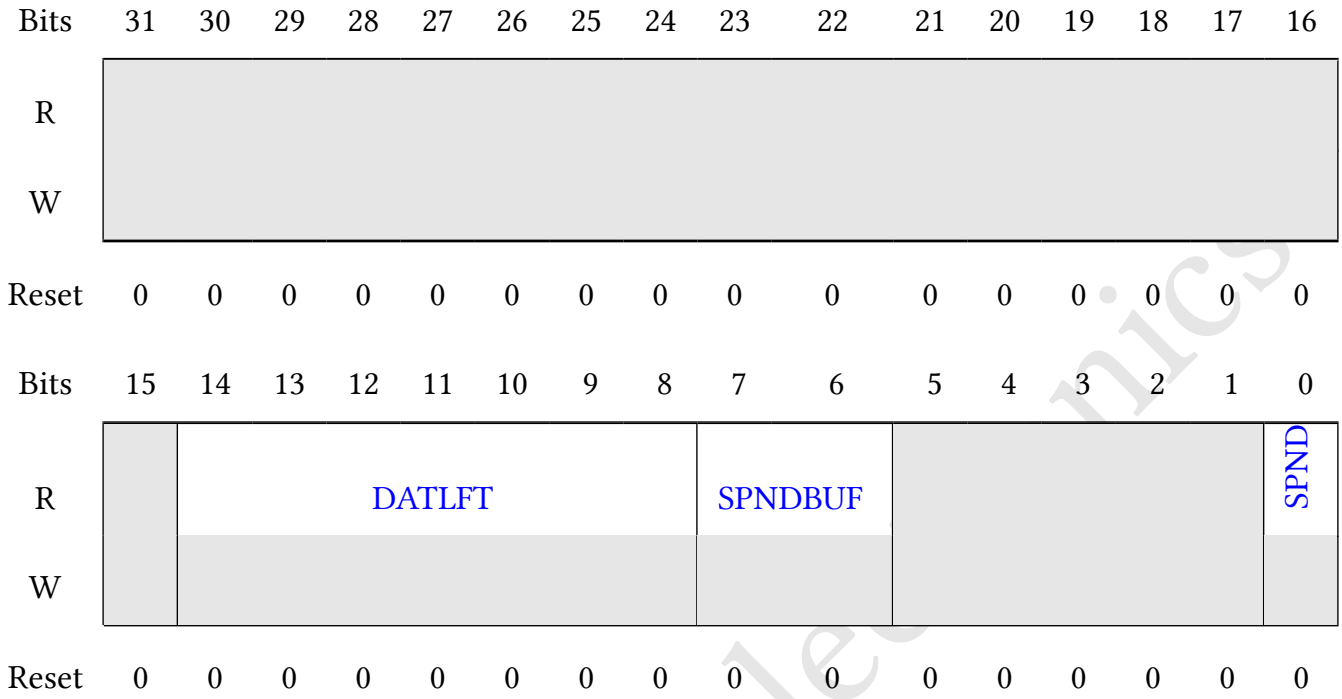


Table 23.12: QSPI SPNDST Register Description

Field	Function
14 - 8 DATLFT	**Data Left** Suspended data left indicator, in 64 bit. Value 0 indicate the left data is less than 64 bits.
7 - 6 SPNDBUF	**Suspend Buffer** Suspended buffer indicator. 00b - Buffer 0 is suspended. 01b - Buffer 1 is suspended. 10b - Buffer 2 is suspended. 11b - Buffer 3 is suspended.
0 SPND	**Suspend** Suspend state indicator. 0b - No suspend occurs. 1b - Suspend occurs.

23.3.1.8 QSPI RBCS Register

Offset: 1ch

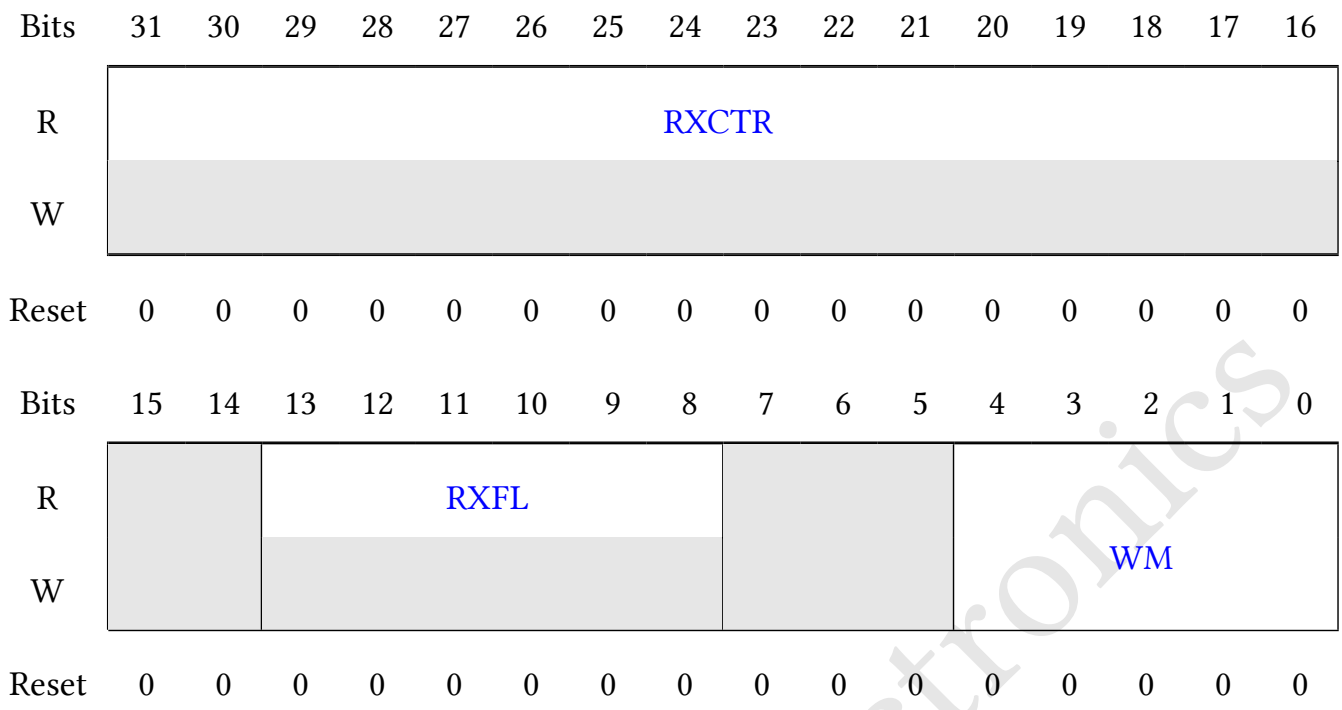


Table 23.13: QSPI RBCS Register Description

Field	Function
31 - 16 RXCTR	**RxFIFO Counter** RxFIFO read data counter, in 32 bit. Value 1 indicate 4 bytes data have been read out from RxFIFO. This counter is incremented by the number (RBCS[WM]+1). This filed can be cleared by MCFG[RXFCLR].
13 - 8 RXFL	**RxFIFO Fill Level** Current RxFIFO data number indicator, in 32 bit. Value 1 indicate current RxFIFO data number is 4 bytes. This filed can be cleared by MCFG[RXFCLR].
4 - 0 WM	**Watermark** RxFIFO watermark, fill lever > watermark will trigger the DMA or interrupt.

Yuntu Microelectronics

23.3.1.9 QSPI TBCS Register

Offset: 20h

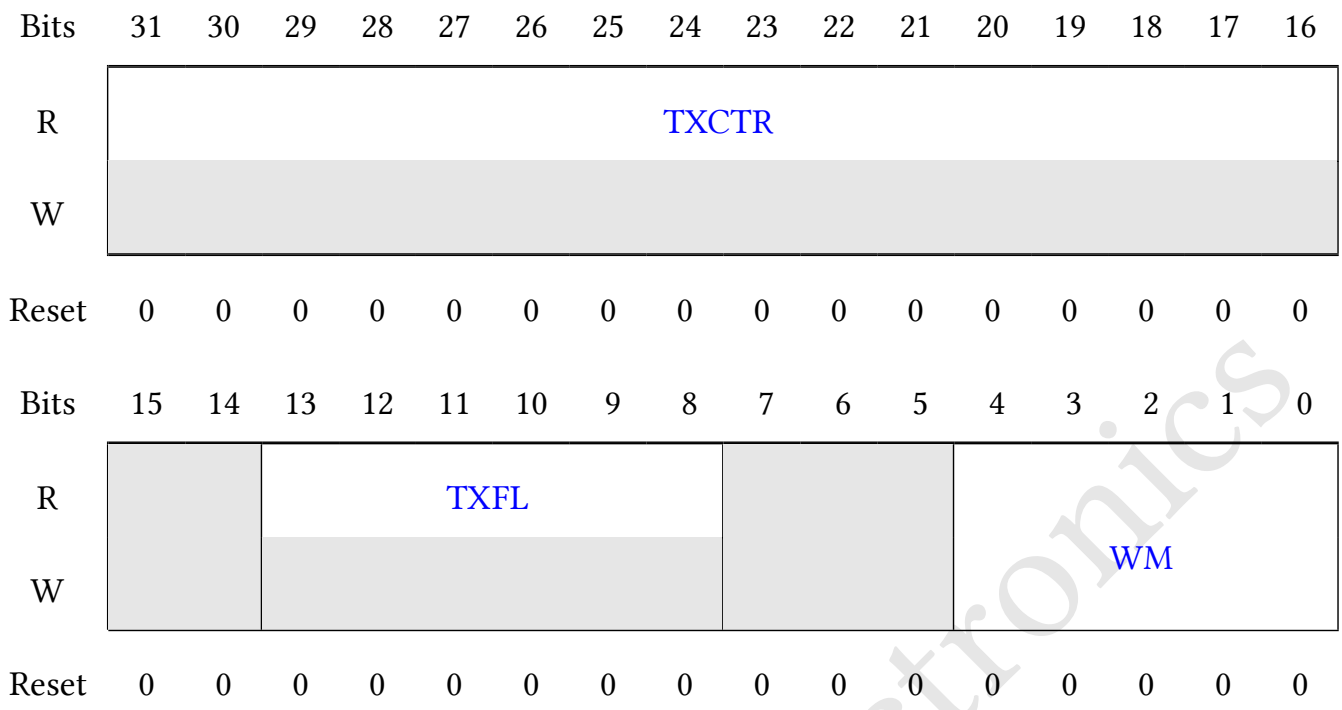


Table 23.14: QSPI TBCS Register Description

Field	Function
31 - 16 TXCTR	**TxFIFO Counter** TxFIFO write data counter, in 32 bit. Value 1 indicate 4 bytes data have been written into TxFIFO. This filed can be cleared by MCFG[TXFCLR].
13 - 8 TXFL	**TxFIFO Fill Level** Current TxFIFO data number indicator, in 32 bit. Value 1 indicate current TxFIFO data number is 4 bytes. This filed can be cleared by MCFG[TXFCLR].
4 - 0 WM	**Watermark** TxFIFO watermark, (max TxFIFO depth - fill lever) > watermark will trigger the DMA.

Yuntu Microelectronics

23.3.1.10 QSPI APBCFG Register

Offset: 24h

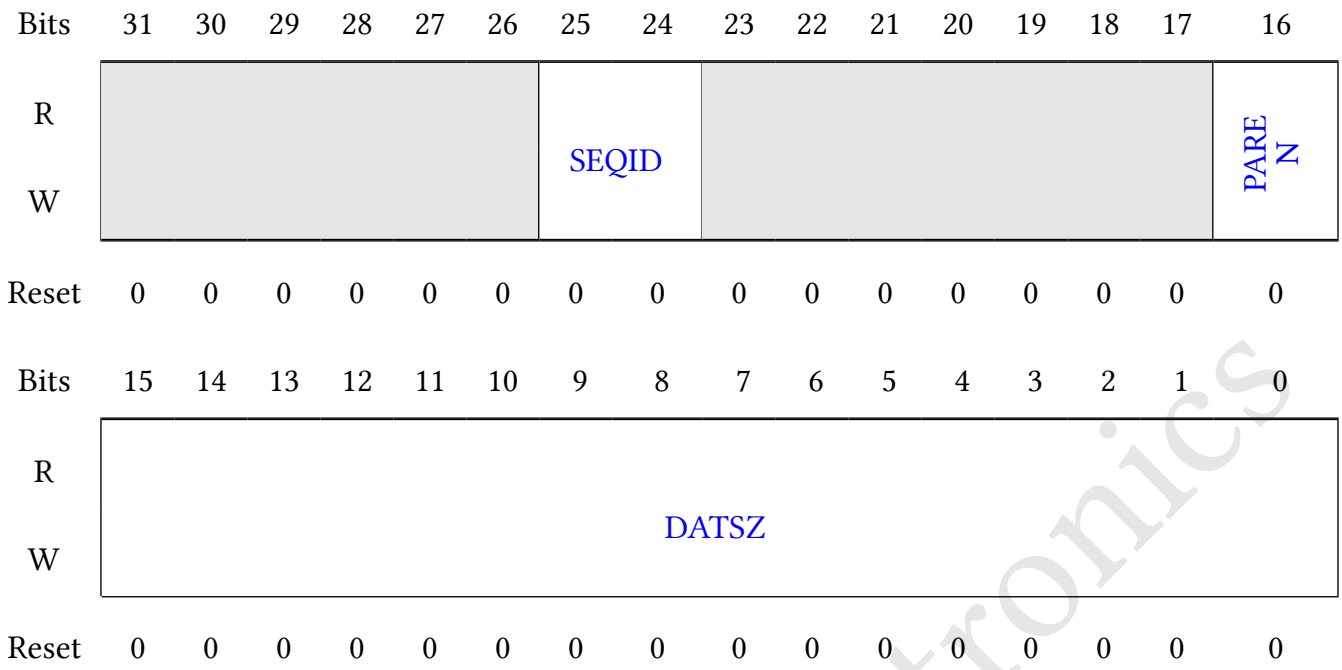


Table 23.15: QSPI APBCFG Register Description

Field	Function
25 - 24 SEQID	**Sequence ID** Select the LUT sequence when access to the serial flash by APB. A write to this field will trigger the transaction.
16 PAREN	**Parallel Enable** Enable/disable the parallel mode when access to serial flash by APB. 0b - Parallel mode is disabled. 1b - Parallel mode is enabled.
15 - 0 DATSZ	**Data Size** Data size of APB write/read access to the serial flash, in 8 bit. Value 1 indicate the write/read data is one byte. When this field is value 0, the write/read data size is up to the operand of the write/read LUT command.

Yuntu Microelectronics

23.3.1.11 QSPI AHBCFG Register

Offset: 28h

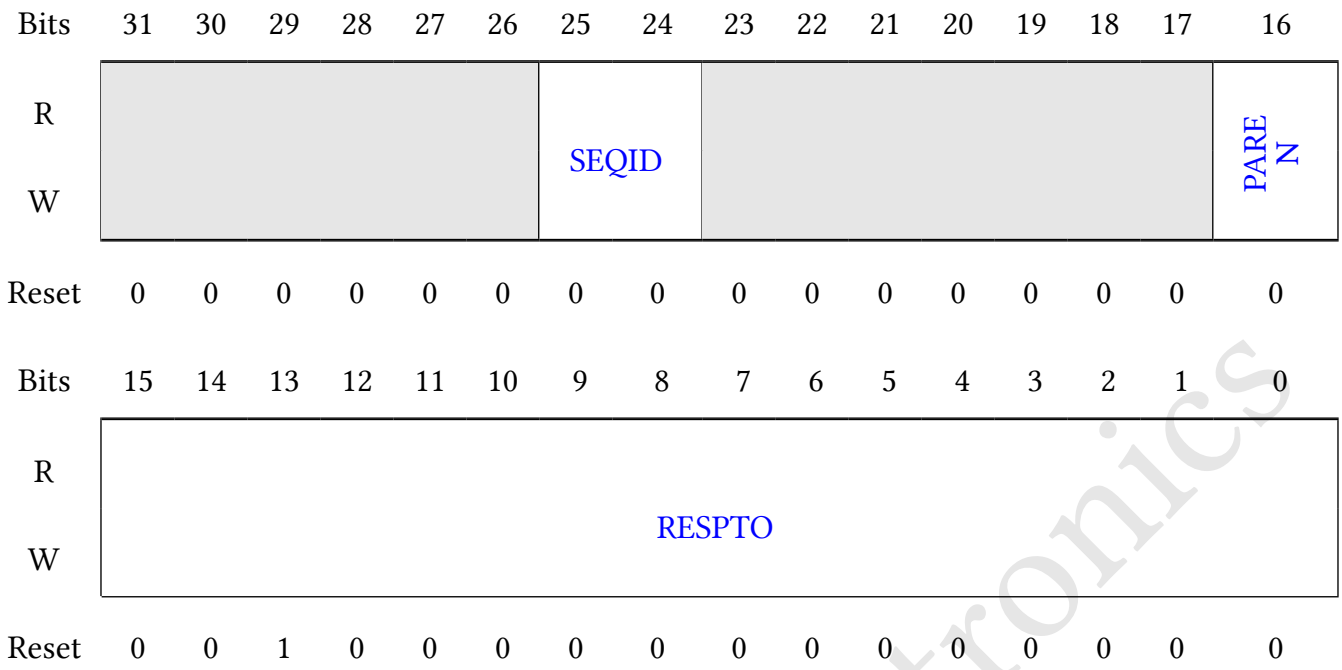


Table 23.16: QSPI AHBCFG Register Description

Field	Function
25 - 24 SEQID	**Sequence ID** Select the LUT sequence when access to the serial flash by AHB.
16 PAREN	**Parallel Enable** Enable/disable the parallel mode when access to serial flash by AHB. 0b - Parallel mode is disabled. 1b - Parallel mode is enabled.
15 - 0 RESPTO	**Response Timeout** Max wait time for AHB access. This field can only be changed when MEN[EN]=0. AHB read access to the serial flash will timeout if data not transmitted or no data received after RESPTO*1024 AHB clock cycles.

23.3.1.12 QSPI BUF0CFG Register

Offset: 2ch

Yuntu Microelectronics

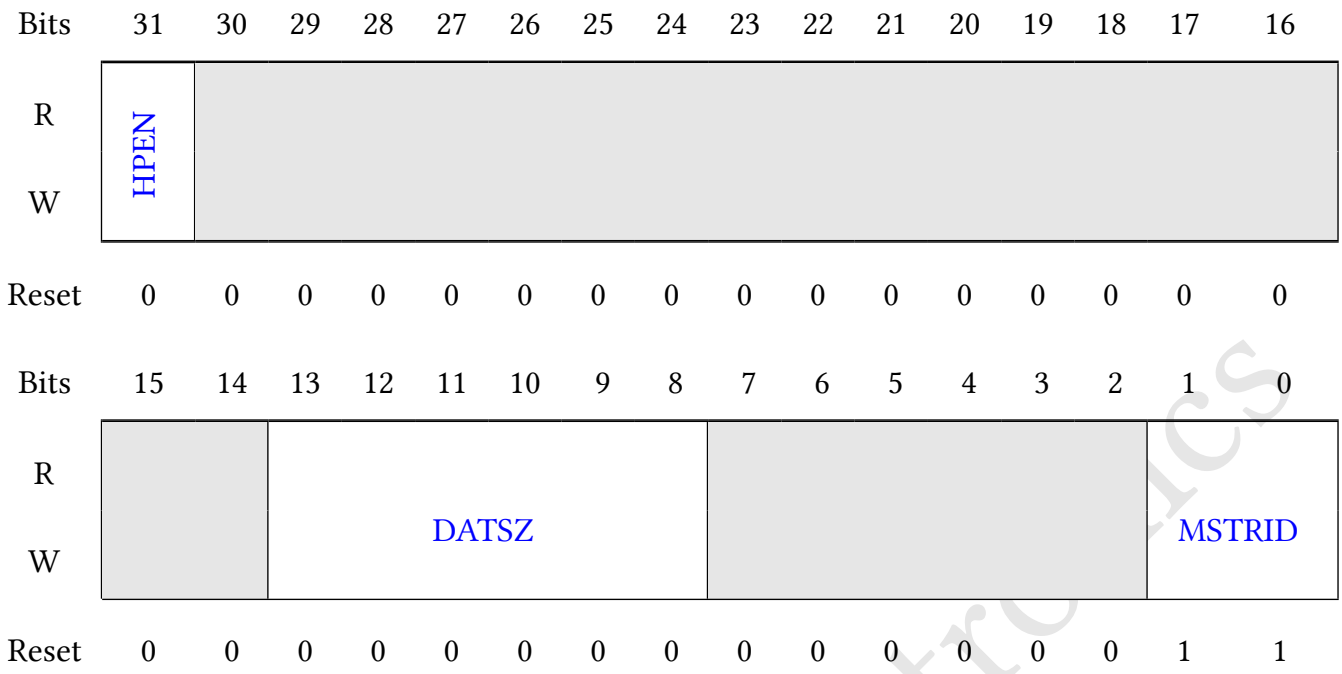


Table 23.17: QSPI BUF0CFG Register Description

Field	Function
31 HPEN	**High Priority Enable** Buffer 0 high priority enable/disable control. 0b - Buffer 0 is same as other buffers. 1b - Buffer 0 has higher priority than other buffers, access to buffer 0 will result in other buffers' access being suspended.
13 - 8 DATSZ	**Data Size** Data size of AHB read access to the serial flash, in 64 bit. Value 1 indicate the read data is 8 bytes. When this field is value 0, the read data size is up to the operand of the read LUT command.
1 - 0 MSTRID	**Master ID** Define the AHB master ID which can access to the buffer 0. 00b - AXI 01b - DMA 10b - RESERVED 11b - AHBP

23.3.1.13 QSPI BUF1CFG Register

Offset: 30h

Yuntu Microelectronics

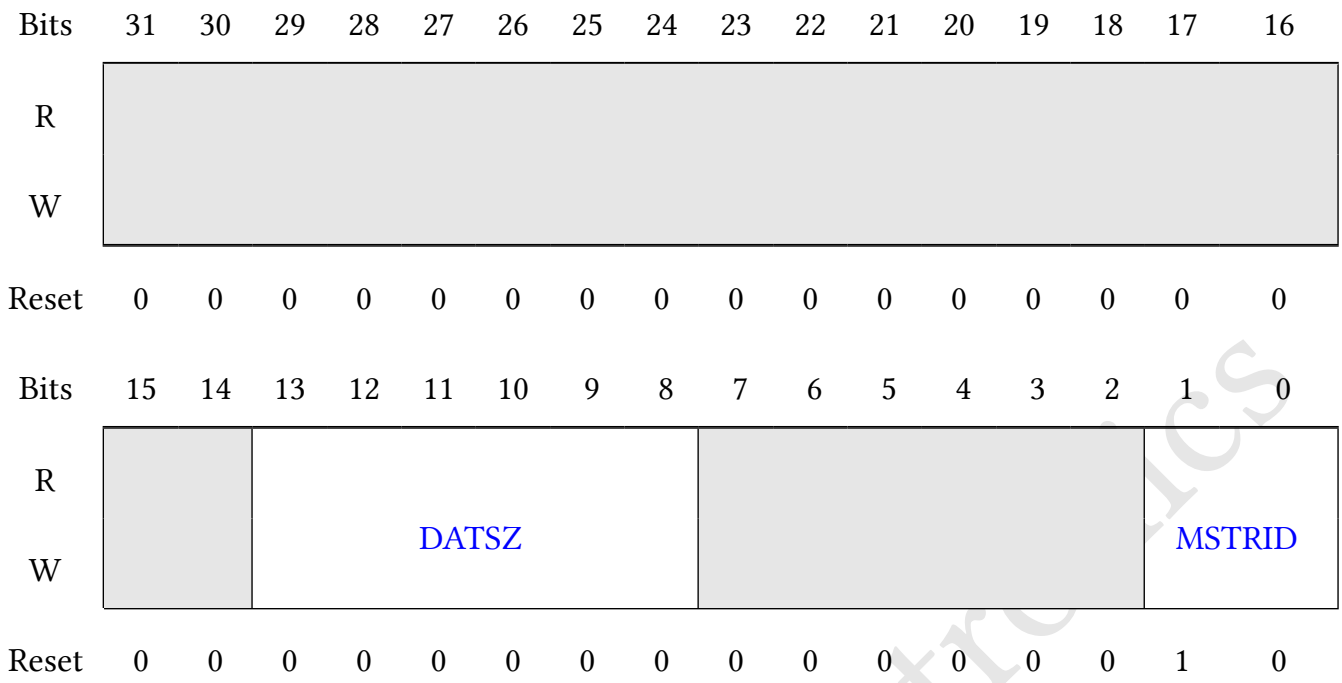


Table 23.18: QSPI BUF1CFG Register Description

Field	Function
13 - 8 DATSZ	**Data Size** Data size of AHB read access to the serial flash, in 64 bit. Value 1 indicate the read data is 8 bytes. When this field is value 0, the read data size is up to the operand of the read LUT command.
1 - 0 MSTRID	**Master ID** Define the AHB master ID which can access to the buffer 1. 00b - AXI 01b - DMA 10b - RESERVED 11b - AHBP

Yuntu Microelectronics

23.3.1.14 QSPI BUF2CFG Register

Offset: 34h

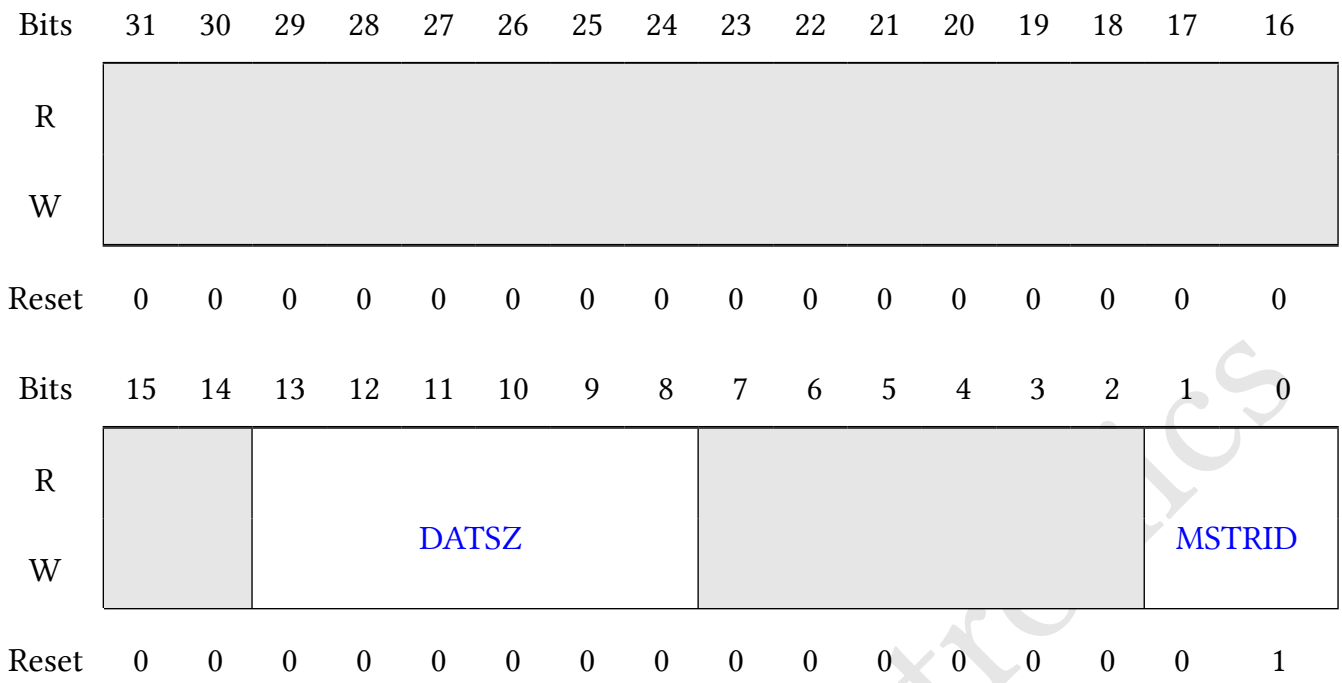


Table 23.19: QSPI BUF2CFG Register Description

Field	Function
13 - 8 DATSZ	**Data Size** Data size of AHB read access to the serial flash, in 64 bit. Value 1 indicate the read data is 8 bytes. When this field is value 0, the read data size is up to the operand of the read LUT command.
1 - 0 MSTRID	**Master ID** Define the AHB master ID which can access to the buffer 2. 00b - AXI 01b - DMA 10b - RESERVED 11b - AHBP

Yuntu Microelectronics

23.3.1.15 QSPI BUF3CFG Register

Offset: 38h

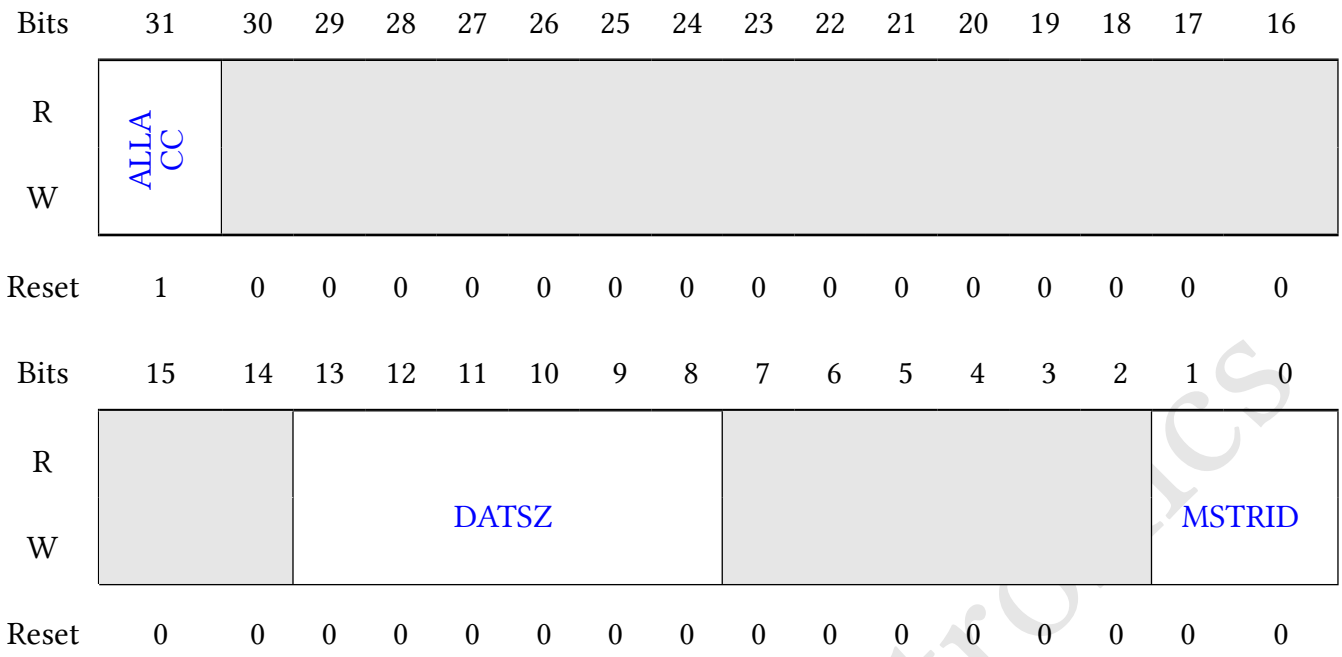


Table 23.20: QSPI BUF3CFG Register Description

Field	Function
31 ALLACC	**All Access** Buffer 3 all master access enable/disable control. 0b - Buffer 3 can only be accessed by matched AHB master that defined by BUF3CFG[MSTRID]. 1b - Buffer 3 can be accessed by all AHB masters.
13 - 8 DATSZ	**Data Size** Data size of AHB read access to the serial flash, in 64 bit. Value 1 indicate the read data is 8 bytes. When this field is value 0, the read data size is up to the operand of the read LUT command.
1 - 0 MSTRID	**Master ID** Define the AHB master ID which can access to the buffer 3, this field is meaningless when BUF3CFG[ALLACC]=1. 00b - AXI 01b - DMA 10b - RESERVED 11b - AHBP

23.3.1.16 QSPI BUF0IND Register

Offset: 3ch

Yuntu Microelectronics

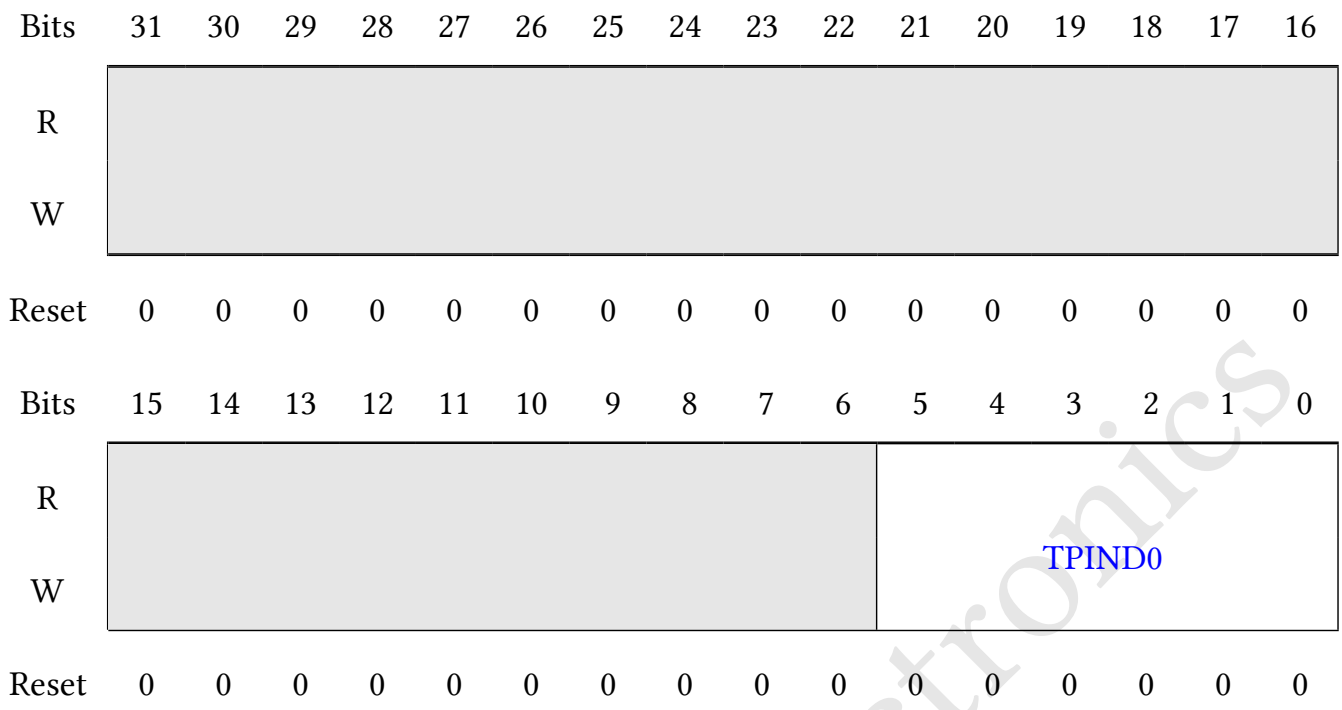


Table 23.21: QSPI BUF0IND Register Description

Field	Function
5 - 0 TPIND0	**Top Index 0** Define the buffer 0 space range, sram address from 0 to (BUF0IND[TPIND0]-1) is the buffer 0 space.

Yuntu Microelectronics

23.3.1.17 QSPI BUF1IND Register

Offset: 40h

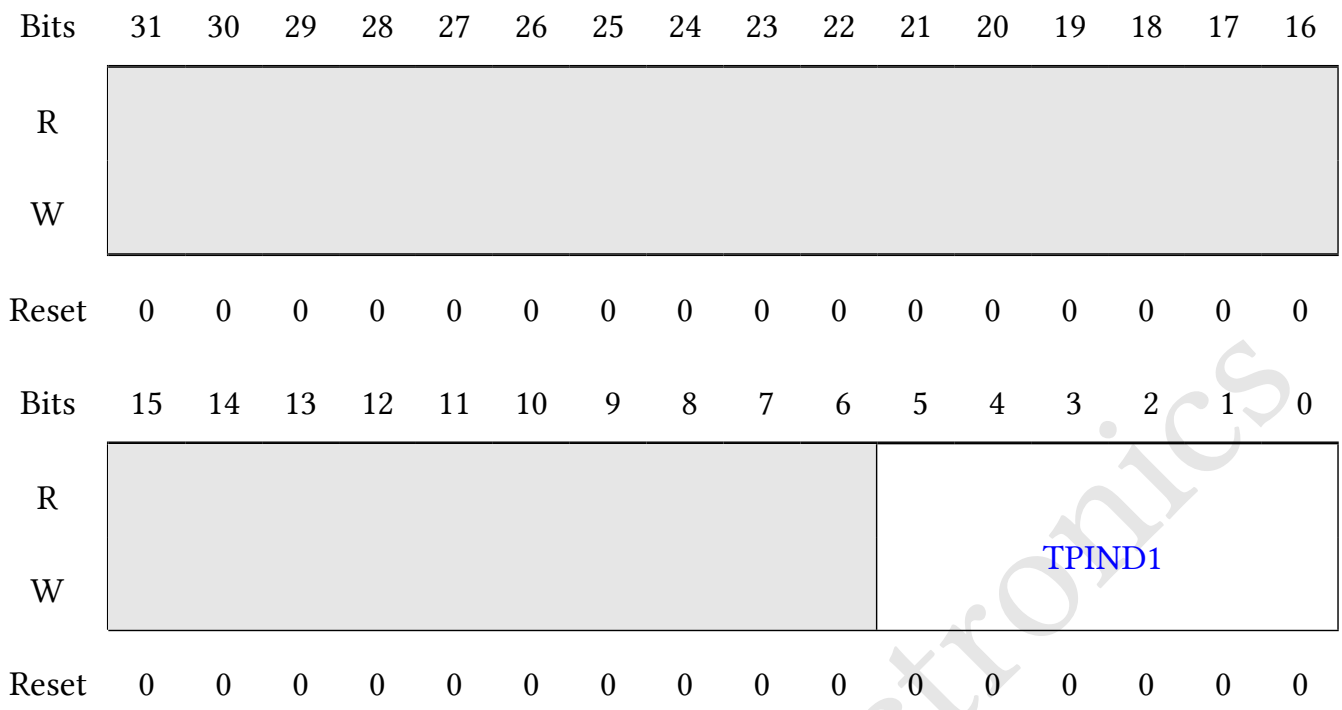


Table 23.22: QSPI BUF1IND Register Description

Field	Function
5 - 0 TPIND1	**Top Index 1** Define the buffer 1 space range, sram address from BUF0IND[TPIND0] to (BUF1IND[TPIND1]-1) is the buffer 1 space.

Yuntu Microelectronics

23.3.1.18 QSPI BUF2IND Register

Offset: 44h

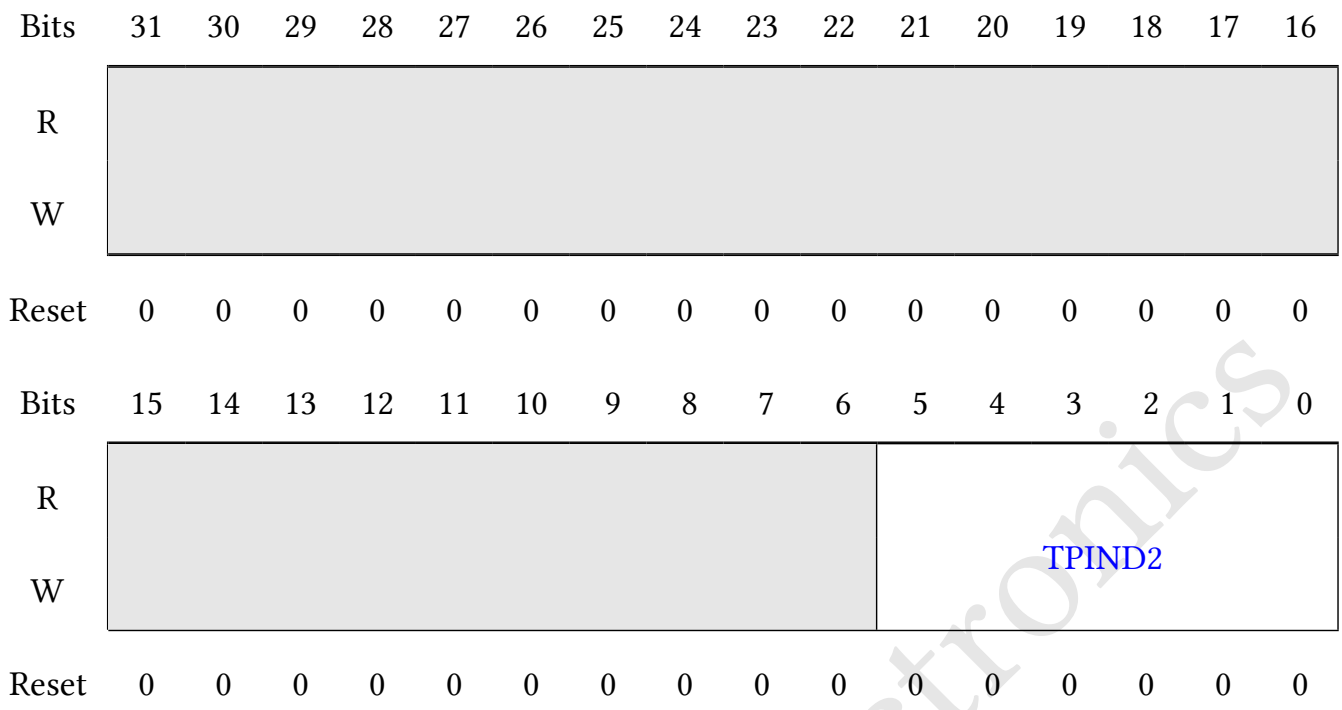


Table 23.23: QSPI BUF2IND Register Description

Field	Function
5 - 0 TPIND2	**Top Index 2** Define the buffer 2 space range, sram address from BUF1IND[TPIND1] to (BUF2IND[TPIND2]-1) is the buffer 2 space.

Yuntu Microelectronics

23.3.1.19 QSPI FACFG Register

Offset: 48h

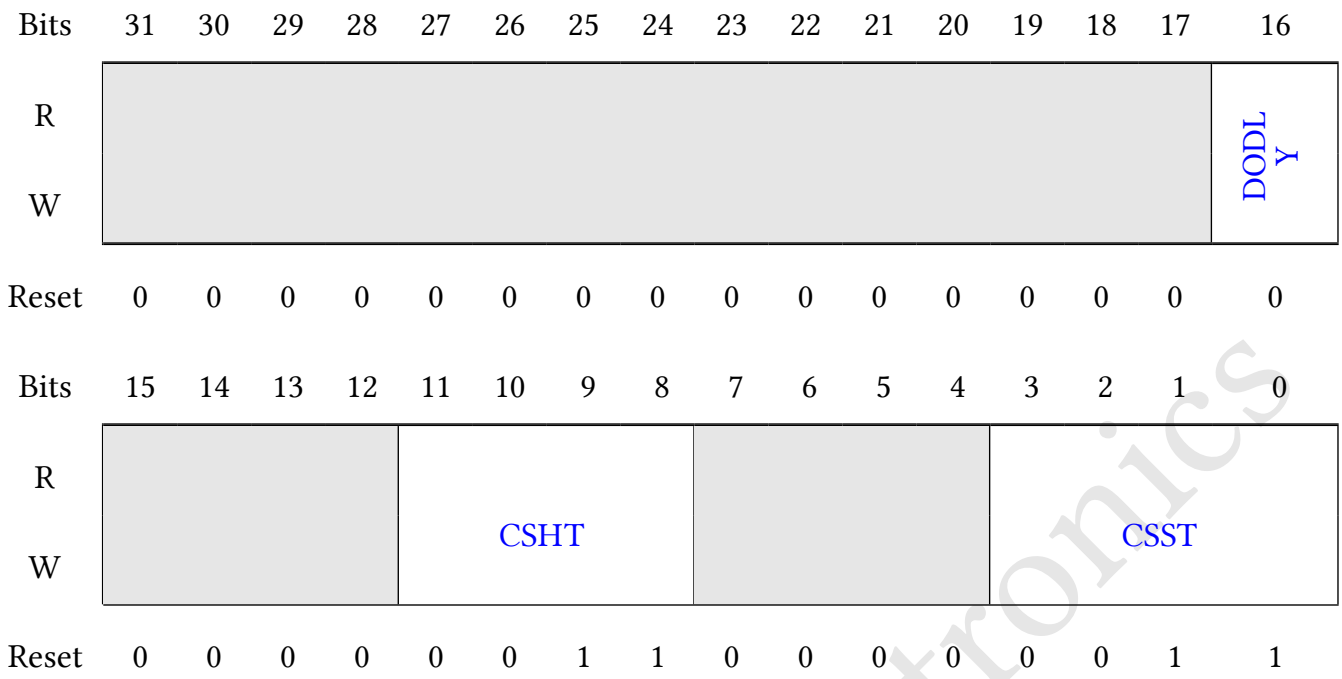


Table 23.24: QSPI FACFG Register Description

Field	Function
16 DODLY	**Data Output Delay** Data output delay control in DDR mode. 0b - Data output is aligned to the clock edge in DDR mode. 1b - Data output is 1/4 serial flash clock cycle later than the clock edge in DDR mode.
11 - 8 CSHT	**Chip Select Hold Time** Define the chip select signal rises hold time that controls the chip select signal rises opportunity after clock output end.
3 - 0 CSST	**Chip Select Setup Time** Define the chip select signal falls setup time that controls the chip select signal falls opportunity before clock output start.

Yuntu Microelectronics

23.3.1.20 QSPI SFA Register

Offset: 4ch

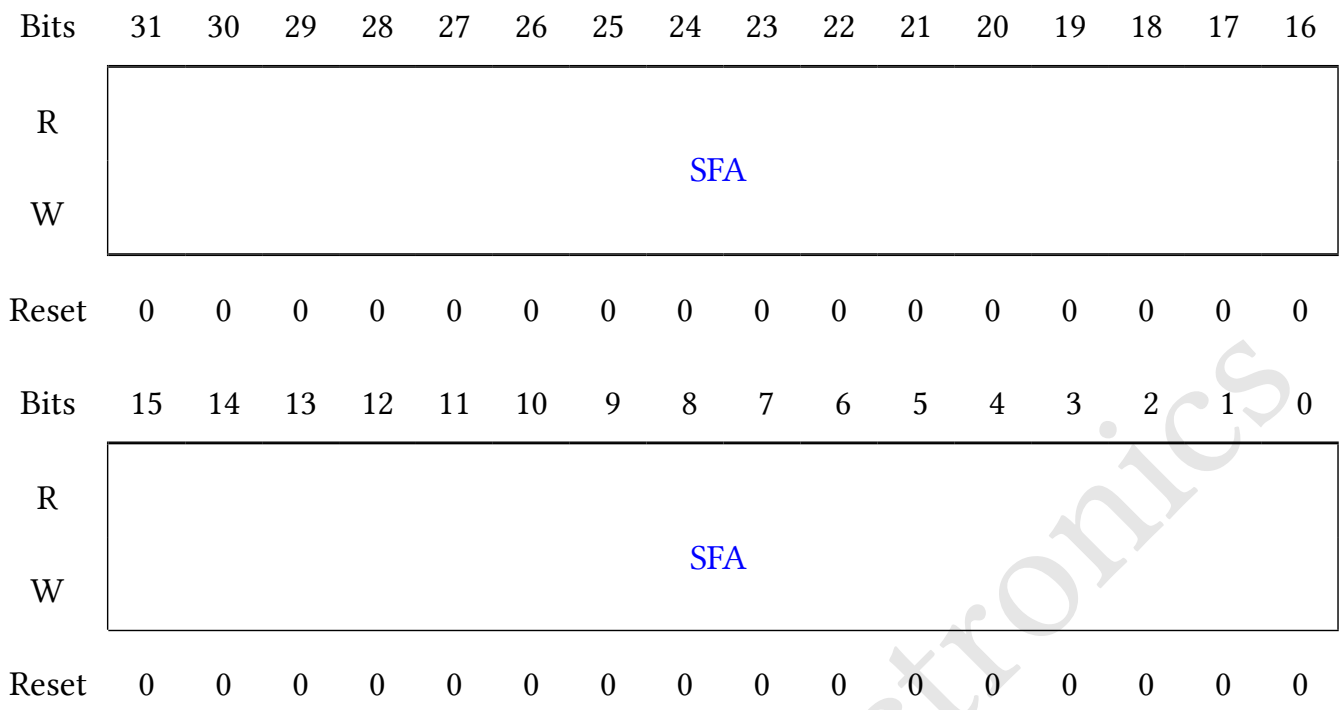


Table 23.25: QSPI SFA Register Description

Field	Function
31 - 0 SFA	**Serial Flash Address** Define the serial flash access address. It's only used by APB access.

Yuntu Microelectronics

23.3.1.21 QSPI SFACFG Register

Offset: 50h

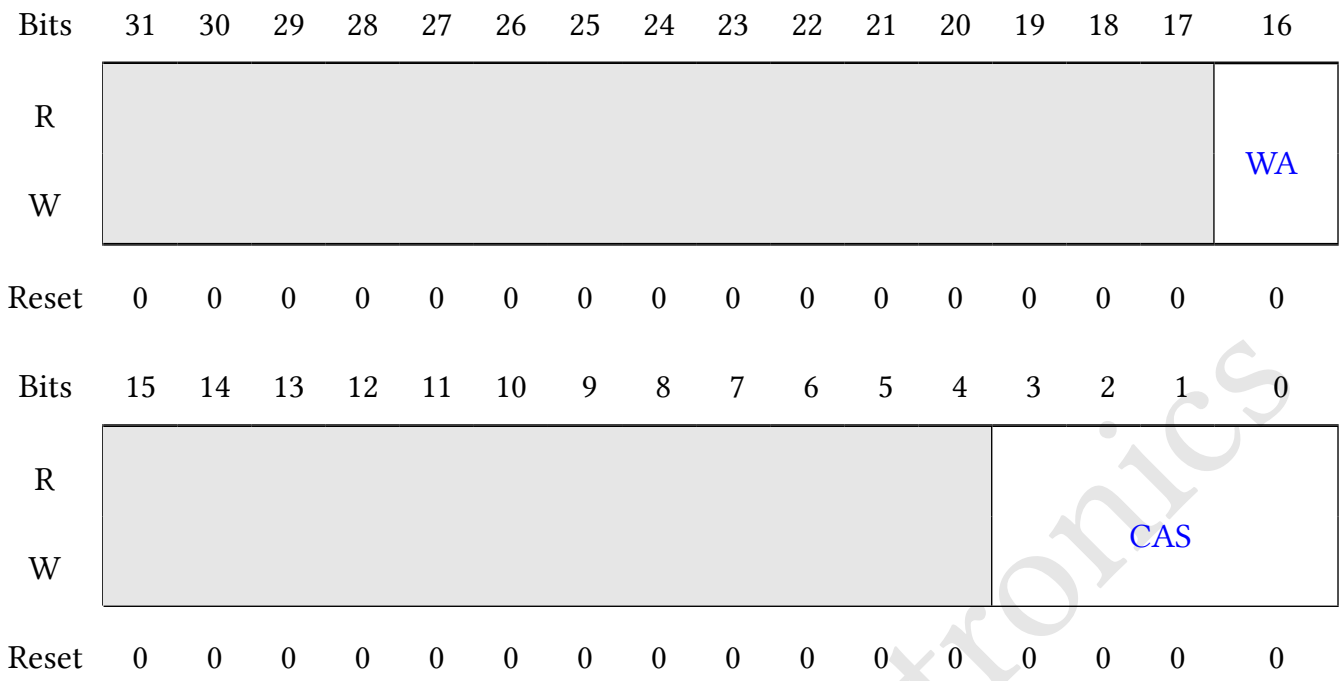


Table 23.26: QSPI SFACFG Register Description

Field	Function
16 WA	**Word Addressable** Word address access enable/disable control. When this bit enabled, the Flash will be accessed in terms of 16 bits. 0b - Word address access is disabled, the serial flash is accessed by byte address. 1b - Word address access is enabled, the serial flash is accessed by half-word address.
3 - 0 CAS	**Column Address Space** Define the column address space in address phase. The column address locate at the end of communication address.

Yuntu Microelectronics

23.3.1.22 QSPI SFATA Register

Offset: 54h

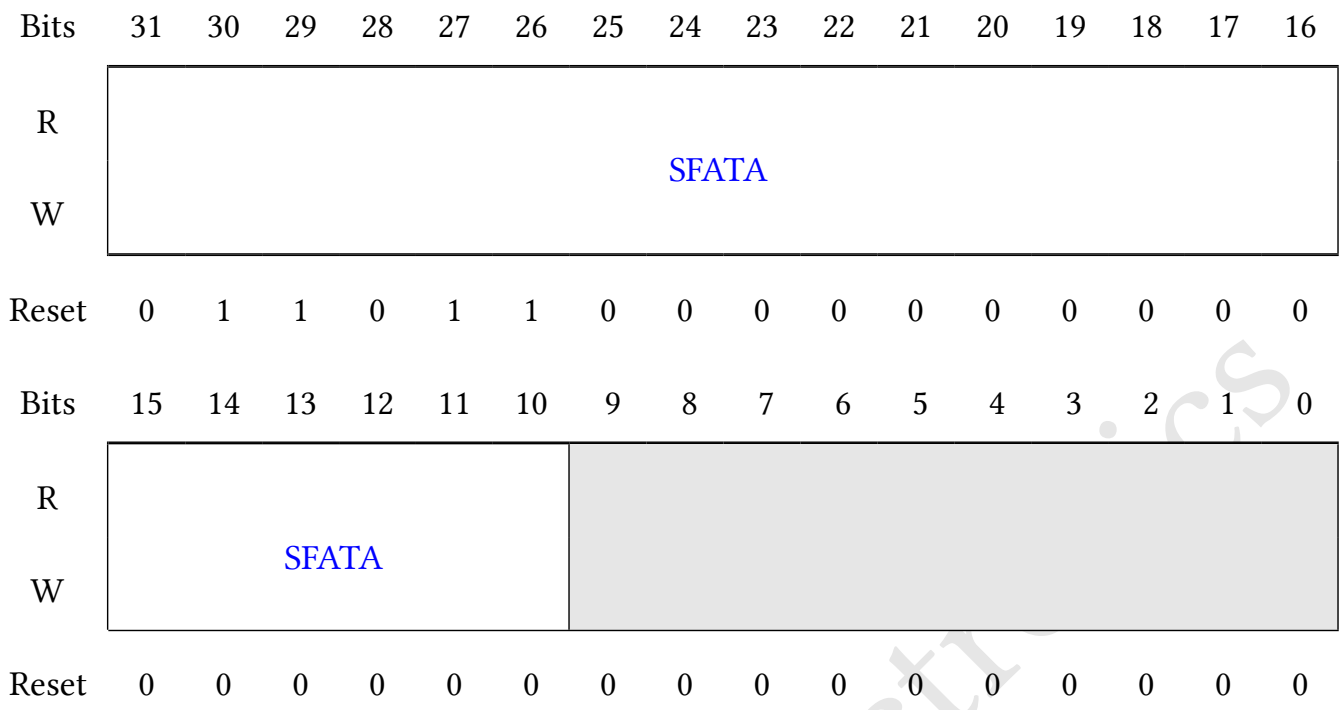


Table 23.27: QSPI SFATA Register Description

Field	Function
31 - 10 SFATA	**Serial Flash A Top Address** Define the flash A top address, flash A address access space is from 0x68000000 to SFATA[SFATA]-1.

23.3.1.23 QSPI SFA1TA Register

Offset: 58h

Yuntu Microelectronics

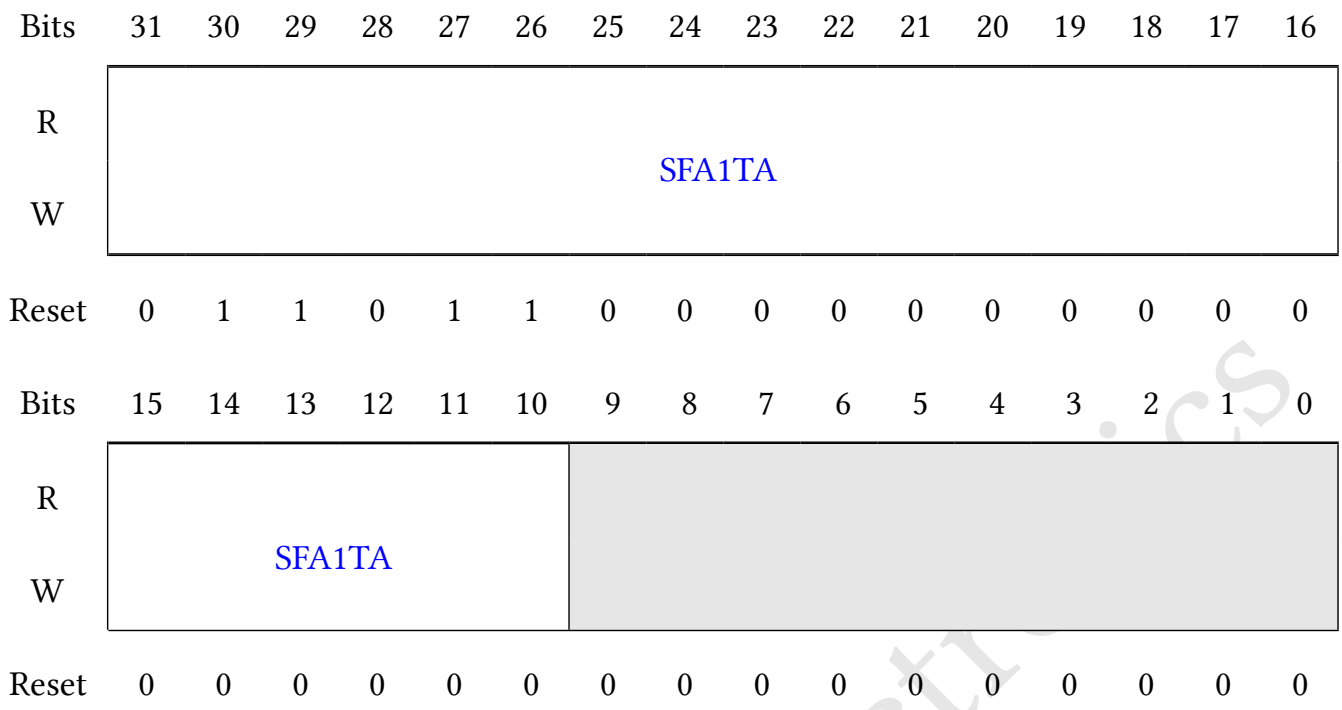


Table 23.28: QSPI SFA1TA Register Description

Field	Function
31 - 10 SFA1TA	Write same value with SFATA[SFATA].

23.3.1.24 QSPI SFBTA Register

Offset: 5ch

Yuntu Microelectronics

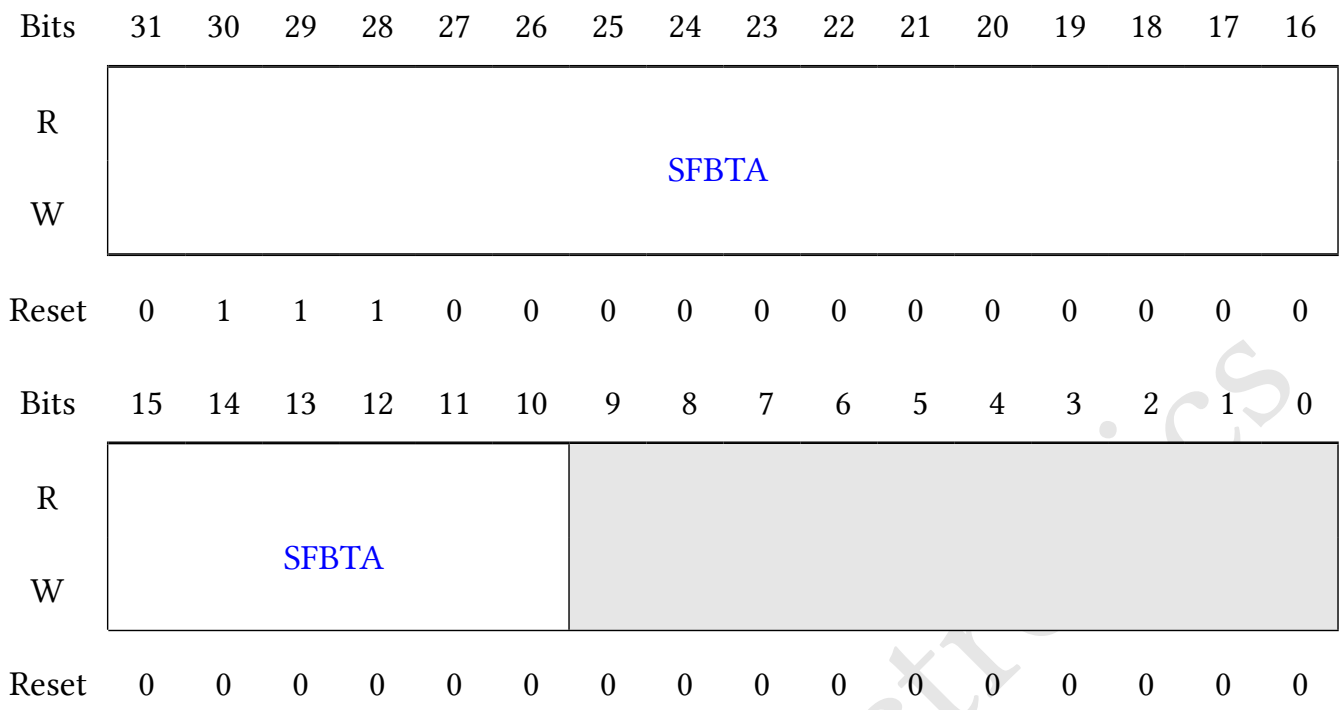


Table 23.29: QSPI SFBTA Register Description

Field	Function
31 - 10 SFBTA	**Serial Flash B Top Address** Define the flash B top address, flash B address access space is from SFATA[SFATA] to SFBTA[SFBTA]-1.

Yuntu Microelectronics

23.3.1.25 QSPI SFB1TA Register

Offset: 60h

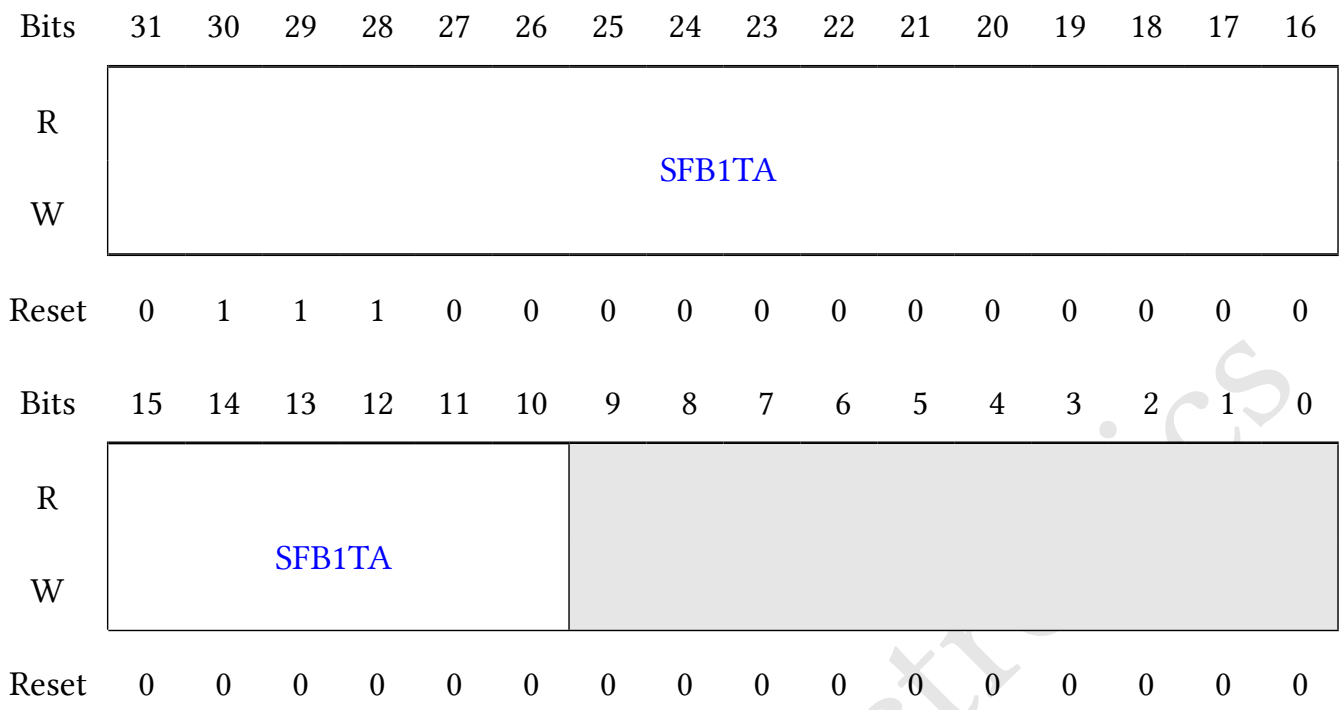


Table 23.30: QSPI SFB1TA Register Description

Field	Function
31 - 10 SFB1TA	Write same value with SFBTA[SFBTA].

Yuntu Microelectronics

23.3.1.26 QSPI DTPR Register

Offset: 64h

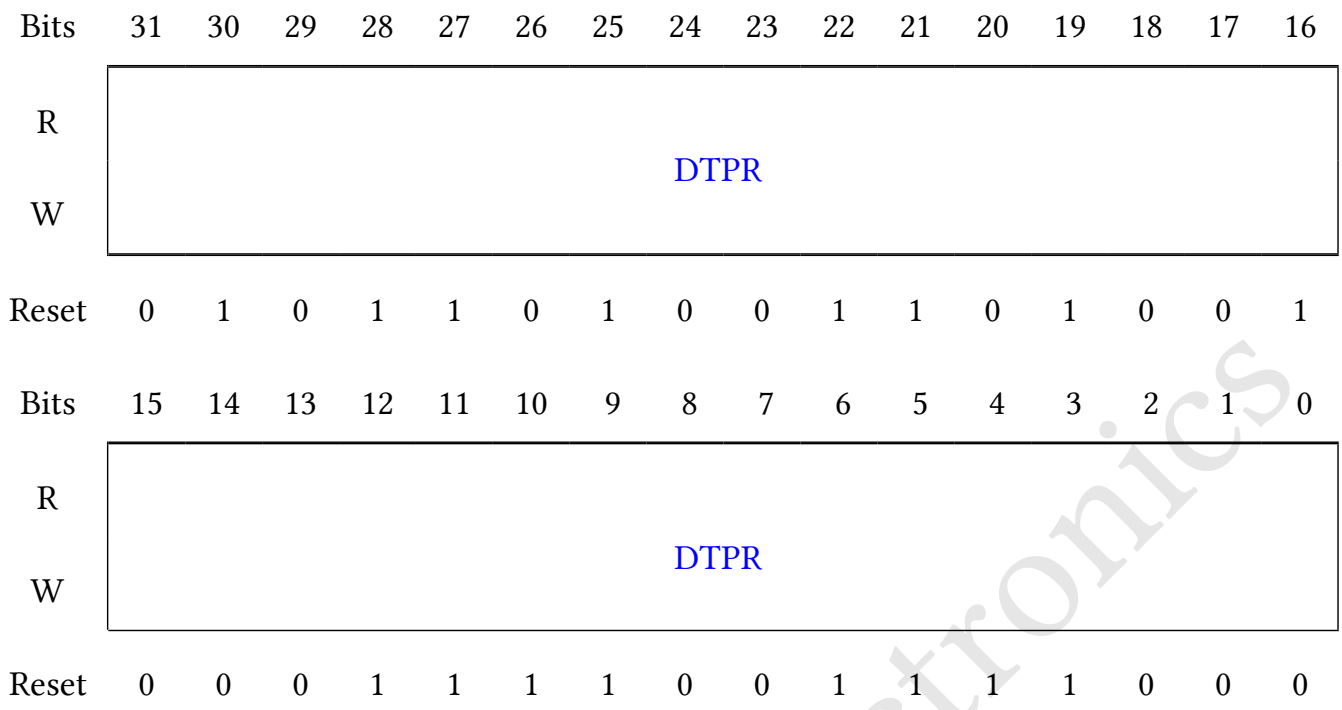


Table 23.31: QSPI DTPR Register Description

Field	Function
31 - 0 DTPR	**Data Test Pattern** Define the pattern that used for data test check.

Yuntu Microelectronics

23.3.1.27 QSPI TXDAT Register

Offset: 7ch

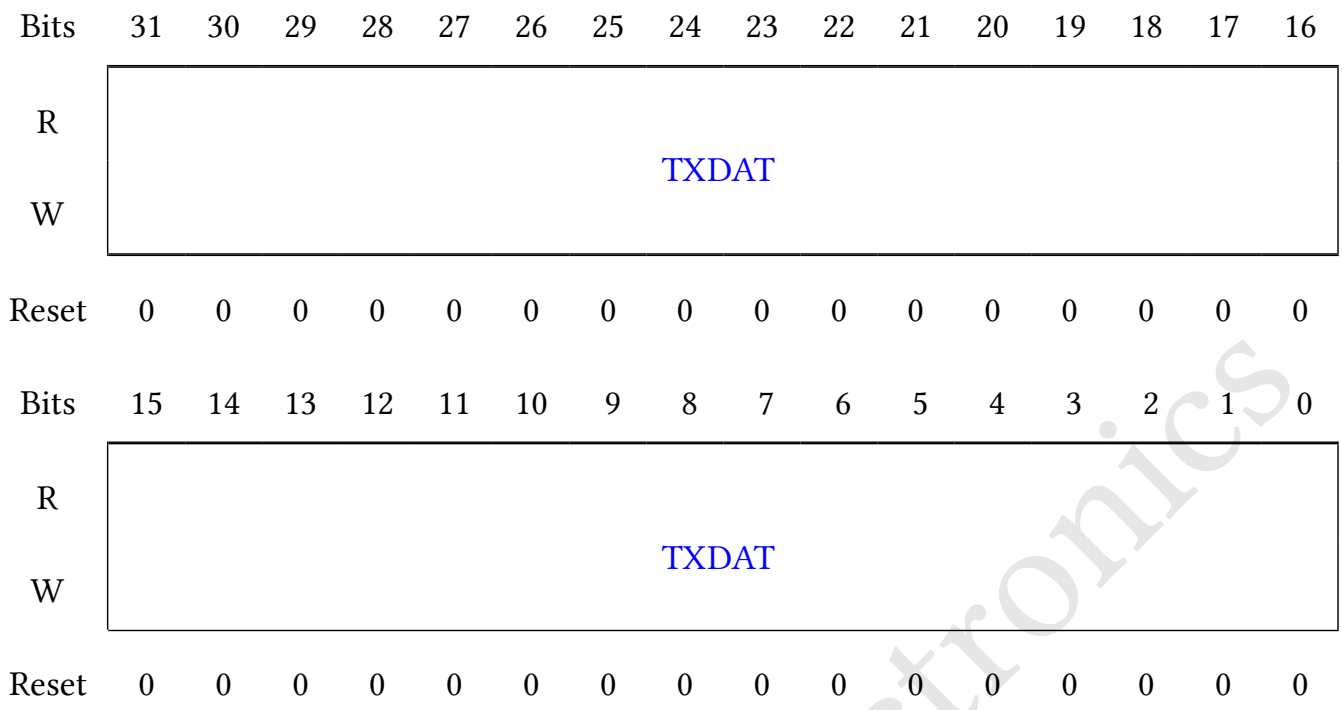


Table 23.32: QSPI TXDAT Register Description

Field	Function
31 - 0 TXDAT	**TxFIFO Data** This register provides access to the TxFIFO. On a read access. the last written data will be returned.

Yuntu Microelectronics

23.3.1.28 QSPI RXDAT Register

These registers provide read access to all RxFIFO received data, the first received data locates at RXDAT0 register.

Offset: 80h + n * 4h

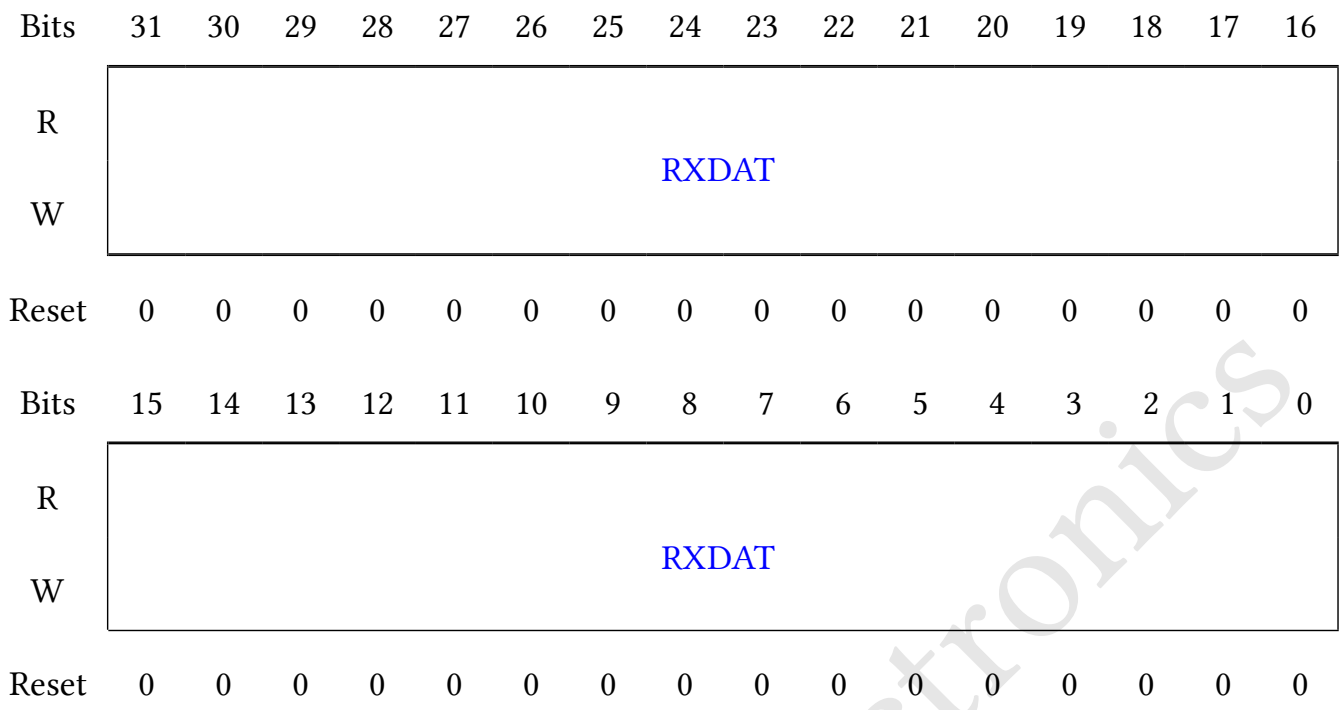


Table 23.33: QSPI RXDAT Register Description

Field	Function
31 - 0 RXDAT	**RxFIFO Data** This register contains the data of RxFIFO.

Yuntu Microelectronics

23.3.1.29 QSPI LUTKEY Register

This register is only used for protecting LOCK register. Writing correct key to this register will close the protection, and access to register except LOCK register will active the protection.

Offset: 100h

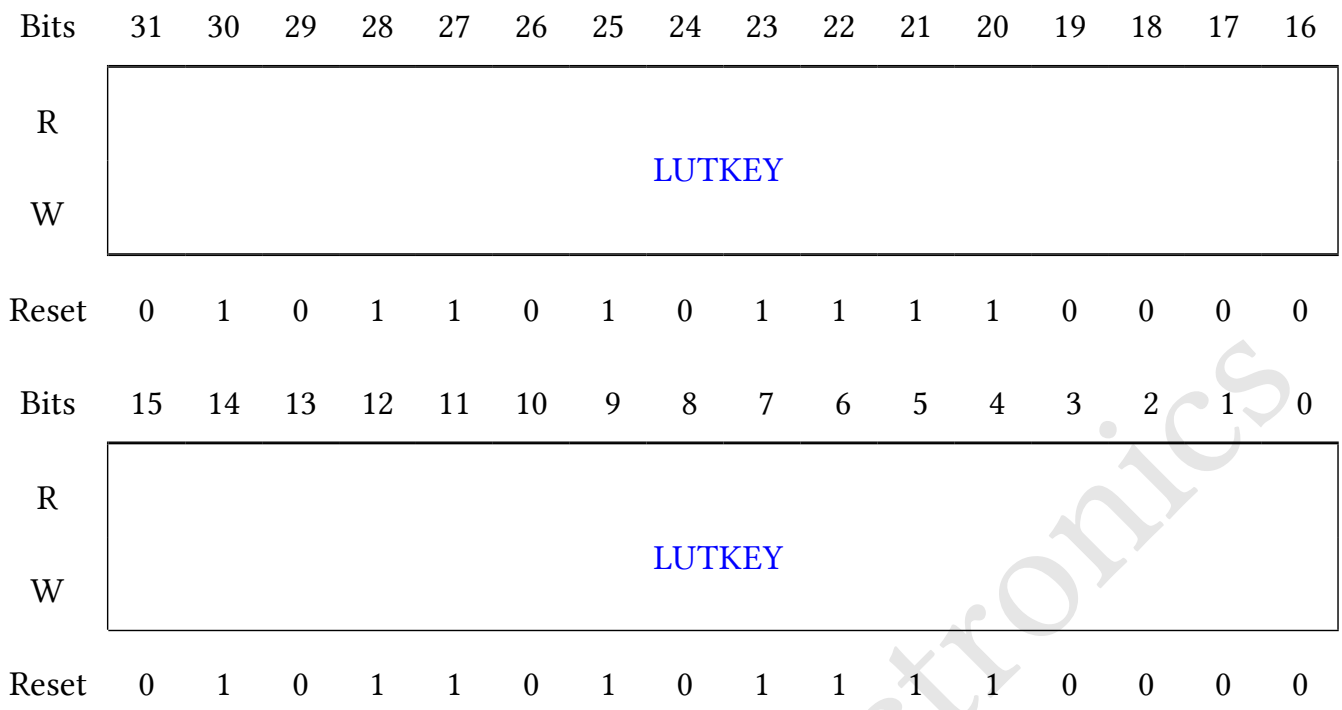


Table 23.34: QSPI LUTKEY Register Description

Field	Function
31 - 0 LUTKEY	**LUT Key** Write this register to lock or unlock the LUT. The key is 0x5AF05AF0 and the read value is always 0x5AF05AF0.

Yuntu Microelectronics

23.3.1.30 QSPI LOCK Register

This register can only access after writing correct key to LUTKEY register.

Offset: 104h

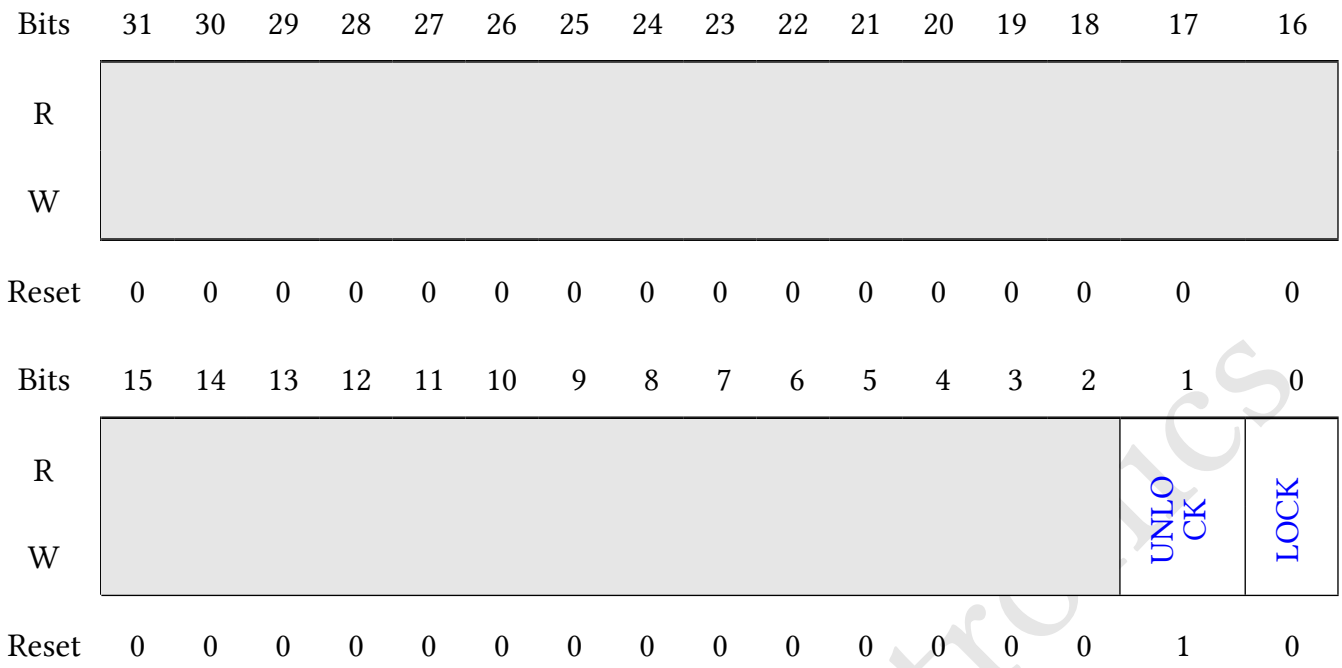


Table 23.35: QSPI LOCK Register Description

Field	Function
1 UNLOCK	**Unlock** Unlock the LUT write access. 0b - Write access to LUT registers is invalid. 1b - Write access to LUT registers is valid.
0 LOCK	**Lock** Lock the LUT write access. 0b - Write access to LUT registers is valid. 1b - Write access to LUT registers is invalid.

Yuntu Microelectronics

23.3.1.31 QSPI LUT Register

LUT0 reset value is 0x08180403, LUT1 reset value is 0x3C002008, other LUTs' reset value are 0x00000000.

Offset: 108h + n * 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPD1							
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPD0							
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Table 23.36: QSPI LUT Register Description

Field	Function
31 - 26 INSTR1	**Instruction 1** Define the instruction that expect this module to execute.
25 - 24 PAD1	**Pad 1** PAD information for INSTR1. 00b - one pad 01b - two pads 10b - four pads 11b - eight pads
23 - 16 OPD1	**Operand 1** Operand for INSTR1.
15 - 10 INSTR0	**Instruction 0** Define the instruction that expect this module to execute.
9 - 8 PAD0	**Pad 0** PAD information for INSTR0. 00b - one pad 01b - two pads 10b - four pads 11b - eight pads
7 - 0	**Operand 0**

Yuntu Microelectronics

Field	Function
OPD0	Operand for INSTR0.

23.4 Function

This section provides the detail functional description of the QSPI module.

The QSPI module outputs clock is from the functional clock, if the functional clock is 100MHz, the output clock is 100MHz in SDR mode and 50MHz in DDR mode.

The bus clock frequency must not less than 1/4 of the functional clock frequency when use this module.

23.4.1 Flash Connection

There are two QSPI interface ports (A port and B port), and each port supports one flash device.

The connection diagram with the 2 devices is as follow:

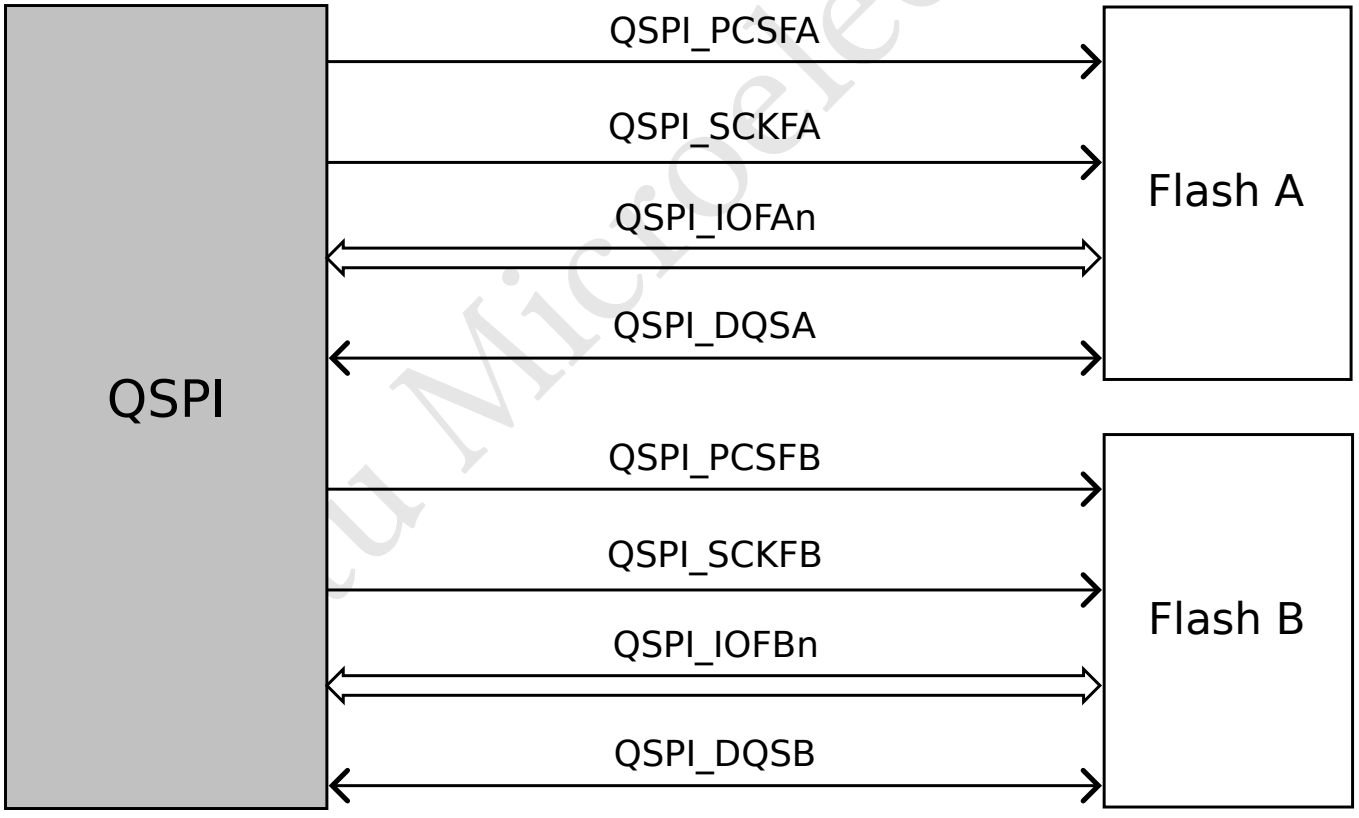


Figure 23.2: Flash Connection Diagram

NOTE:

Port A has 5 high speed IOs that support max to 100MHz frequency, there are SCKFA and IO0FA~IO3FA. Other IOs can only support max to 50MHz frequency. DQSA and DQSB are only used when connect with the HyperBus device.

Yuntu Microelectronics

23.4.2 Flash Access Mode

23.4.2.1 Single/Dual/Quad/Octal Mode

In Single mode, the module transmits/receives data on 1 data pad (IOFx[0] for transmitting, IOFx[1] for receiving).

In Dual mode, the module transmits/receives data on 2 data pads (IOFx[0]~IOFx[1] for both transmitting and receiving).

In Quad mode, the module transmits/receives data on 4 data pads (IOFx[0]~IOFx[3] for both transmitting and receiving).

In Octal mode, the module transmits/receives data on 8 data pads (IOFx[0]~IOFx[7] for both transmitting and receiving).

Single, Dual, Quad and Octal mode is determined by instruction (PADx) in the LUT sequence dynamically.

The unused pad in single/dual mode output value can be defined by MCFG[FxDDO2] and MCFG[FxDDO3].

Yuntu Microelectronics

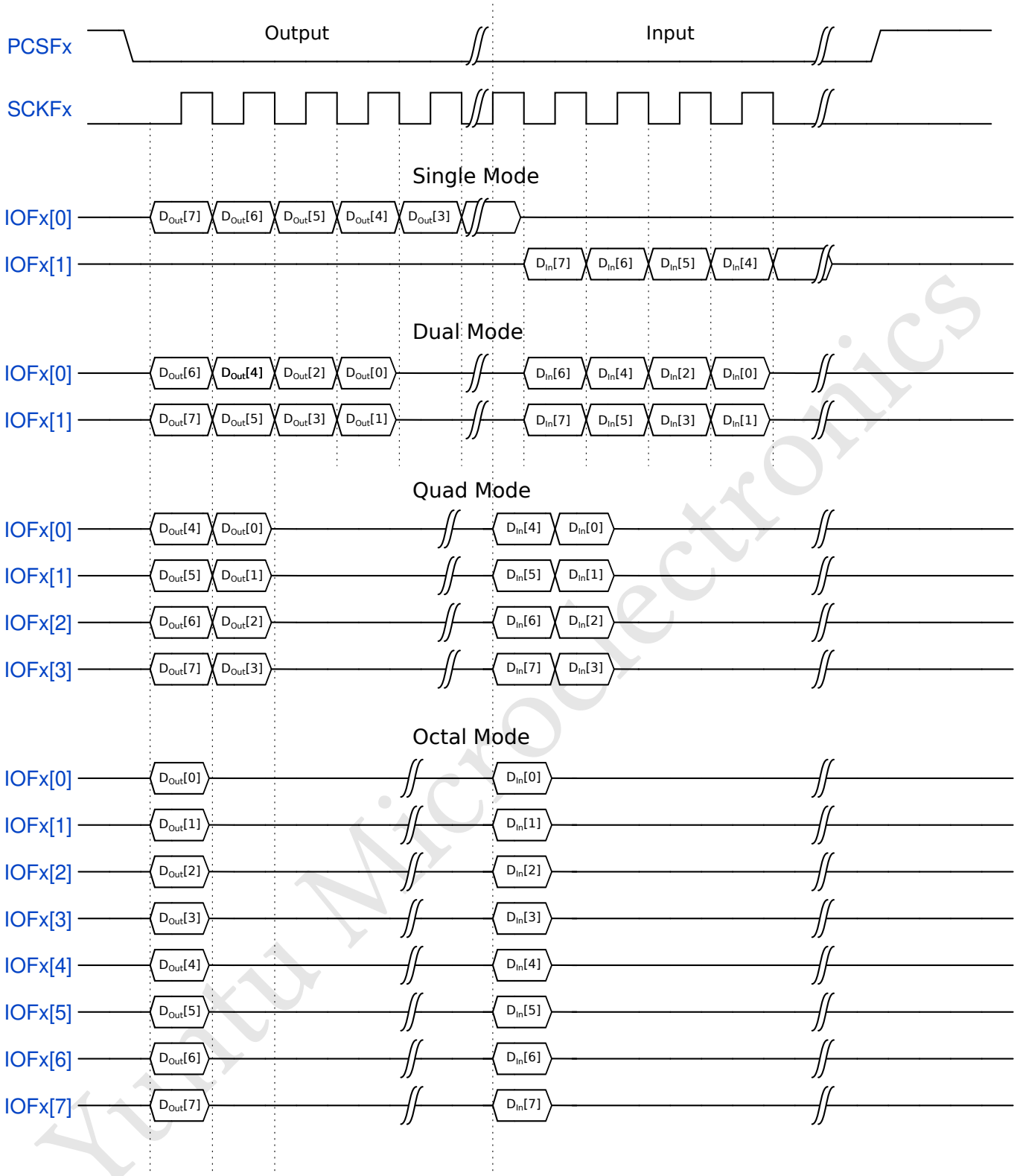


Figure 23.3: Single/Dual/Quad/Octal Mode Timing

23.4.2.2 Individual Mode and Parallel Mode

Individual mode and Parallel mode are determined by the register field APBCFG[PAREN] or AHBCFG[PAREN].

In Individual mode, only one flash device can be accessed at a time.

Yuntu Microelectronics

In Parallel mode, port A and port B can access two flash devices at a time. QSPI will merge/split the flash read/write data automatically. Only read/write data is merged/split. In other cases, the same data information will be transmitted to both port A and port B devices.

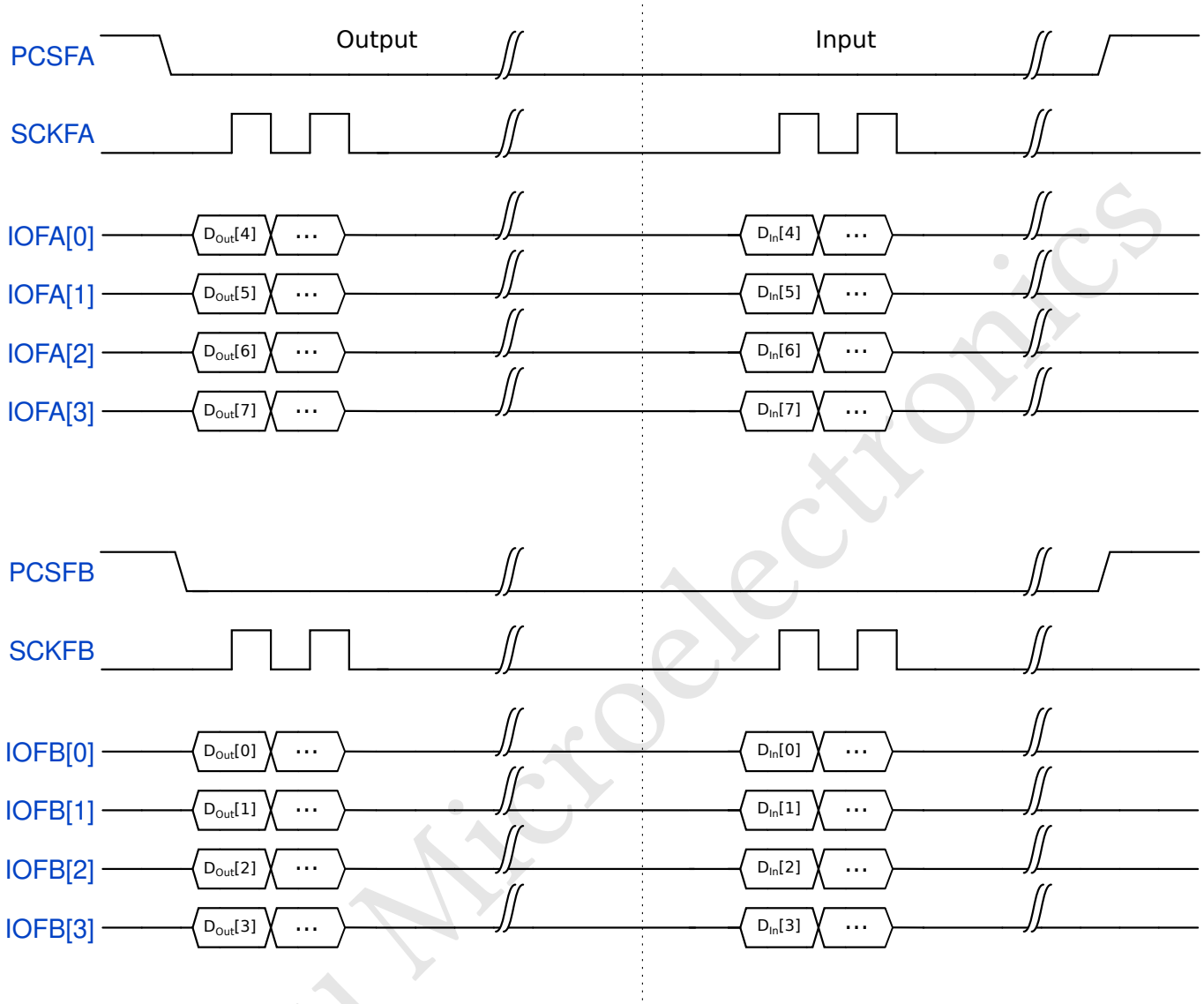


Figure 23.4: Parallel Mode Timing

Yuntu Microelectronics

NOTE:

1. When access flash in Parallel mode, the actual access address is half of the expected address due to the data size has doubled. For example, configure the access address as 0x5, then the actual access address for port A and port B device is 0x2.
2. Parallel mode can't work with Octal mode together.

23.4.2.3 SDR Mode and DDR Mode

In SDR (Single Data transfer Rate) mode, the flash device receives data on SCKFx rise edge and transmits data on SCKFx fall edge.

In DDR (Dual Data transfer Rate) mode, the flash device receives data on both SCKFx rise and fall edges and transmit data on both SCKFx rise and fall edges.

SDR and DDR mode is determined by instruction (INSTRx) in the LUT sequence dynamically.

Setting MCFG[DDREN] to enable the DDR mode.

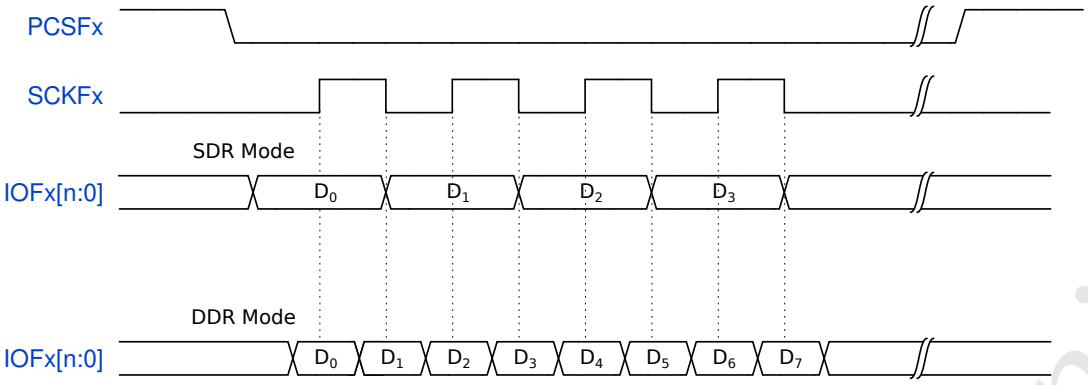


Figure 23.5: SDR/DDR Timing

23.4.2.4 HyperBus Mode

HyperBus mode is used for communication with HyperFlash devices. This mode is enabled/disabled by the register field MCFG[DQSEN].

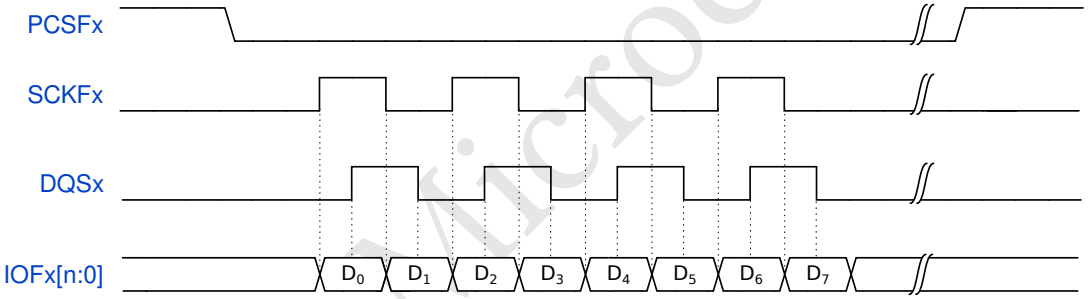


Figure 23.6: HyperBus Timing

23.4.3 Operation Mode

23.4.3.1 Busy Mode

In this mode, all clocks work normally.

Any case of below occurs, the module is under Busy mode:

- DMA is accessing QSPI
- QSPI is accessing flash device (ST[BUSY]=1)
- AHB bus is accessing QSPI
- RxFIFO is not empty
- Any QSPI interrupt is valid

Yuntu Microelectronics

23.4.3.2 Disable Mode

In this mode, AHB Domain and Serial Flash Domain clocks are gated off internally. APB Domain clock works normally in this mode, and registers' access has no limitation. This mode is entered by clearing the register field MEN[EN].

If enter Disable mode during Busy mode, the module will only enter Disable mode until exiting Busy mode.

23.4.3.3 Deepsleep Mode

In this mode, all clocks are gated off internally.

If enter Deepsleep mode during Busy mode, the module will only enter Deepsleep mode until exiting Busy mode.

23.4.3.4 Debug Mode

In this mode, all clocks are gated off internally only when the register field MCFG[DBGDIS] is set.

If enter Debug mode during Busy mode, the module will only enter Debug mode until exiting Busy mode.

23.4.4 Flash Memory Map

Flash memory map in Individual and Parallel mode is as follow:

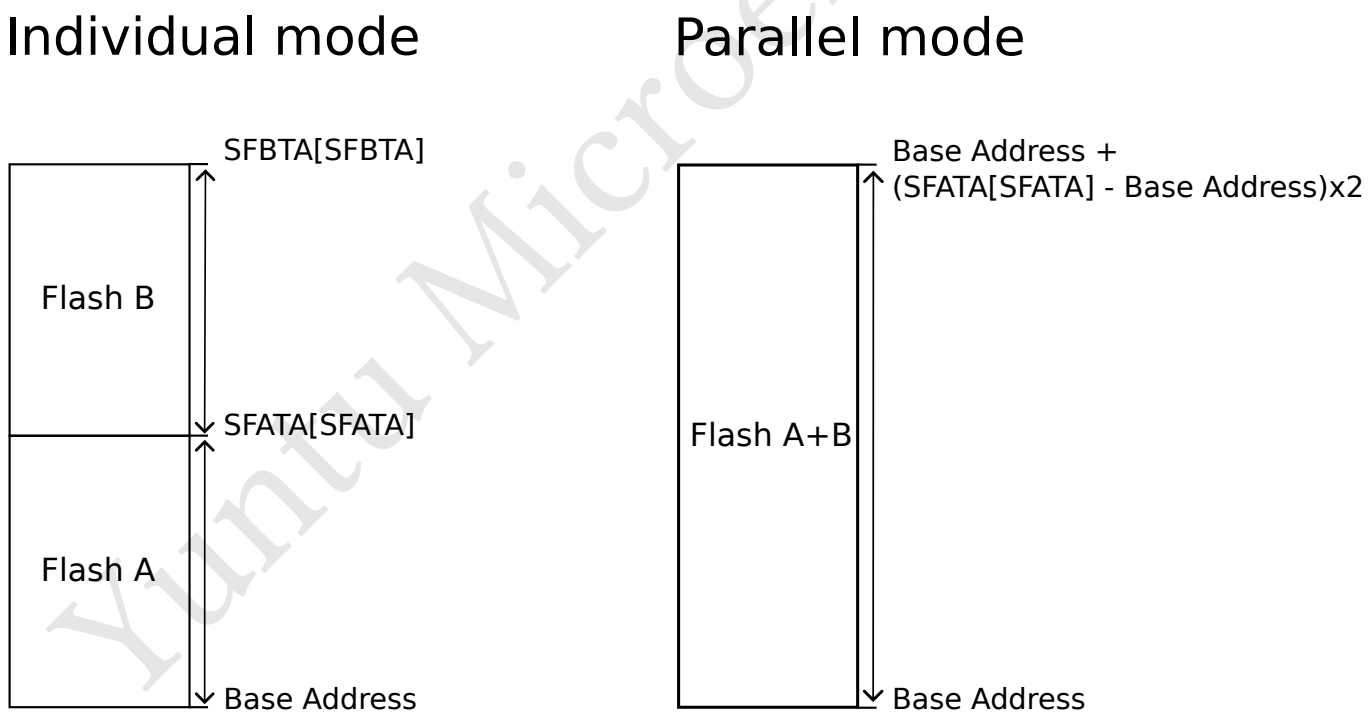


Figure 23.7: Flash Memory Map

The flash base address is 0x68000000 in this chip.

Flash memory map in Individual mode:

- Flash A address range: Base address~(SFATA[SFATA]-1)
- Flash B address range: SFATA[SFATA]~SFBTA[SFBTA]

Yuntu Microelectronics

Flash memory map in Parallel mode:

- Flash A + Flash B address range: Base address~Base address+(SFATA[SFATA]-Base address)*2

NOTE:

1. Value in offset address 0x58 should be same with SFATA register.
2. Value in offset address 0x60 should be same with SFBTA register.

23.4.4.1 Addressing Mode

When the flash device is accessed by the AHB command, the address that sent to the device is (AHB access address-Base address).

When the flash device is accessed by the APB command, the address that sent to the device is (SFA[SFA]-Base address).

The flash address is sent to devices in two parts: row address and column address. For flash devices doesn't support column address, the register field SFACFG[CAS] should keep 0, then all flash address will be sent to devices as row address. For flash devices support column address, the address that sent to the device is composed of row address and column address.

Table 23.37: Flash Access Row/Column Address

Parallel mode	Word addressable	Row Address	Column Address
0	0	FA[31:CAS]	FA[CAS-1:0]
0	1	FA[31:CAS+1]	FA[CAS:1]
1	0	FA[31:CAS+1]	FA[CAS:1]
1	1	FA[31:CAS+2]	FA[CAS+1:2]

NOTE:

1. FA is a flash address with a flash base address removed.
2. CAS is a register field SFACFG[CAS].
3. Word addressable is controlled by the register field SFACFG[WA].

23.4.5 Look Up Table

The LUT (Look Up Table) is an internal memory to preserve a number of preprogrammed sequences. Each sequence consists of up to 10 instructions which are executed sequentially. When a flash access is triggered by an APB command or an AHB command, QSPI controller will fetch the sequence from LUT according to sequence index and execute it to generate a valid flash transaction on interface. LUTs that in sequences will be executed from the low address to the high address. For example, when the sequence 0 is selected, the LUT 0 will be executed first, and the LUT 4 will be executed last. In a LUT, the instruction 0 will be executed before the instruction 1.

Yuntu Microelectronics

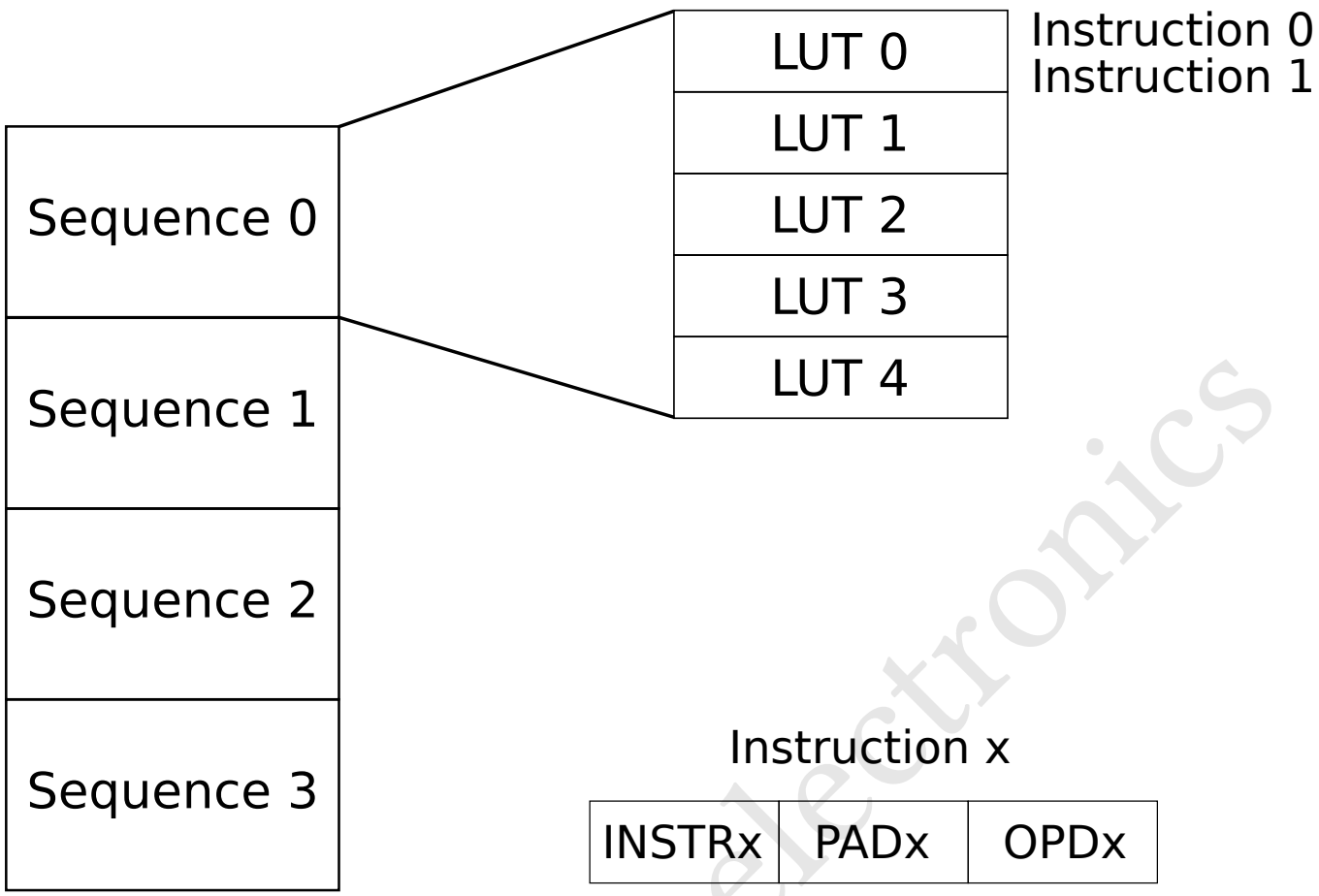


Figure 23.8: LUT Sequence Structure

To communicate with external flash devices, LUTs need to be programmed according to the device requirements. Instructions in LUTs will be sent when a APB or AHB command is triggered.

For APB command, the sequence number is decided by register field APBCFG[SEQID]. For AHB command, the sequence number is decided by register field AHBCFG[SEQID].

23.4.5.1 LUT Lock

The LUTKEY register and LOCK register are applied to protect the LUT contents from being changed by mistakes.

The contents of LUTs can only be changed when register field LOCK[UNLOCK] is 1. Each time when write access to the LOCK register, users need write specified value to LUTKEY register.

23.4.5.2 Programmable Sequence Engine

The QSPI module implements a programable sequence engine that executes the LUT sequence. The follow table is the complete instruction list.

Table 23.38: Instruction Table

Name	Opcode	Action	Transmit Data	Data Number	Pads
STOP	6'h00	Stop execution. Deassert the CS. The next start instruction of LUT sequence is instruction 0.	–	–	Ignore the pad number.

Yuntu Microelectronics

Table 23.38 continued from previous page

Name	Opcode	Action	Transmit Data	Data Number	Pads
CMD_SDR CMD_DDR	6'h01 6'h11	Transmit command code to Flash.	Operand[7:0]	8 bits	2'h0 - one pad 2'h1 - two pads 2'h2 - four pads 2'h3 - eight pads
ADDR_SDR ADDR_DDR	6'h02 6'h12	Transmit row address to Flash.	Row address[31:0]	Operand[7:0] bits	
DUMMY	6'h03	Leave data lines undriven.	–	Operand[7:0] bits, describe the number of serial flash clock cycle.	
CADDR_SDR CADDR_DDR	6'h04 6'h14	Transmit column address to Flash.	Column address[31:0]	Operand[7:0] bits	
MODE_SDR MODE_DDR	6'h05 6'h15	Transmit mode bits to Flash.	Operand[0]	1 bit	
MODE2_SDR MODE2_DDR	6'h06 6'h16	Transmit mode bits to Flash.	Operand[1:0]	2 bits	
MODE4_SDR MODE4_DDR	6'h07 6'h17	Transmit mode bits to Flash.	Operand[3:0]	4 bits	
READ_SDR READ_DDR	6'h08 6'h18	Receive read data from Flash.	–	Byte number that defined by register field (BUFnCFG[DATSZ]/APBCFG[DATSZ]) or operand[7:0].	
WRITE_SDR WRITE_DDR	6'h09 6'h19	Transmit programming data to Flash.	Data in TxFIFO.	Byte number that defined by register field (BUFnCFG[DATSZ]/APBCFG[DATSZ]) or operand[7:0].	
RDTST_SDR RDTST_DDR	6'h0A 6'h1A	Read data test, receive read data from Flash and compare with the expect value (DTPR[DTPR]).	–	32 bits	
SELFTST_SDR SELFTST_DDR	6'h0B 6'h1B	Transmit specified pattern (DTPR[DTPR]) and read back check.	DTPR[DTPR]	32 bits	
JUMP	6'h0F	Stop execution, deassert the CS. Operand[7:0] specifies the next start instruction of LUT sequence.	–	–	Ignore the pad number.

23.4.6 Device Accessing

To access the flash device, you must build the LUT sequence first, get command information from the device datasheet. There are two ways for command transmission, APB command and AHB command.

23.4.6.1 APB Command

For programming operation,

1. Write data into the TxFIFO, the data size must large than 128 bits.
2. Write sequence ID to trigger the LUT sequence transmission (APBCFG[SEQID]).
3. Wait program process done (ST[BUSY]).

For reading operation,

1. Keep enough space in RxFIFO.
2. Write sequence ID to trigger the LUT sequence transmission (APBCFG[SEQID]).
3. Wait program process done (ST[BUSY]).

23.4.6.2 AHB Command

AHB command doesn't support programming operation.

Yuntu Microelectronics

For reading operation,

1. Read access address that located at valid range (0x68000000~0x6FFFFFFF).
2. Wait program process done (ST[BUSY]).

The AHB accessing has a timeout detection function, config AHBCFG[RESPTO] to define the timeout scope.

23.4.7 AHB Buffers

To reduce the latency of the reads for AHB masters, the data read from serial flash memory are stored into AHB buffers. There is a 256 bytes ram space that divided into four buffers, the size of each buffer is configurable with the minimum size being 0 bytes and maximum size being 256 bytes. The size of buffer 0 ranges from 0 to BUF0IND[TPIND0]-1. The size of buffer 1 ranges from BUF0IND[TPIND0] to BUF1IND[TPIND1]-1, buffer 2 ranges from BUF1IND[TPIND1] to BUF2IND[TPIND2]-1, buffer 3 ranges from BUF2IND[TPIND2] to the size of the complete buffer (256 bytes).

Each buffer can only be accessed with a specified AHB master that is defined by BUFnCFG[MSTRID]. When defined AHB master read access the serial flash device, the read data will be stored into the corresponding buffer. Optionally, buffer 3 can be configured as a all master accessible buffer when BUF3CFG[ALLACC] set.

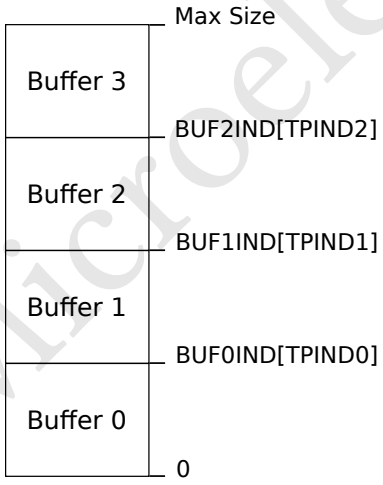


Figure 23.9: AHB Buffer Structure

23.4.7.1 Prefetch

When access to the serial flash device by AHB bus, the module will trigger a read sequence to get the flash data, then the data located in aimed address will return to the bus at once when the module receives it. The reading accessing won't stop until the read data number reaches the defined size, data in adjacent address will be stored into the accessing buffer. The next time when AHB bus access the serial flash device space, the data will be returned from the buffer directly if it has been prefetched. Otherwise, a new reading access will start.

23.4.7.1.1 Prefetch Abort Mechanism

- High priority accessing

Yuntu Microelectronics

When buffer 0 high priority accession is enabled (BUF0CFG[HPEN]=1), other buffers' prefetch accessing to the serial flash device can be suspended by buffer 0 accessing. After buffer 0 accessing finished, the suspended accessing will be recovered.

- Same master ID accessing

Any ongoing read transaction will be aborted if a request from the same master ID arrives for a new address.

The information of the suspended communication is record by the SPNDST register.

23.4.7.2 AHB Burst Support

This module supports AHB burst transaction. Four, eight, and sixteen beat bursts are supported and the bursts might either be incrementing or wrapping, and it supports 8-, 16-, 32-, and 64-bit data sizes. If the total burst size is more than the data prefetch size, an error response is generated and the error flag will be set (IF[AHBBSE]). The data prefetch size can be defined by BUFxCFG[DATSZ] or data size mentioned in the LUT OPDx field when BUFxCFG[DATSZ] is programmed as 0.

23.4.8 Flash Device Programming

The QSPI module can only support programming the flash device by APB command, any programming access to the flash device by AHB bus will result an error. Writing program data into TXDAT register, and the data size must not less than 128 bits. Building a LUT program sequence to communicate with the flash device and program the TxFIFO data. The program data size is decided by the LUT operand or APBCFG[DATSZ].

The TxFIFO supports generating DMA request by watermark (TBCS[WM]).

23.4.9 Flash Device Reading

The QSPI module supports reading the flash device by APB or AHB command. If read the flash device by APB command, the read back data are stored into the RxFIFO, access them by RXDAT register. If read the flash device by AHB command, the read back data are stored into in AHB buffers. Specially, AHB bus is able to access the RxFIFO by address mapping directly, the valid address ranges from 0x67000000 to 0x67000080. The read data size is decided by the LUT operand or APBCFG[DATSZ]/BUFxCFG[DATSZ].

The RxFIFO supports generating DMA request by watermark (RBCS[WM]).

The QSPI module will stop the communication clock when RxFIFO is full and reading access to the flash device is ongoing, set MCFG[CLKSPEN] to enable this function. The communication will recover when the RxFIFO is not full.

23.4.10 Error

This module will generate a bus error if use it in wrong way, there are some error cases:

- Trigger a LUT program sequence by AHB
- AHB burst size is larger than the buffer empty size
- AHB burst size is larger than the request data size
- AHB access an invalid address
- AHB accessing timeout

Yuntu Microelectronics

23.4.11 Input Sample Configuration

In order to collect the correct input data, you may need to change the input sample configuration. It's recommended to use the loopback sample mode and data test instruction together for maximum convenience. If change the input sample configuration by manual, add the sample delay number (SMP[SMPDLY]) if the device output data behinds the clock too much; change the sample offset (SMP[DDRHF]) in DDR mode if sampled data has bit shift; invert the sample edge (SMP[SMPEDG]) in SDR mode if sampled data has bit shift.

23.4.11.1 Loopback Sample Mode

Recommend using the loopback sample mode, we can ignore the pad delay in this mode and only take care of the device delay, set SMP[LBPEN] to enable the loopback sample mode. Usually it's not necessary config the sample delay (SMP[SMPDLY]), sample offset (SMP[DDRHF]) and sample edge (SMP[SMPEDG]), they are automatically adjusted if data test executed successfully.

When the QSPI module communicates with HyperBus devices, the loopback sample mode should be disabled (SMP[LBPEN]=0).

A configurable delay chain is applied for the sample clock. There are 31 delay cell in total, each cell has delay about 180~540ps, config SMP[CLKDLL] to select the number of delay cell. Please change this configuration according to data output timing of the flash device.

You can use the delay chain in both loopback sample mode and HyperBus mode.

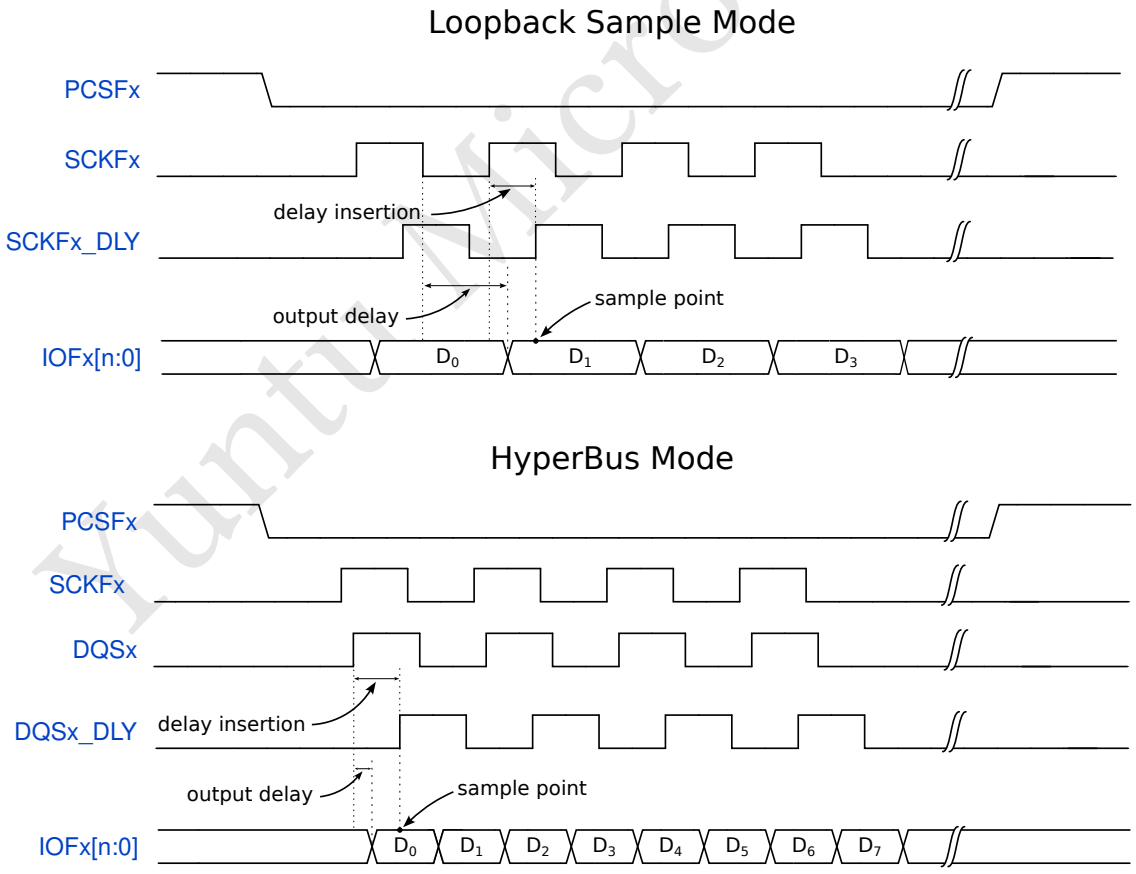


Figure 23.10: Delay Cell Insertion

Yuntu Microelectronics

23.4.11.2 Normal Sample Mode

Only use this mode when loopback sample mode is useless. Disable the loopback sample mode (SMP[LPEN]=0) and data test mode (SMP[DTEN]=0), change the sample delay (SMP[SMPDLY]), sample offset (SMP[DDRHF]) and sample edge (SMP[SMPEDG]) by manual, try the communication again.

23.4.12 Data-test Mechanism

The data-test mechanism is designed to adapt different data output delay from various flash devices. It's suggested to conduct the RDTST instruction before using the flash device. The following is the usage process of the RDTST instruction:

1. Data test must use with the loopback sample mode, enable loopback sample mdoe (SMP[LPBKEN]=1) and data test mode (SMP[DTEN]=1)
2. Execute data test instruction
   - Write the DTPR register with known flash data, select a complexity data as possible, such as 0x12345678
   - Build the read LUT sequence, use RDTST instruction instead the read instruction
   - Trigger the LUT sequence transmission
   - Wait communication done (ST[BUSY]=0)
3. Check data test result, if ST[DTRTY]=1, execute data test instruction again
4. Check data test result, if ST[DTFAIL]=1, data test fail and need disable data test and apply manual operation method

If there is no right data for data test reading, it's suggested to program a self-defined pattern instead first and then execute the data test instruction.

Once the data test instruction is successfully executed, please keep setting SMP[DTEN] during normal use.

23.4.13 Communication Example

The following is an example that communicate with the serial NOR flash device (Winbond W25Q80DV) in one line SDR mode.

1. Data test
   sequence: CMD(0xB)+ADDR(0x18)+DUMMY(0x8)+DATA_TEST(0x2)+JUMP(0x0)
2. Flash write enable
   sequence: CMD(0x6)+JUMP(0x0)
3. Flash chip erase
   sequence: CMD(0xC7)+JUMP(0x0)
4. Read status register
   sequence: CMD(0x5)+READ(0x1)+JUMP(0x0)
5. Volatile SR write enable
   sequence: CMD(0x50)+JUMP(0x0)
6. Write status register
   sequence: CMD(0x1)+MODE(0x0)+MODE(0x2)+JUMP(0x0)
7. Write enable
   sequence: CMD(0x6)+JUMP(0x0)
8. Write data
   sequence: CMD(0x32)+ADDR(0x18)+WRITE(0x2)+JUMP(0x0)

Yuntu Microelectronics

9. Read check
   sequence: CMD(0x6B)+ADDR(0x18)+DUMMY(0x8)+READ(0x2)+JUMP(0x0)

23.5 Glossary

QSPI Quad Serial Peripheral Interface

Yuntu Microelectronics

Chapter 24

Inter-Integrated Circuit (I2C)

24.1 Chip-specific I2C Information

24.1.1 Instantiation Information

YTM32B1HA0x has 5 I2C instances.

24.1.2 Signal Description

Table 24.1: I2C Signal Table

Signal	Description	Direction
SCL	I2C clock line	IO
SDA	I2C data line	IO

24.1.3 Clock Description

Table 24.2: I2C Clock Table

Clock Name	Description
APB Clock	Clock for master and slave register access
Function Clock	Main clock for I2C master and slave logic, when using clock enabled in sleep mode, I2C could still working under sleep mode
External Clock	Only valid in slave mode, I2C slave can work with external SCL clock even main function clock disabled. **NOTE**. Filter and sample configuration is disable in external clock mode.

24.1.4 Reset Description

Yuntu Microelectronics

Table 24.3: I2C Reset Table

Reset Name	Description
Chip reset	The logic and registers for the I2C master and slave are reset to their default state.
FIFO reset	Setting the TXFIFO[RESET] bit and RXFIFO[RESET] bit will reset the TXFIFO and RXFIFO respectively. After a FIFO is reset, the FIFO is empty.

24.1.5 Power Mode Description

Table 24.4: I2C Power Mode Table

Power Mode	Description
Active	Full function
Sleep	• Keeps working when function clock available • Slave working when function clock available or only external clock available
Deep Sleep	• Keeps working when function clock available • Slave working when function clock available or only external clock available

24.2 Introduction

24.2.1 Features

Features of the I2C module include:

- Support standard, fast and ultra fast mode
- Support 7-bit/10-bit address mode with master and slave
- Support SMBus mode
- Support multi-master arbitration and synchronization
- Support Master and slave clock stretching
- Transmit/Receive FIFO (Master only)
- Analog and digital filter on both SCL and SDA pins
- Support Transmit/Receive via DMA

24.3 Register Definition

24.3.1 I2C Memory Map

I2C0 base address 40094000h
I2C1 base address 40098000h
I2C2 base address 4009c000h
I2C3 base address 400a0000h
I2C4 base address 400a4000h

Yuntu Microelectronics

Table 24.5: I2C Memory Map

Offset	Register	Width	Access	Reset Value
0h	Master Control Register (MCTRL)	32	RW	00000000h
4h	Master Clock Configuration Register (MCLKCFG)	32	RW	00000000h
8h	Master Clock Filter Configuration and Divider Register (MFLTCFG)	32	RW	00000000h
ch	Master Status Register (MSTS)	32	RW	00000002h
10h	Master Interrupt Register (MIE)	32	RW	00000000h
14h	Master TX FIFO Control and Status Register (TXFIFO)	32	RW	02000000h
18h	Master RX FIFO Control and Status Register (RXFIFO)	32	RW	02000000h
1ch	Master Data Register (MDATA)	32	RW	00010000h
20h	Data Match Register (MATCH)	32	RW	00000000h
24h	Timeout Configure Register (TOCFG)	32	RW	00000000h
28h	Master Clock Configuration Under High Speed Mode Register (HSCLK)	32	RW	00000000h
100h	Slave Control Register (SCTRL)	32	RW	00000000h
104h	Slave Clock Configuration Register (SCLKCFG)	32	RW	00000000h
108h	Slave Filter Configuration Register (SFLTCFG)	32	RW	00000000h
10ch	Slave Status Register (SSTS)	32	RW	00000000h
110h	Slave Interrupt Enable Register (SIE)	32	RW	00000000h
114h	Slave Data Register (SDATA)	32	RW	00010000h
118h	Slave Ack Register (SACK)	32	RW	00000000h
11ch	Slave Address Match Configuration Register (SADDR)	32	RW	00000000h
120h	Slave Received Address Register (SRADDR)	32	RW	00000000h

24.3.1.1 I2C MCTRL Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W			TXDMAEN	RXDMAEN			RDMO							MATCFG		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IGACK	ASTOP					HSMOD	PPEN								MEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 24.6: I2C MCTRL Register Description

Field	Function
29 TXDMAEN	**Master TX DATA DMA Enable** TX DATA request DMA enable when TX fifo less than watermark or buff empty 0b - DMA request disabled 1b - DMA request enabled
28 RXDMAEN	**Master RX Data DMA Enable** RX DATA request DMA enable when RX fifo great than watermark or buff full 0b - DMA request disabled 1b - DMA request enabled
25 RDMO	**Receive Data Match Only** Setting this bit will set I2C only stores data after received matched match configuration 0b - I2C will store all received data 1b - I2C will only store data after data matched
18 - 16 MATCFG	**Match Configuration** I2C match feature configuration 000b - Match disabled 010b - First data match MATCH0 or MATCH1 011b - Any data match MATCH0 or MATCH1 100b - First data match MATCH0 and second data match MATCH1 101b - Any data match MATCH0 and next data match MATCH1 110b - First data byte AND with MATCH1 match MATCH0 111b - Any data byte AND with MATCH1 match MATCH0
15 IGACK	**Ignore I2C ACK** Setting this bit will let I2C generate internal ACK even if NACK received. **NOTE**. This bit should be set in Ultra-Fast mode. 0b - I2C will receive ACK and NACK normally 1b - I2C will treat received NACK as ACK
14 ASTOP	**Automatically STOP** Setting this bit will let I2C automatically generate STOP condition when TX FIFO empty 0b - I2C master generated STOP with STOP command 1b - I2C master generated STOP with STOP command or TX FIFO is empty
9 HSMOD	**High Speed Mode Enable** Setting this bit will let I2C working under High-Speed mode(only support data output) 0b - I2C under normal mode 1b - I2C in High-Speed output only mode
8	**Push Pull Mode Enable**

Yuntu Microelectronics

Field	Function
PPEN	Setting this bit will let I2C SCL/SDA to push-pull mode 0b - SCL/SDA in open drain mode 1b - SCL/SDA in push pull mode
0 MEN	**Master Mode Enable** Control all I2C master logic 0b - Master mode is disabled 1b - Master mode is enabled

24.3.1.2 I2C MCLKCFG Register

MCLK register not support update when master logic is enabled

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W			HIGH								LOW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W			VALID								STHD					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24.7: I2C MCLKCFG Register Description

Field	Function
29 - 24 HIGH	**SCL High Period** SCL clock high period is (HIGH + 1) * divided clock period
21 - 16 LOW	**SCL Low Period** SCL clock low period is (LOW + 1) * divided clock period
13 - 8 VALID	**SDA Data Valid Period** SDA data valid period is (VALID + 1) * divided clock period
5 - 0 STHD	**SDA Data Setup/Hold Period** SDA data setup/hold period is (STHD + 1) * divided clock period

24.3.1.3 I2C MFLTCFG Register

FLTDIV register not support update when master logic is enabled

Yuntu Microelectronics

Offset: 8h

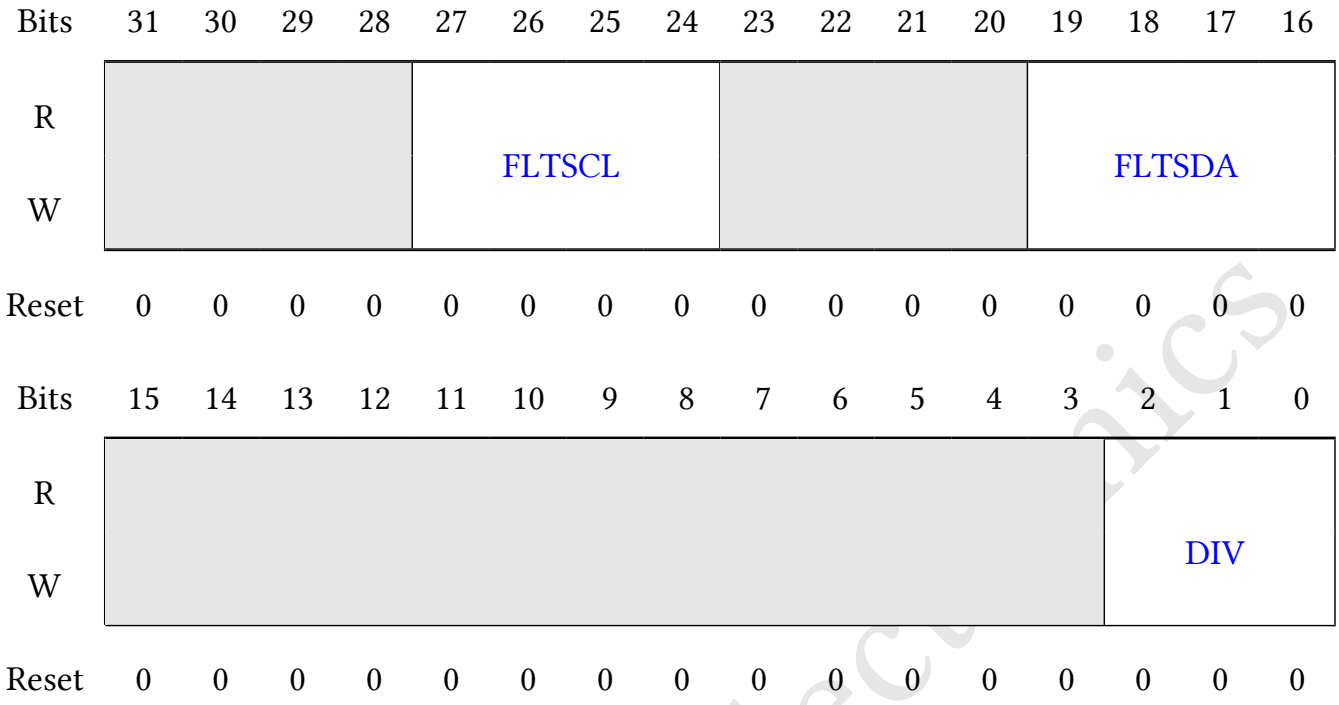


Table 24.8: I2C MFLTCFG Register Description

Field	Function
27 - 24 FLTSCL	**SCL Filter Control** Control digital filter on SCL line SCL digital filter use undivided I2C function clock Noise less or equal FLTSCL * function period will be filtered
19 - 16 FLTSDA	**SDA Filter Control** Control digital filter on SDA line SDA digital filter use undivided I2C function clock Noise less or equal FLTSDA * function period will be filtered
2 - 0 DIV	**Function Clock Divider** Control I2C master logic clock divider 000b - I2C master clock divided by 1 001b - I2C master clock divided by 2 010b - I2C master clock divided by 4 011b - I2C master clock divided by 8 100b - I2C master clock divided by 16 101b - I2C master clock divided by 32 110b - I2C master clock divided by 64 111b - I2C master clock divided by 128

Yuntu Microelectronics

24.3.1.4 I2C MSTS Register

Offset: ch

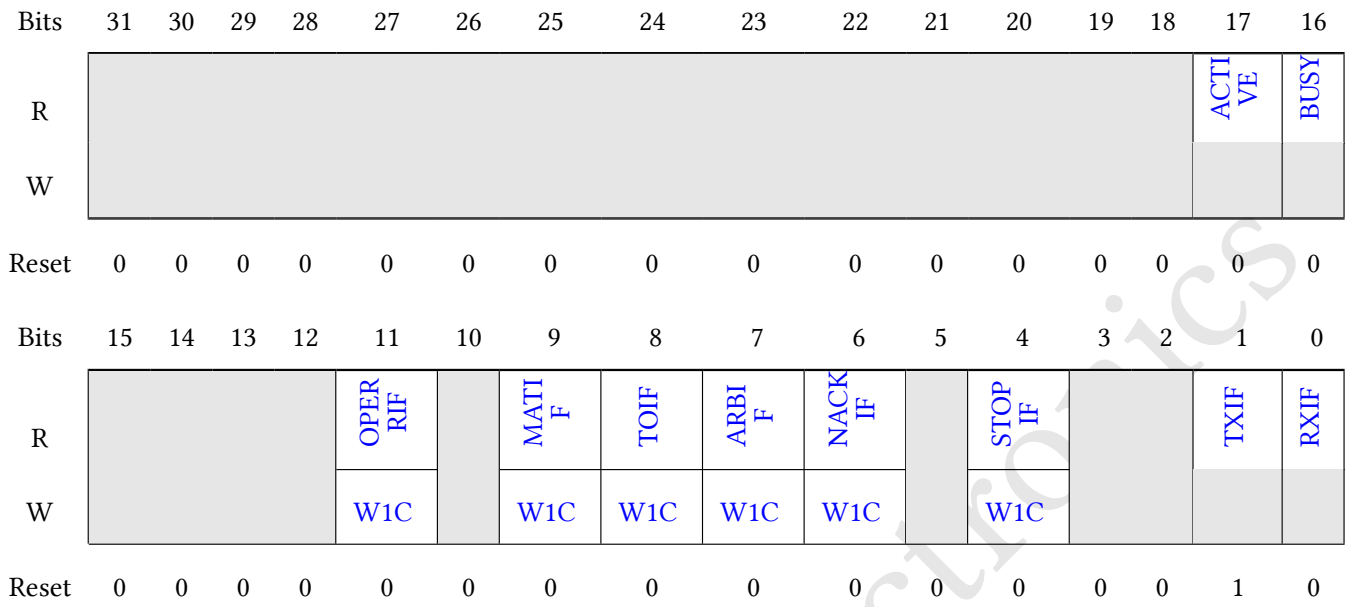


Table 24.9: I2C MSTS Register Description

Field	Function
17 ACTIVE	**I2C Master Active** I2C master is transmitting or receiving data 0b - I2C master is idle 1b - I2C master is busy
16 BUSY	**I2C Bus Busy** I2C Bus is transmitting or receiving data 0b - I2C bus is idle 1b - I2C bus is busy
11 OPERRIF	**Operation Error Interrupt Flag** I2C master operation error, this bit is set when TX FIFO is empty and no automatic STOP generated. User have to clear this bit before start a new data transfer
9 MATIF	**Data Match Interrupt Flag** Indicates I2C master received data matched data match configuration 0b - Received data not matched 1b - Received data matched
8	**Timeout Interrupt Flag**

Yuntu Microelectronics

Field	Function
TOIF	This bit set when line low timeout detection enable and SCL/SDA low timeout This bit only cleared when SCL/SDA line return to 1 User have to solve pin low error before start a new data transfer 0b - Line low timeout not detected 1b - Line low timeout detected
7 ARBIF	**Arbitration Lost Interrupt Flag** This bit is set when Master send out logic 1 but detect logic 0 on SCL or SDA line Master detected START or STOP condition on bus User have to clear this bit before start a new data transfer 0b - I2C master working normal 1b - I2C master lost arbitration
6 NACKIF	**NACK Received Interrupt Flag** If ACK detected logic is enable (IGACK is cleared), I2C master will send out STOP and terminate current transfer, NACKIF bit will set. 0b - I2C master working normal 1b - I2C master detected NACK
4 STOPIF	**STOP Received Interrupt Flag** Set when I2C master send out a STOP condition successfully 0b - I2C master not generated STOP 1b - I2C master generated STOP
1 TXIF	**TX Request Flag** Indicates words in TX FIFO is equal or less than TX FIFO watermark setting 0b - Word in TX FIFO is greater than watermark 1b - Word in TX FIFO is equal or less than watermark, request to write new data
0 RXIF	**RX Request Flag** Indicates words in RX FIFO is greater than RX FIFO watermark setting 0b - Word in RX FIFO is equal or less than watermark 1b - Word in RX FIFO is greater than watermark, should read data register

Yuntu Microelectronics

24.3.1.5 I2C MIE Register

Offset: 10h

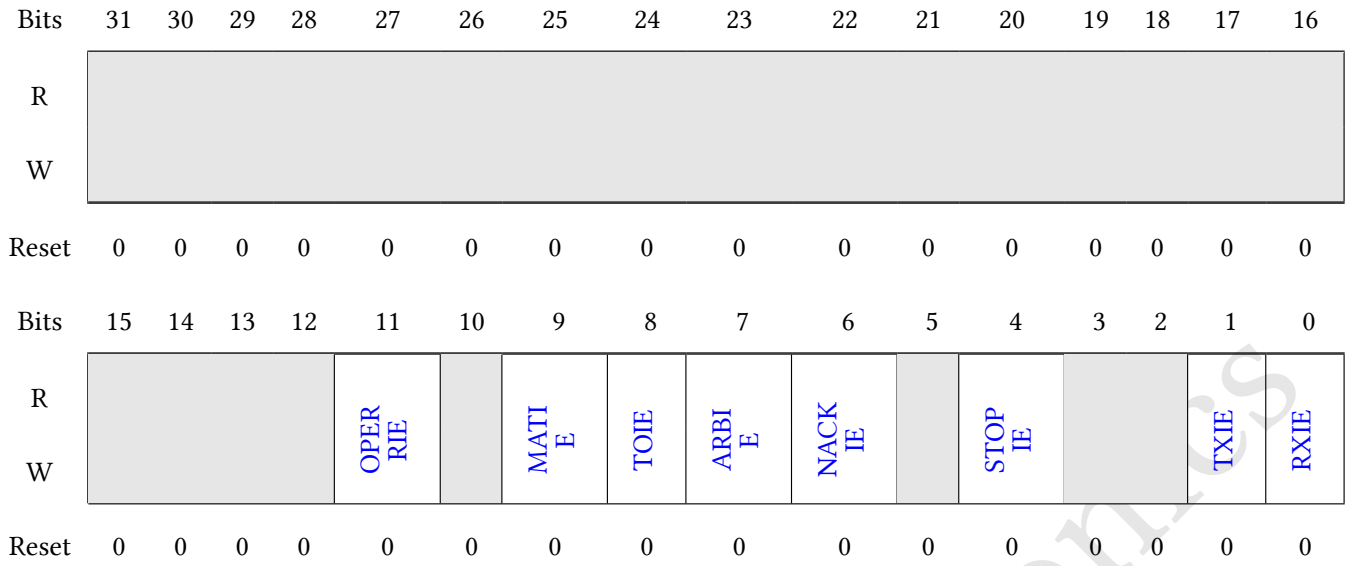


Table 24.10: I2C MIE Register Description

Field	Function
11 OPERRIE	**Operation Error Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
9 MATIE	**Data Match Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
8 TOIE	**Timeout Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
7 ARBIE	**Arbitration Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
6 NACKIE	**NACK Detected Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
4 STOPIE	**STOP Detected Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
1 TXIE	**TX DATA Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
0	**RX DATA Interrupt Enable**

Yuntu Microelectronics

Field	Function
RXIE	0b - Interrupt disabled 1b - Interrupt enabled

24.3.1.6 I2C TXFIFO Register

Offset: 14h

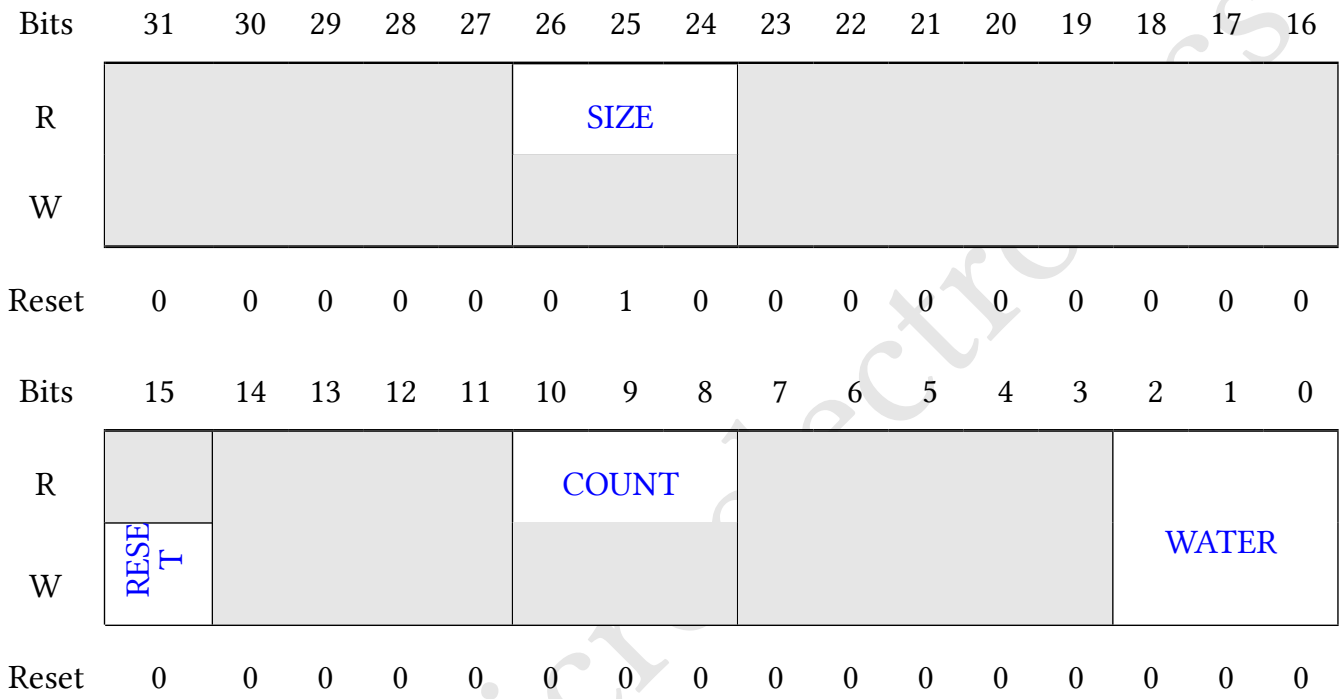


Table 24.11: I2C TXFIFO Register Description

Field	Function
26 - 24 SIZE	**I2C Master TX FIFO Size** Available TX FIFO Size is 2^SIZE words
15 RESET	**I2C Master TX FIFO Reset** Reset TX FIFO address pointer
10 - 8 COUNT	**Word Count in TX FIFO** I2C Master TX FIFO current data count
2 - 0 WATER	**TX FIFO Watermark** TXIF set when words count in TX FIFO equal or less than watermark

24.3.1.7 I2C RXFIFO Register

Offset: 18h

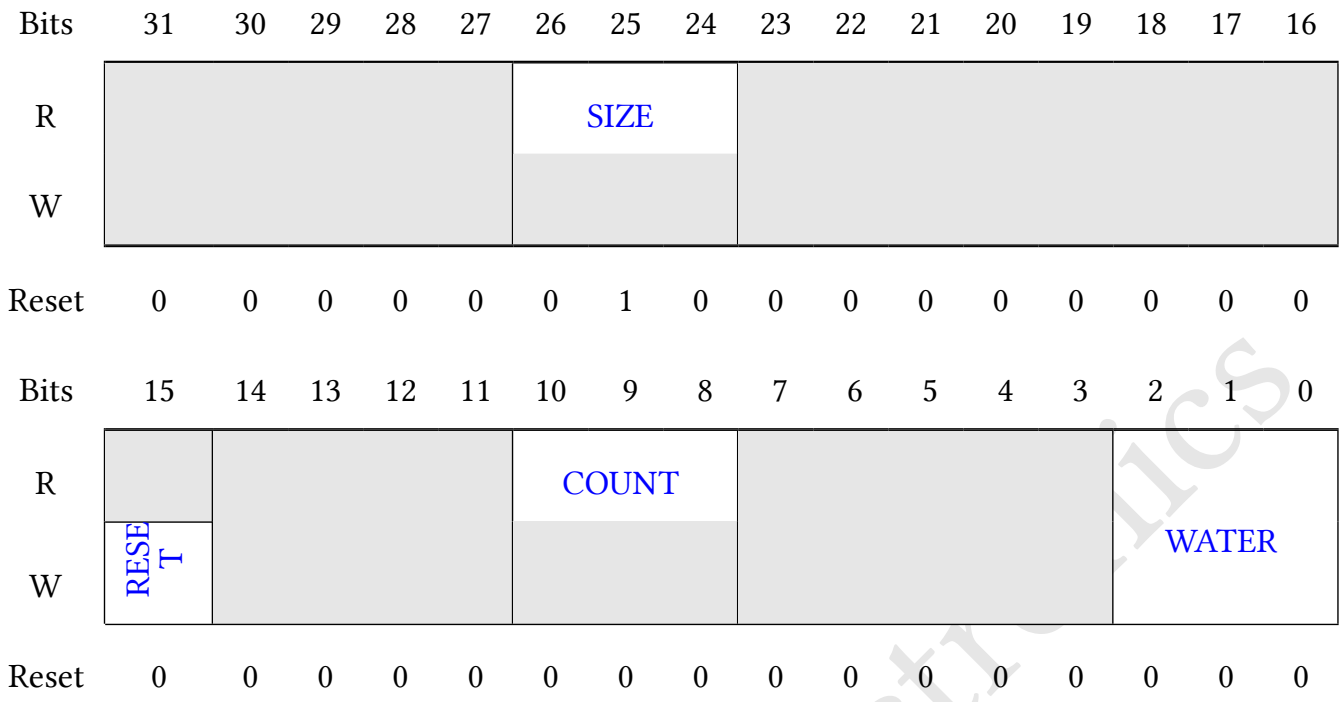


Table 24.12: I2C RXFIFO Register Description

Field	Function
26 - 24 SIZE	**I2C Master RX FIFO Size** Available RX FIFO Size is 2^SIZE words
15 RESET	**I2C Master RX FIFO Reset** Reset I2C master RX FIFO address pointer
10 - 8 COUNT	**Word Count in TX FIFO** Current data count in RX FIFO
2 - 0 WATER	**RX FIFO Watermark** RXIF set when words count in RX FIFO greater than watermark

Yuntu Microelectronics

24.3.1.8 I2C MDATA Register

Write MDATA will let I2C master send out address/data/repeat start/stop to I2C bus and read MDATA will return I2C master received data and status.

Offset: 1ch

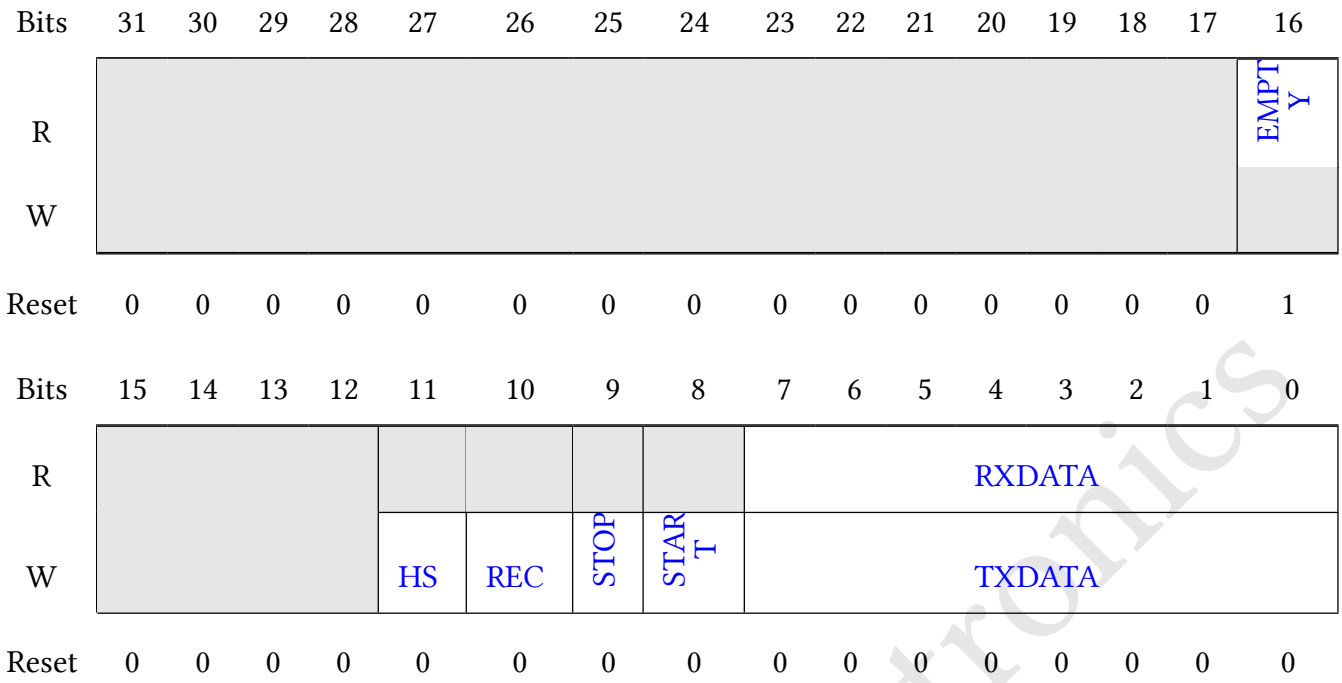


Table 24.13: I2C MDATA Register Description

Field	Function
16 EMPTY	**RX Data Empty** RX FIFO or buffer is empty
11 HS	**High Speed Mode** Setting this bit will let I2C master send data in high speed mode
10 REC	**Receive Data** Setting this bit with target bytes count in DATA field will let I2C receive (DATA + 1) bytes data
9 STOP	**Send STOP**
8 START	**Generate START/repeat START** Setting this bit with slave address in DATA field will let I2C master generate START/repeat START to given slave
7 - 0 DATA	**TX RX DATA** Write DATA will let I2C master send out data or address Read DATA will return I2C master received data in FIFO or buffer

24.3.1.9 I2C MATCH Register

Offset: 20h

Yuntu Microelectronics

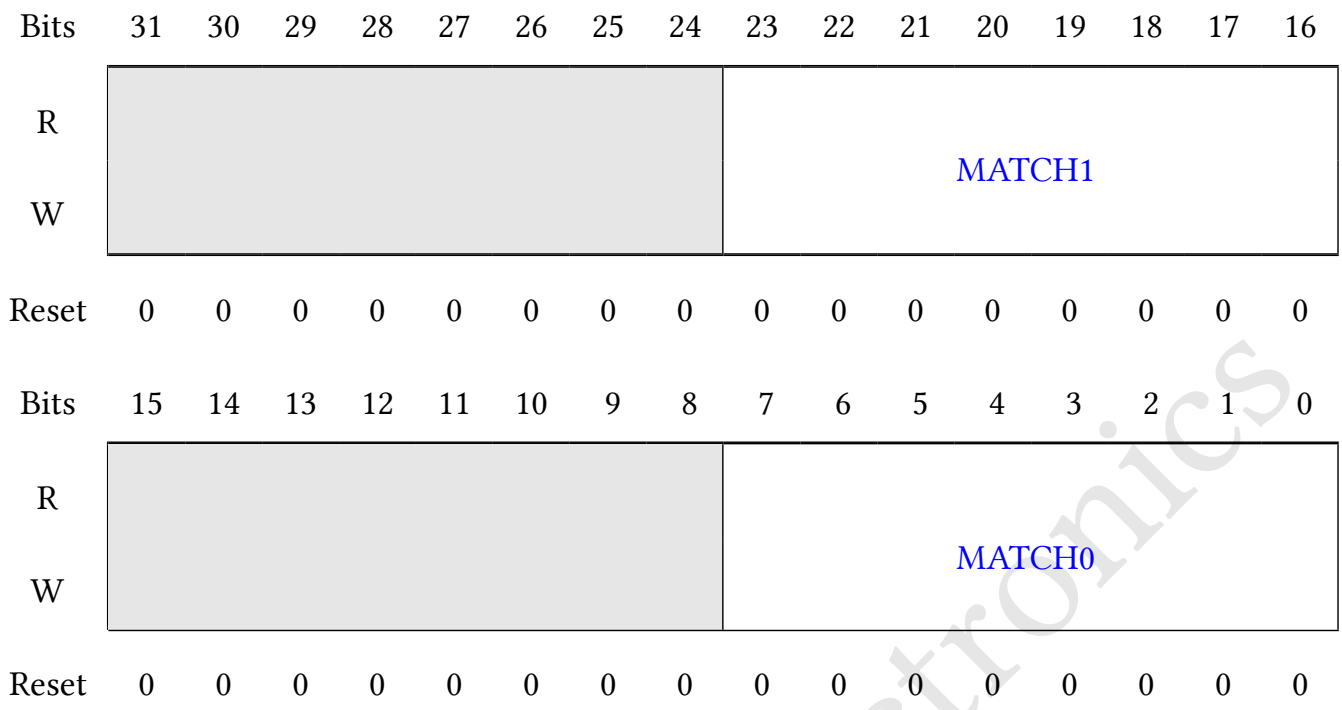


Table 24.14: I2C MATCH Register Description

Field	Function
23 - 16 MATCH1	**Match Data 1**
7 - 0 MATCH0	**Match Data 0**

Yuntu Microelectronics

24.3.1.10 I2C TOCFG Register

Offset: 24h

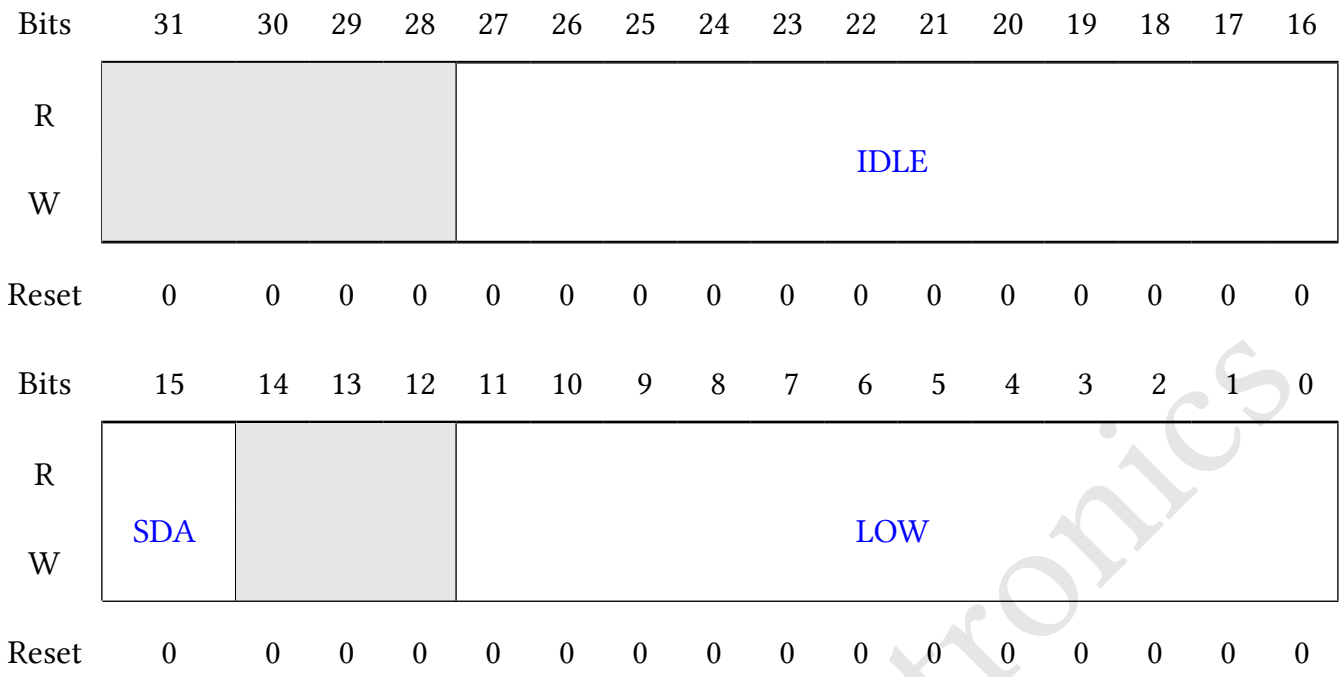


Table 24.15: I2C TOCFG Register Description

Field	Function
27 - 16 IDLE	**Line Idle Timeout Control** Line idle flag set when SDA and SCL line high for (IDLE) * divided function clock, after line idle, master can send out START.
15 SDA	**SDA Line Low Detect** Set this bit to enable SDA line low detection 0b - Line low only detect SCL 1b - Line low detect SDA and SCL
11 - 0 LOW	**Line Low Timeout Control** Line low period is (LOW * 256) * divided function clock

Yuntu Microelectronics

24.3.1.11 I2C HSCLK Register

Offset: 28h

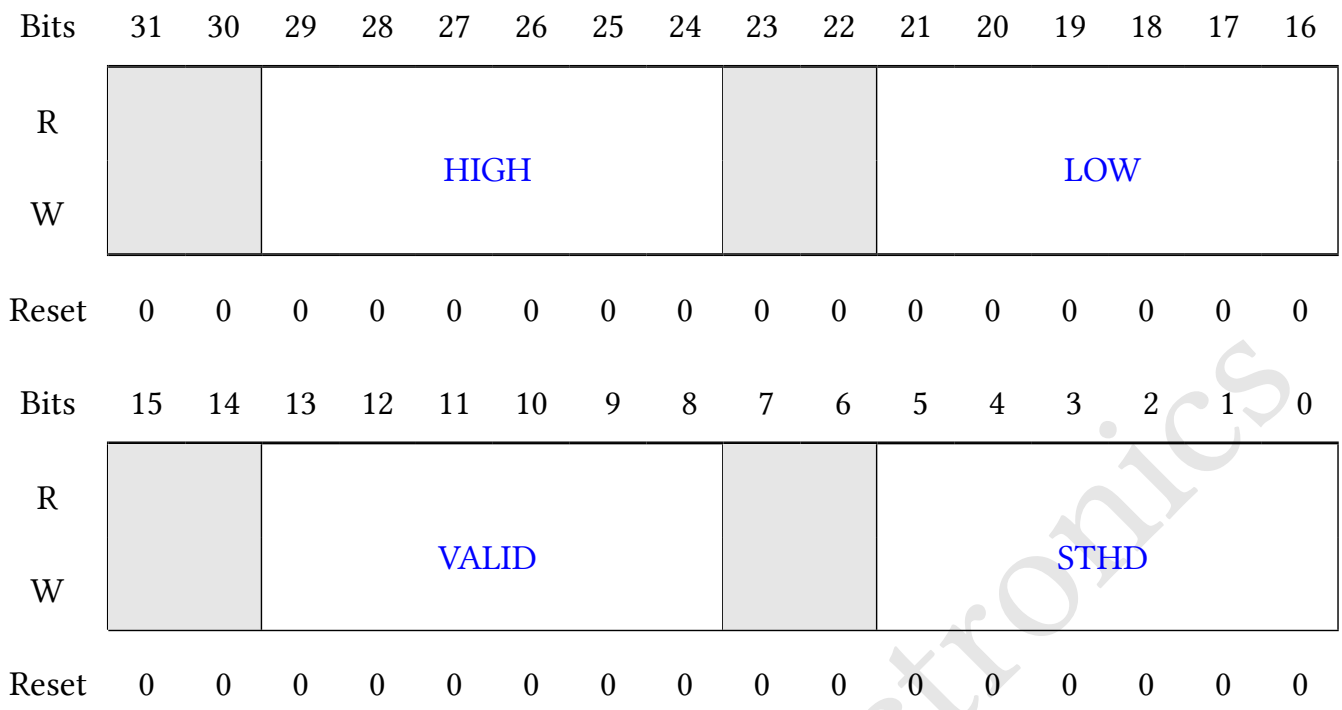


Table 24.16: I2C HSCLK Register Description

Field	Function
29 - 24 HIGH	**SCL High Period** SCL clock high period is (HIGH + 1) * divided clock period
21 - 16 LOW	**SCL Low Period** SCL clock low period is (LOW + 1) * divided clock period
13 - 8 VALID	**SDA Data Valid Period** SDA data valid period is (VALID + 1) * divided clock period
5 - 0 STHD	**SDA Data Setup/Hold Period** SDA data setup/hold period is (STHD + 1) * divided clock period

Yuntu Microelectronics

24.3.1.12 I2C SCTRL Register

Offset: 100h

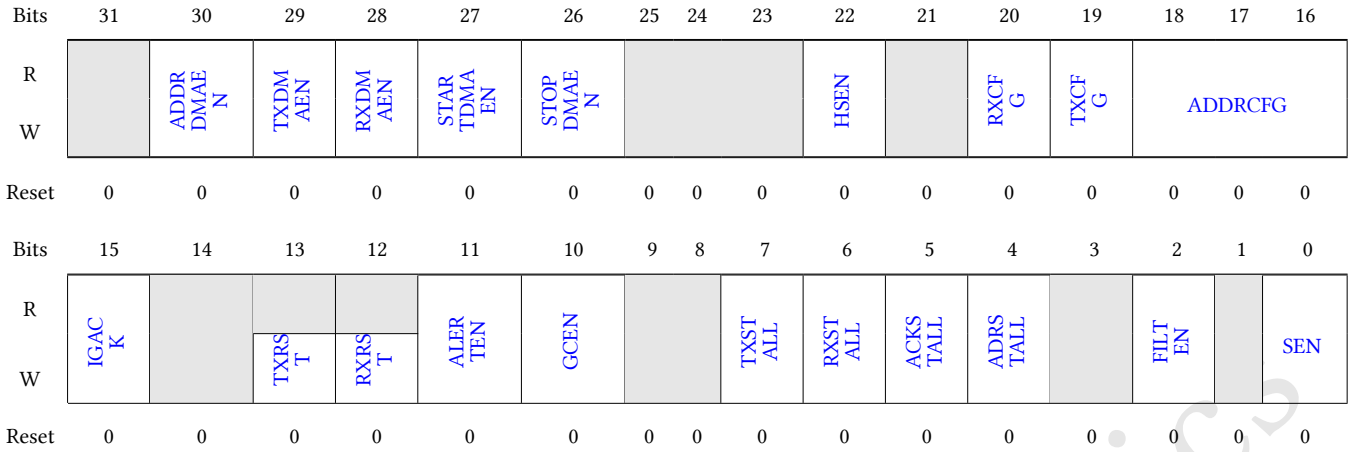


Table 24.17: I2C SCTRL Register Description

Field	Function
30 ADDRDMAEN	**Address Valid DMA Enable** Control weather DMA request generated after a valid address received 0b - DMA request disabled 1b - DMA request enabled
29 TXDMAEN	**TX Data DMA Enable** TX DATA request DMA enable when TX buff empty 0b - DMA request disabled 1b - DMA request enabled
28 RXDMAEN	**RX Data DMA Enable** RX DATA request DMA enable when RX buff full 0b - DMA request disabled 1b - DMA request enabled
27 STARTDMAEN	**Receive START DMA Enable** Control weather DMA request generated after a START condition received 0b - DMA request disabled 1b - DMA request enabled
26 STOPDMAEN	**Receive STOP DMA Enable** Control weather DMA request generated after a STOP condition received 0b - DMA request disabled 1b - DMA request enabled
22 HSEN	**High Speed Mode Enable** Setting this bit enables I2C salve high speed address match and switch enable 0b - Disable high speed mode switch 1b - Enable high speed mode switch
20	**RX Data Config**

Yuntu Microelectronics

Field	Function
RXCFG	Controls weather address include in reading RX data register 0b - Reading RX data register will only return received data and clear RX data flag 1b - Reading RX data register will only return received address when valid address mode flag set, and clear address valid flag. Reading RX data when address valid flag clear will return received data and clear RX data flag
19 TXCFG	**TX Data Config** Controls weather TX data flag asserted when I2C slave idle 0b - TX data flag only asserted when I2C active and TX buffer empty 1b - TX data flag asserted when TX buffer empty
18 - 16 ADDRCFG	**Slave Address Match Config** Control slave address match configuration 000b - Address match 0 (7-bit) 001b - Address match 0 (10-bit) 010b - Address match 0 (7-bit) or match 1(7-bit) 011b - Address match 0 (10-bit) or match 1(10-bit) 100b - Address match 0 (7-bit) or match 1(10-bit) 101b - Address match 0 (10-bit) or match 1(7-bit) 110b - Address match from match 0 (7-bit) to match 1(7-bit) 111b - Address match from match 0 (10-bit) to match 1(10-bit)
15 IGACK	**Ignore I2C ACK** Setting this bit will let I2C generate internal ACK even if NACK received. **NOTE**. This bit should be set in Ultra-Fast mode. 0b - I2C will receive ACK and NACK normally 1b - I2C will treat received NACK as ACK
13 TXRST	**TX Buffer Reset** Reset TX buffer pointer 0b - No effect 1b - Reset TX buffer pointer
12 RXRST	**RX Buffer Reset** Reset RX buffer pointer 0b - No effect 1b - Reset RX buffer pointer
11 ALERTEN	**Alert Address Detect Enable** Setting this bit will let I2C slave set SMBus alert address(0001100xb) as a match condition 0b - Alert address detect is disabled 1b - Alert address detect is enabled
10	**General Call Address Detect Enable**

Yuntu Microelectronics

Field	Function
GCEN	Setting this bit will let I2C slave set general call address(0000000xb) as a match condition 0b - General call address detect is disabled 1b - General call address detect is enabled
7 TXSTALL	**TX SCL Stall Enable** Enables I2C slave stretching SCL line to low when slave TX buffer is empty 0b - Slave stretching is disabled 1b - Slave stretching is enabled
6 RXSTALL	**RX SCL Stall Enable** Enables I2C slave stretching SCL line to low when slave RX buffer is full 0b - Slave stretching is disabled 1b - Slave stretching is enabled
5 ACKSTALL	**ACK SCL Stall Enable** Enables I2C slave stretching SCL line to low when slave waiting for software write ACK register **NOTE**. Setting ACKSTALL will set RXSTALL and ADRSTALL automatically. 0b - Slave stretching is disabled 1b - Slave stretching is enabled
4 ADRSTALL	**Address SCL Stall Enable** Enables I2C slave stretching SCL line to low when valid address received but not read 0b - Slave stretching is disabled 1b - Slave stretching is enabled
2 FILTEN	**Slave Digital Filter Enable** Slave digital filter not update when I2C slave enabled 0b - Digital filter is disabled 1b - Digital filter is enabled
0 SEN	**Slave Module Enable** Setting this bit will enable I2C slave logic 0b - Slave module is disabled 1b - Slave module is enabled

Yuntu Microelectronics

24.3.1.13 I2C SCLKCFG Register

Offset: 104h

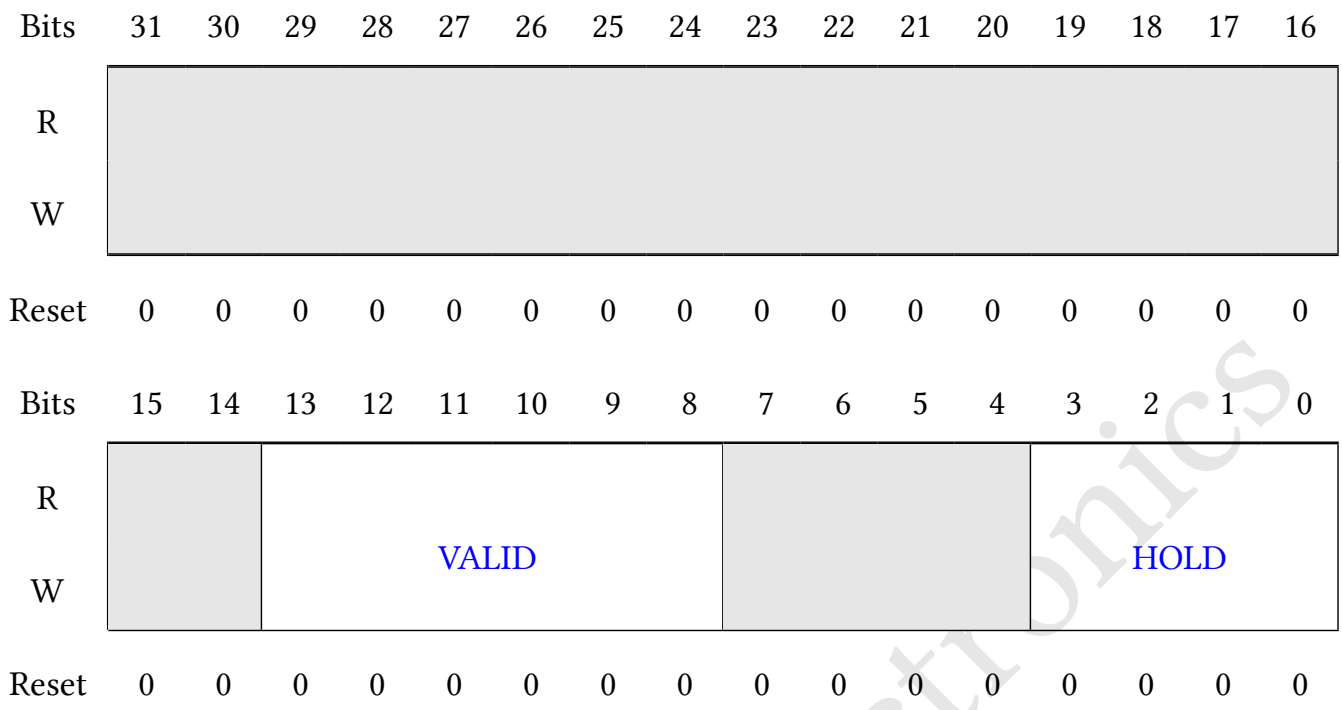


Table 24.18: I2C SCLKCFG Register Description

Field	Function
13 - 8 VALID	**Data Valid Period** SDA data delay is (FILTSCL + VALID + 3) function clock (not divided clock)
3 - 0 HOLD	**Data Hold Period** Data hold delay is (HOLD + 3) function clock (not divided clock)

Yuntu Microelectronics

24.3.1.14 I2C SFLTCFG Register

Offset: 108h

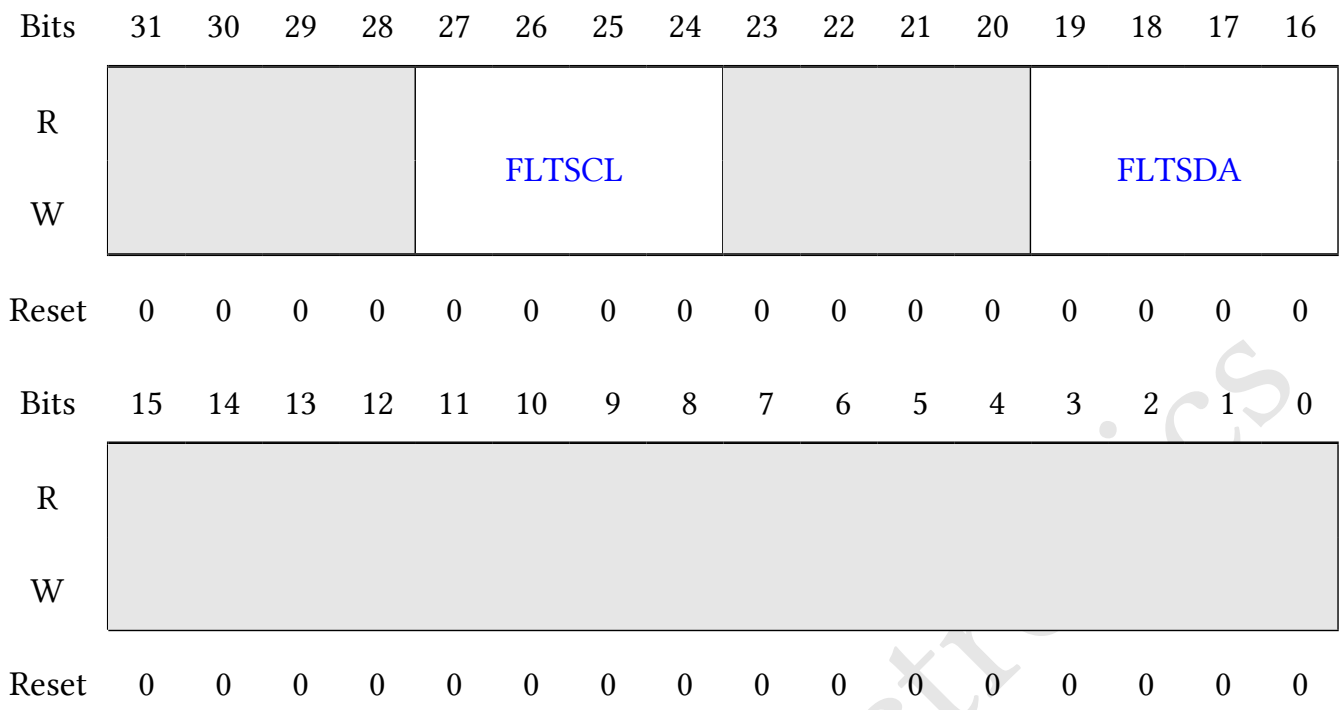


Table 24.19: I2C SFLTCFG Register Description

Field	Function
27 - 24 FLTSCL	**SCL Filter Control** Control digital filter on SCL line SCL digital filter use undivided I2C function clock Noise less or equal FLTSCL * function period will be filtered
19 - 16 FLTSDA	**SDA Filter Control** Control digital filter on SDA line SDA digital filter use undivided I2C function clock Noise less or equal FLTSDA * function period will be filtered

Yuntu Microelectronics

24.3.1.15 I2C SSTS Register

Offset: 10ch

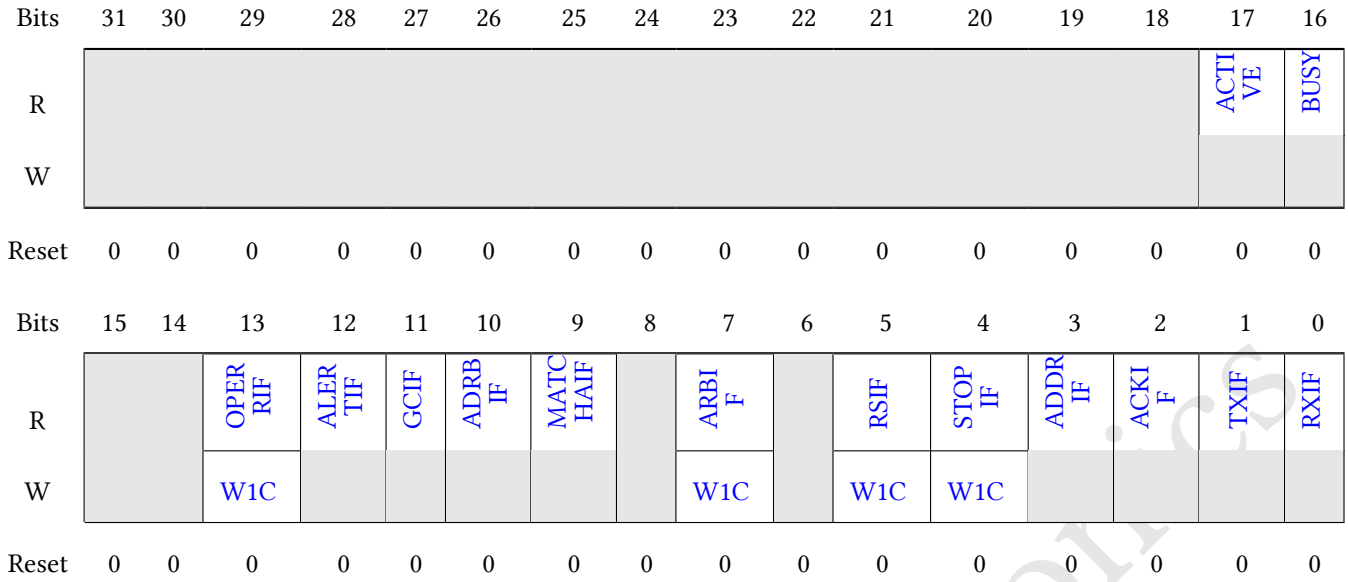


Table 24.20: I2C SSTS Register Description

Field	Function
17 ACTIVE	**I2C Slave Active** I2C slave is transmitting or receiving data 0b - I2C slave is idle 1b - I2C slave is busy
16 BUSY	**I2C Bus Busy** I2C Bus is transmitting or receiving data 0b - I2C bus is idle 1b - I2C bus is busy
13 OPERRIF	**Operation Error Interrupt Flag** I2C slave operation error, this bit is set when • Slave need to send data when TX buffer empty and TXSTALL not enabled • Slave received new data when RX buffer is full and RXSTALL not enabled • Slave received new address when address already valid and ADDRSTALL not enabled
12 ALERTIF	**Alert Address Received Interrupt Flag** 0b - Alert address not received 1b - Alert address received
11 GCIF	**General Call Address Received Interrupt Flag** 0b - General call address not received 1b - General call address received
10 ADRBIF	**Address B Received Interrupt Flag** Slave received address match address B flag 0b - Address B not received 1b - Address B received

Yuntu Microelectronics

Field	Function
9 MATCHAIF	**Address A Received Interrupt Flag** Slave received address match address A flag 0b - Address A not received 1b - Address A received
7 ARBIF	**Arbitration Lost Interrupt Flag** This bit is set when Slave send out logic 1 but detect logic 0 on SCL or SDA line Slave will ignore following transfer until new (repeat) START received 0b - I2C slave working normal 1b - I2C slave lost arbitration
5 RSIF	**Repeat Start Interrupt Flag** Repeat start set when I2C received a repeat START and previous address matched 0b - I2C slave not received repeat start 1b - I2C slave received repeat start
4 STOPIF	**Stop Interrupt Flag** Stop interrupt set when I2C already addressed and received a stop condition 0b - I2C slave not received STOP 1b - I2C slave received STOP
3 ADDRIF	**Address Valid Interrupt Flag** Address valid set when address match received and cleared after read address register or read DATA register when RXCFG bit set 0b - I2C slave not received any valid address 1b - I2C slave received a valid address
2 ACKIF	**Ack Request Interrupt Flag** Ack request set when slave need to send ACK or NACK to master, this bit cleared by write ACK register 0b - I2C slave don't need to send ACK or NACK to master 1b - I2C slave need to send ACK or NACK to master
1 TXIF	**TX Request Flag** Indicates words in TX FIFO is equal or less than TX FIFO watermark setting 0b - Word in TX FIFO is greater than watermark 1b - Word in TX FIFO is equal or less than watermark, request to write new data
0 RXIF	**RX Request Flag** Indicates words in RX FIFO is greater than RX FIFO watermark setting 0b - Word in RX FIFO is equal or less than watermark 1b - Word in RX FIFO is greater than watermark, should read data register

Yuntu Microelectronics

24.3.1.16 I2C SIE Register

Offset: 110h

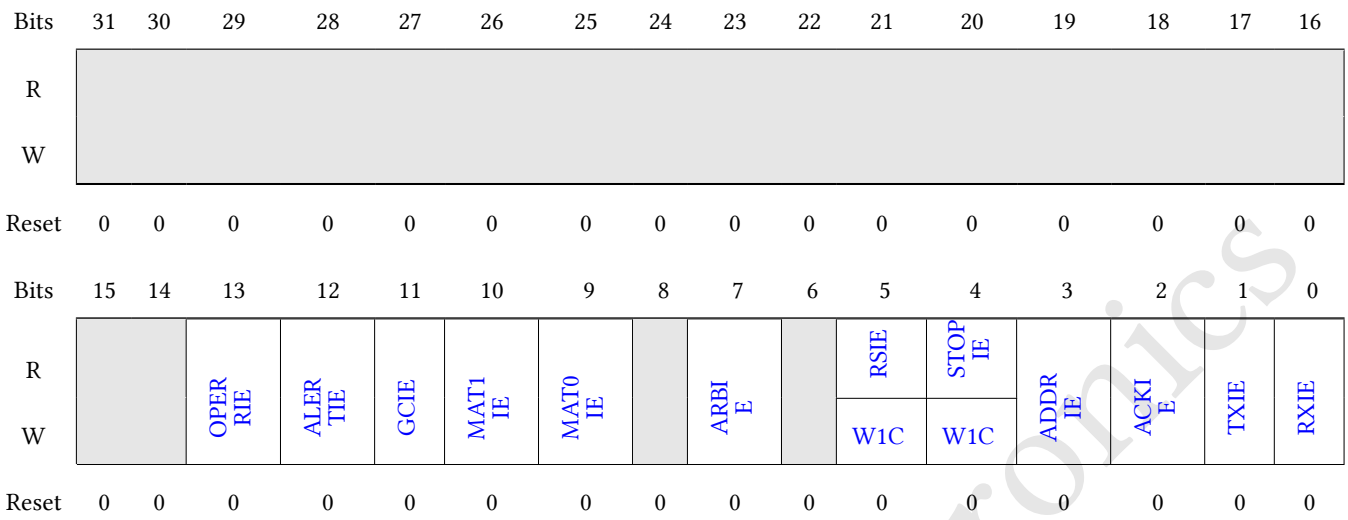


Table 24.21: I2C SIE Register Description

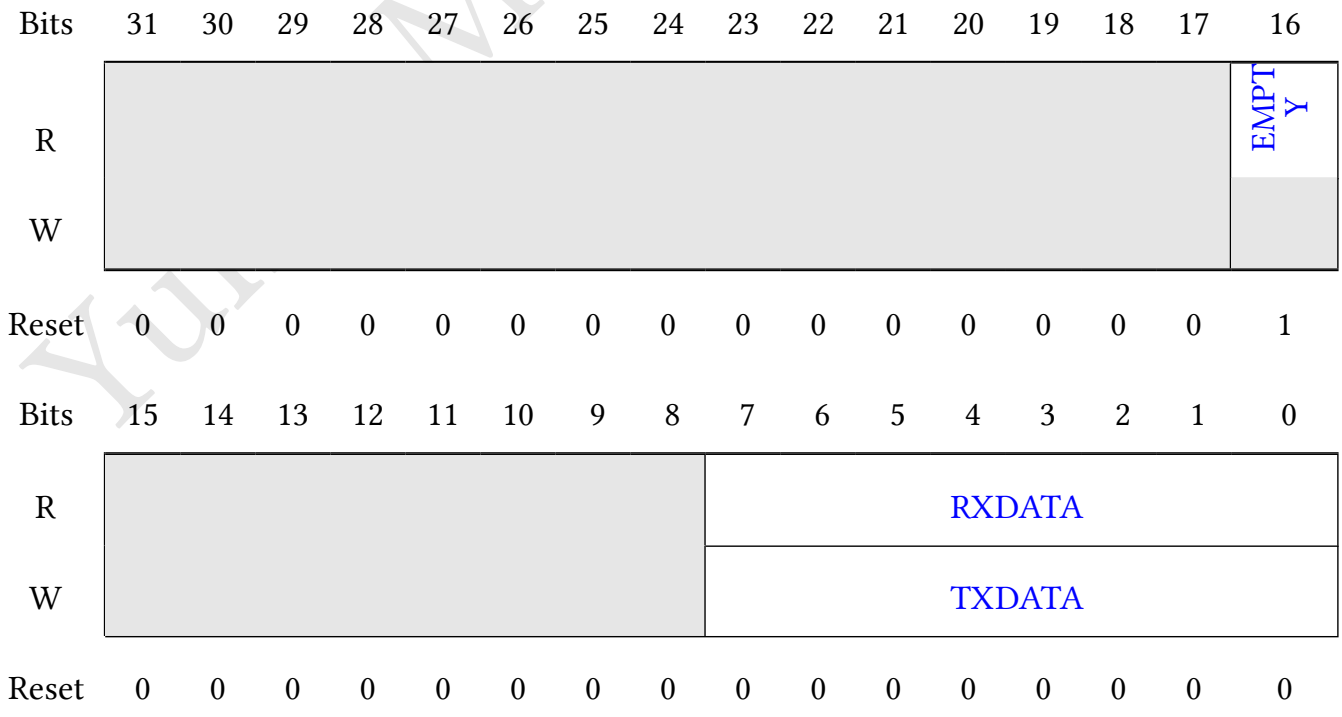
Field	Function
13 OPERRIE	**Slave Operation Error Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
12 ALERTIE	**Alert Address Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
11 GCIE	**General Call Address Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
10 MAT1IE	**Match Address 1 Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
9 MAT0IE	**Match Address 0 Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
7 ARBIE	**Arbitration Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
5	**Repeat Start Interrupt Enable**

Yuntu Microelectronics

Field	Function
RSIE	0b - Interrupt disabled 1b - Interrupt enabled
4 STOPIE	**Stop Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
3 ADDRIE	**Address Valid Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
2 ACKIE	**ACK Request Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
1 TXIE	**TX DATA Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled
0 RXIE	**RX DATA Interrupt Enable** 0b - Interrupt disabled 1b - Interrupt enabled

24.3.1.17 I2C SDATA Register

Offset: 114h



Yuntu Microelectronics

Table 24.22: I2C SDATA Register Description

Field	Function
16 EMPTY	**RX Buffer Empty** 0b - RX buffer is not empty 1b - RX buffer is empty
7 - 0 DATA	**Slave TX RX Data** Write will push data to TX buffer and read will return received data or address

24.3.1.18 I2C SACK Register

Offset: 118h

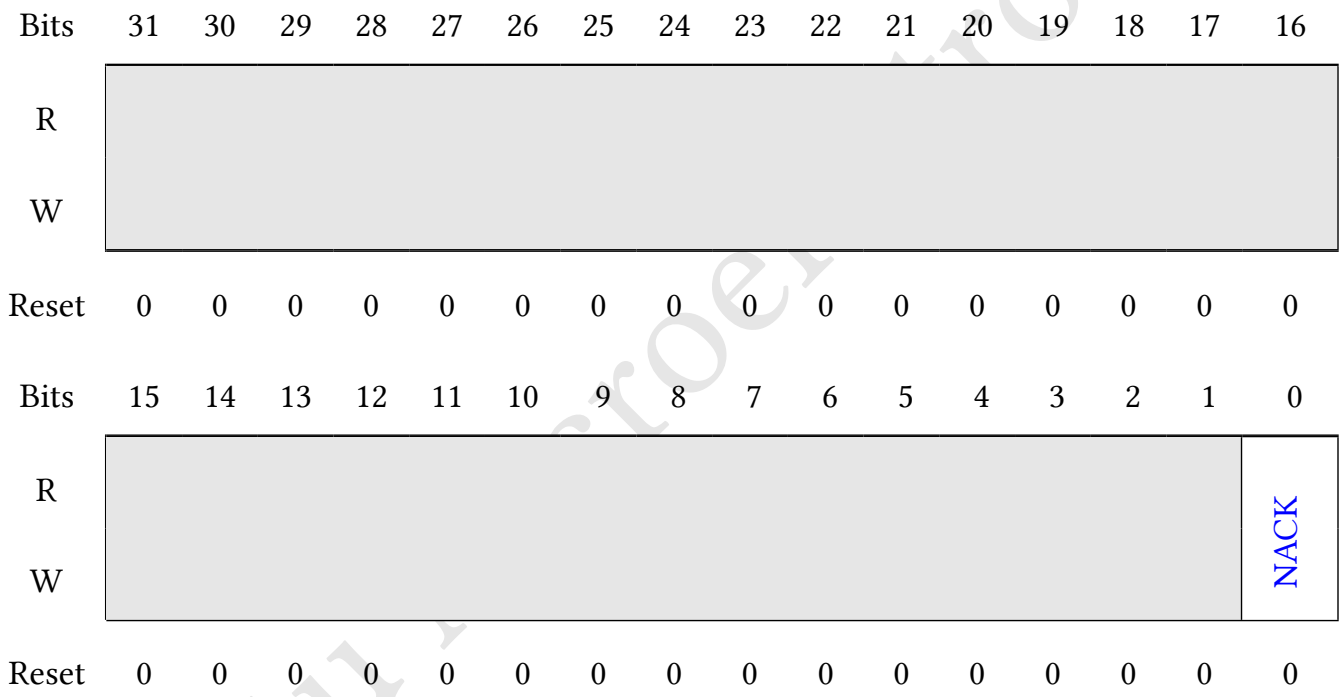


Table 24.23: I2C SACK Register Description

Field	Function
0 NACK	**Sent NACK** After slave received data, software can write NACK with 1 to send NACK or 0 to send ACK 0b - Send ACK 1b - Send NACK

24.3.1.19 I2C SADDR Register

Offset: 11ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W						ADDRB										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W						ADDRA										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 24.24: I2C SADDR Register Description

Field	Function
26 - 16 ADDRB	**Slave ADDR B**
10 - 0 ADDRA	**Slave ADDR A**

24.3.1.20 I2C SRADDR Register

Offset: 120h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VALID					ADDR										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 24.25: I2C SRADDR Register Description

Field	Function
15 VALID	**Address Valid** 0b - Slave address is not valid 1b - Slave address is valid
10 - 0 ADDR	**Slave Received Address**

24.4 Functional Description

The I2C module uses two physical lines for communication:

- Serial Data Line (SDA) for data transfer
- Serial Clock Line (SCL) for the bus clock

A transaction starts with the I2C master sending the start condition, followed by address and a direction bit (read or write to/from the slave).

The addressed I2C slave will then acknowledge (ACK) the address, and data packet transactions can begin. Every 9-bit data packet consists of 8 data bits followed by a one-bit reply indicating whether the data was acknowledged or not.

If a data packet is not acknowledged (NACK), whether by the I2C slave or master, the I2C master takes action by either terminating the transaction by sending the stop condition, or by sending a repeated start to transfer more data.

24.4.1 I2C Functional Clock

There are a few clock sources for the I2C functional clock, and it can be configured by IPC. It is asynchronous to the bus clock. Whether I2C is able to work in the low power modes depends on whether the I2C functional clock remains enabled. The clock used for timing control in the register MFLTCFG, SCLKCFG and SFLTCFG is the I2C functional clock, while the clock used for timing control in the register MCLKCFG and HSCLK is the divided I2C functional clock.

24.4.2 Master Mode

24.4.2.1 Tx and CMD FIFO

The Tx and CMD FIFO stores the command and the data together in one word. There are various command and data combination corresponding to different I2C operations.

- START + DATA: Generate START or repeated START and transmit the address in DATA.
- STOP + START + DATA: Generate START or repeated START and transmit the address in DATA. Expect NACK from slave.
- No command + DATA: Transmit the data in DATA.
- REC + DATA: Receive (DATA+1) bytes data.
- REC + STOP + DATA: Receive (DATA+1) bytes data and discard them.
- STOP: Generate STOP.

Yuntu Microelectronics

- HS + START + DATA: Generate START or repeated START and transmit the address in DATA with high speed mode.
- HS + STOP + START + DATA: Generate START or repeated START and transmit the address in DATA with high speed mode. Expect NACK from slave.
- HS + DATA: Transmit the data in DATA with high speed mode.

The transmitted or received commands can be inserted between START and STOP. But it needs repeated START before changing the transmission direction. The I2C master will automatically transmit a NACK on the last byte of a received data command unless the next command in the FIFO is also a received data command. The I2C master supports 10-bit addressing through a (repeated) START condition, followed by a transmitted data byte containing the second address byte, followed by any number of data bytes with the transmitted or received data.

24.4.2.2 Master Operations

If I2C is enabled, the I2C starts transmission when it is idle by checking the MSTS[BUSY] bit. When either SCL or SDA are low, the I2C bus is busy. The I2C bus becomes idle if a STOP condition or a bus idle timeout is detected. After the I2C bus is idle, the I2C will check Tx FIFO. If there is any data in Tx FIFO and the first word is read out, and the legal command will be executed. Below are the steps.

- Detect a STOP condition on the I2C bus or Wait the bus idle time equal to (MCLKCFG[LOW]+1) cycles divided clocks (MFLTCFG[DIV]).
- Transmit a START condition and address byte.
- Perform master-transmit or master-receive transfers, as configured by the Tx FIFO.
- Transmit NACK on the last byte of a master-receive transfer, unless the next command in the Tx FIFO is also a received data command.
- Transmit a Repeated START or STOP condition as configured by the Tx FIFO.

If the I2C master is disabled, the I2C will continue to empty the Tx FIFO or a STOP condition is transmitted.

The I2C master can stall the I2C bus under certain conditions; this will result in SCL pulled low continuously:

- I2C master is enabled and busy, the Tx FIFO is empty, and MCTRL[ASTOP] is clear.
- I2C master is enabled and receiving data, received data is not being discarded, and the Rx FIFO is full.

24.4.2.3 Rx FIFO and Data Matching

In Master mode, the Rx FIFO stores received data during master-receiver transfers. Receive data will be discarded instead of storing in the Rx FIFO when discarded command is configured.

The I2C supports a received data match function that can match received data against one of two bytes or against a masked data byte. The data match function can also be configured to compare only the first one or two received data words since the last (repeated) START condition. The receiver match function can also be configured to discard all receive data until a data match is detected, using the MCTRL[RDMO] control bit.

24.4.3 Slave Mode

24.4.3.1 Address Matching

The I2C slave can be configured with the address match conditions.

Yuntu Microelectronics

- Match one of the two addresses, both 7-bit and 10-bit addressing modes are supported.
- Match a range of addresses, both 7-bit and 10-bit addressing modes are supported.
- Match the General Call Address.
- Match the SMBus Alert Address.
- Detect the high speed mode master code, and then disable the digital filters use the valid delay time until the next STOP condition is detected.

After a valid address is matched, the I2C slave will automatically perform slave-transmit or slave-receive transfers until:

- A NACK is detected (unless SCTRL[IGACK] is set).
- A bit error is detected (the I2C slave is driving SDA, but a different value is sampled)
- A (repeated) START or STOP condition is detected.

24.4.3.2 Data Processing

- The Tx and Rx Data registers are double-buffered
- The slave address that was received can be read from either the Slave Data register or the Slave Receive Address register.
- The Slave Data register can be configured to only request data whenever the Tx buffer is empty or a slave-transmit transfer is detected when the Tx buffer is empty.
- The Slave Data register should only be written when the SSTS[TXIF] is set.
- The Slave Receive Address register should only be read the received address when the Address Valid flag is set.

24.4.3.3 Clock Stretching

The I2C slave supports a mechanism for clock stretching. The following conditions can be configured to perform clock stretching.

- During the 9th clock pulse of the address byte and the Address Valid flag is set which enabled by SCTRL[ADRSTALL].
- During the 9th clock pulse of a slave-transmit transfer and the Transmit Data flag is set which enabled by SCTRL[TXSTALL].
- During the 9th clock pulse of a slave-receive transfer and the RX Request flag is set which enabled by SCTRL[RXSTALL].
- During the 8th clock pulse of an address byte or a slave-receive transfer and the ACK Request flag is set which enabled by SCTRL[ACKSTALL]. In high speed mode, this is disabled.
- Clock stretching can also be extended for CLKHOLD cycles, to allow additional setup time to sample the SDA pin externally. In high speed mode, this is disabled.

24.5 Glossary

SCL	Serial clock line
SDA	Serial data line

Yuntu Microelectronics

Chapter 25

Single Edge Nibble Transmission (SENT)

25.1 Chip-specific SENT Information

25.1.1 Instantiation Information

YTM32B1HA0x contains 2 SENT modules with 4 channels that can be used to receive serial data frames from sensors.

25.1.2 Signal Description

Table 25.1: SENT Signal Table

Signal Name	Description	Direction
SENT_CH0	Channel 0 inputs from pad.	I
SENT_CH1	Channel 1 inputs from pad.	I
SENT_CH2	Channel 2 inputs from pad.	I
SENT_CH3	Channel 3 inputs from pad.	I

25.1.3 Clock Description

Table 25.2: SENT Clock Table

Clock Name	Description
Bus Clock	Bus Clock used for register access
Functional Clock	Functional Clock used for receiving serial data

25.1.4 DMA Description

Table 25.3: SENT DMA Table

No.	Request Source	Description
102	SENT0 Fast Message receive done or FIFO data reach the watermark	SENT0 Fast Message data is ready for read will generate request

Yuntu Microelectronics

Table 25.3 continued from previous page

No.	Request Source	Description
103	SENT0 Slow Serial Message receive done	SENT0 Short Serial Message or Enhanced Serial Message is ready for read will generate request
104	SENT1 Fast Message receive done or FIFO data reach the watermark	SENT1 Fast Message data is ready for read will generate request
105	SENT1 Slow Serial Message receive done	SENT1 Short Serial Message or Enhanced Serial Message is ready for read will generate request

25.1.5 Interrupt Description

Table 25.4: SENT Interrupt Table

Model	Vector	IRQn	Request Source	Description
SENT0	173	157	Message received or error detected	Fast Message received, Slow Serial Message received, error detected will generate interrupt
SENT1	174	158	Message received or error detected	Fast Message received, Slow Serial Message received, error detected will generate interrupt

25.2 Introduction

25.2.1 Overview

The Single Edge Nibble Transmission (SENT) module on this chip acts as a multichannel receiver for receiving serial data frames from sensors, these data frames should be complied with the SENT communication protocol. This module is based on the SAE J2716 information report titled "SENT - Single Edge Nibble Transmission for Automotive Applications" that released on January 27, 2010.

25.2.2 Features

This chip contains two SENTs (SENT0 and SENT1), and they have the same features:

- Always acts as a receiver
- Support selectable functional clock for message receiving
- Support four channels for different devices
- Support SENT protocol specification J2716 JAN2010
- Support programmable input filter for each channel
- Support configurable receive tick times from 3μs to 90μs for each channel
- Support auto compensation for variation in SENT transmit clock up to $\pm$ 25%
- Support configurable number of data nibbles for each channel
- Support status nibble optionally included in the checksum
- Support pause pulse for each channel
- Support separate channel buffer for storing Fast and Slow Serial Message
- Support FIFO mode to store Fast Message from all channels (Depth 16)
- Support DMA to access Fast and Slow Serial Message

Yuntu Microelectronics

- Support time stamp for all receive message
- Support varied interrupts
  - Fast Message receive interrupt
  - Slow Serial Message receive interrupt
  - Channel receive error interrupt
  - FIFO/buffer overflow and underflow interrupt
  - Wake up interrupt (any channel busy to trigger)

25.2.3 Block diagram

The following figure shows the SENT block diagram.

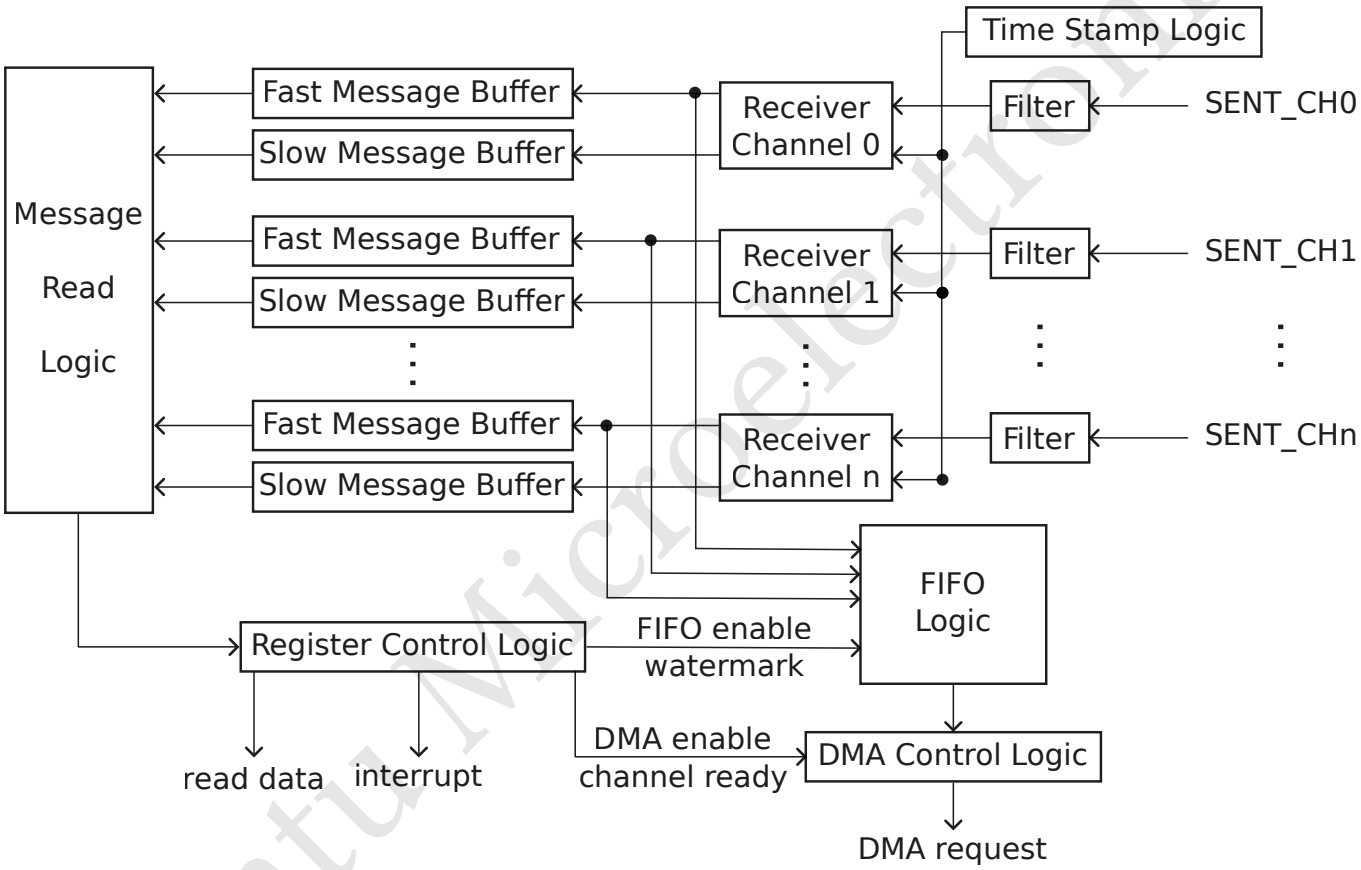


Figure 25.1: SENT Block Diagram

25.3 Register definition

Registers of this module allow 32/16/8-bit access.

The values of reserved register bits are 0 when reading, write will have no influence.

Accessing a reserved address will result in a bus error.

Register_address = Base_address + Address_offset
For example: as CHn_CLK
Address_offset = 0x50 + n * 0x30 (n=0/1/2/3)

Yuntu Microelectronics

25.3.1 SENT Memory Map

SENT0 base address 40160000h
SENT1 base address 40164000h

Table 25.5: SENT Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Channel Control Register (CHCTRL)	32	RW	00000000h
8h	Global Status Register (GSTS)	32	RW	00000000h
ch	Global Interrupt Register (GINTE)	32	RW	00000000h
10h	Fast Message Ready Register (FRDY)	32	RW	00000000h
14h	Slow Serial Message Ready Register (SRDY)	32	RW	00000000h
18h	Fast Message Ready Interrupt Register (FINTE)	32	RW	00000000h
1ch	Slow Serial Message Ready Interrupt Register (SINTE)	32	RW	00000000h
20h	DMA Control Register (DMACTRL)	32	RW	00000000h
24h	Configuration Register (CFG)	32	RW	00000016h
28h	Nibble Configuration Register (NIBCFG)	32	RW	00000011h
30h	Fast Message Data Register (FMDATA)	32	RO	00000000h
34h	Fast Message Channel Number Register (FMCHNUM)	32	RO	00000000h
38h	Fast Message Time Stamp Register (FMTS)	32	RO	00000000h
3ch	Slow Serial Message Bit 3 Register (SMBIT3)	32	RO	00000000h
40h	Slow Serial Message Bit 2 Register (SMBIT2)	32	RO	00000000h
44h	Slow Serial Message Time Stamp Register (SMTS)	32	RO	00000000h
50h	Channel n Clock Register (CH0_CLK)	32	RW	00008000h
54h	Channel n Status Register (CH0_STS)	32	RW	00000000h
58h	Channel n Interrupt Enable Register (CH0_INTE)	32	RW	00000000h
5ch	Channel n Configuration Register (CH0_CFG)	32	RW	00800400h
60h	Channel n Fast Message Data Register (CH0_FMDATA)	32	RO	00000000h
64h	Channel n Fast Message Channel Number Register (CH0_FMCHNUM)	32	RO	00000000h
68h	Channel n Fast Message Time Stamp Register (CH0_FMTS)	32	RO	00000000h
6ch	Channel n Slow Serial Message Bit 3 Register (CH0_SMBIT3)	32	RO	00000000h
70h	Channel n Slow Serial Message Bit 2 Register (CH0_SMBIT2)	32	RO	00000000h
74h	Channel n Slow Serial Message Time Stamp Register (CH0_SMTS)	32	RO	00000000h
80h	Channel n Clock Register (CH1_CLK)	32	RW	00008000h
84h	Channel n Status Register (CH1_STS)	32	RW	00000000h
88h	Channel n Interrupt Enable Register (CH1_INTE)	32	RW	00000000h
8ch	Channel n Configuration Register (CH1_CFG)	32	RW	00800400h
90h	Channel n Fast Message Data Register (CH1_FMDATA)	32	RO	00000000h
94h	Channel n Fast Message Channel Number Register (CH1_FMCHNUM)	32	RO	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
98h	Channel n Fast Message Time Stamp Register (CH1_FMTS)	32	RO	00000000h
9ch	Channel n Slow Serial Message Bit 3 Register (CH1_SMBIT3)	32	RO	00000000h
a0h	Channel n Slow Serial Message Bit 2 Register (CH1_SMBIT2)	32	RO	00000000h
a4h	Channel n Slow Serial Message Time Stamp Register (CH1_SMTS)	32	RO	00000000h
b0h	Channel n Clock Register (CH2_CLK)	32	RW	00008000h
b4h	Channel n Status Register (CH2_STS)	32	RW	00000000h
b8h	Channel n Interrupt Enable Register (CH2_INTE)	32	RW	00000000h
bch	Channel n Configuration Register (CH2_CFG)	32	RW	00800400h
c0h	Channel n Fast Message Data Register (CH2_FMDATA)	32	RO	00000000h
c4h	Channel n Fast Message Channel Number Register (CH2_FMCHNUM)	32	RO	00000000h
c8h	Channel n Fast Message Time Stamp Register (CH2_FMTS)	32	RO	00000000h
cch	Channel n Slow Serial Message Bit 3 Register (CH2_SMBIT3)	32	RO	00000000h
d0h	Channel n Slow Serial Message Bit 2 Register (CH2_SMBIT2)	32	RO	00000000h
d4h	Channel n Slow Serial Message Time Stamp Register (CH2_SMTS)	32	RO	00000000h
e0h	Channel n Clock Register (CH3_CLK)	32	RW	00008000h
e4h	Channel n Status Register (CH3_STS)	32	RW	00000000h
e8h	Channel n Interrupt Enable Register (CH3_INTE)	32	RW	00000000h
ech	Channel n Configuration Register (CH3_CFG)	32	RW	00800400h
f0h	Channel n Fast Message Data Register (CH3_FMDATA)	32	RO	00000000h
f4h	Channel n Fast Message Channel Number Register (CH3_FMCHNUM)	32	RO	00000000h
f8h	Channel n Fast Message Time Stamp Register (CH3_FMTS)	32	RO	00000000h
fch	Channel n Slow Serial Message Bit 3 Register (CH3_SMBIT3)	32	RO	00000000h
100h	Channel n Slow Serial Message Bit 2 Register (CH3_SMBIT2)	32	RO	00000000h
104h	Channel n Slow Serial Message Time Stamp Register (CH3_SMTS)	32	RO	00000000h

Yuntu Microelectronics

25.3.1.1 SENT CTRL Register

Offset: 0h

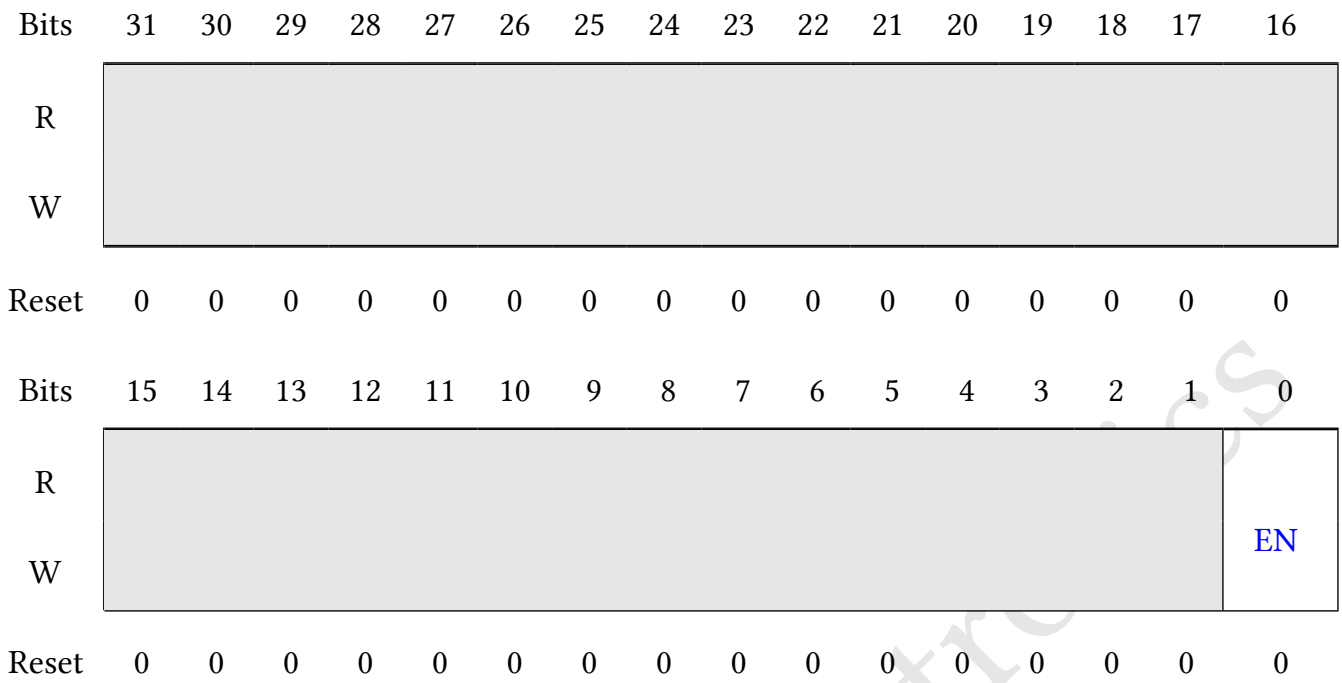
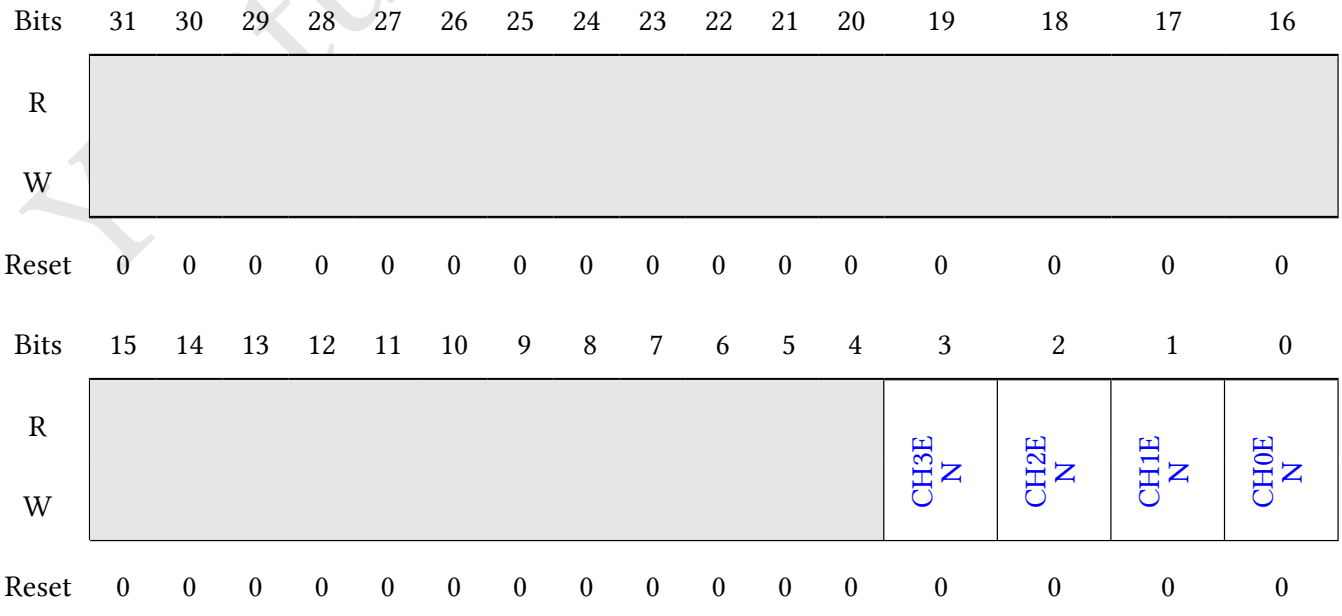


Table 25.6: SENT CTRL Register Description

Field	Function
0 EN	**Enable** Sent module enable control. 0b - Disabled. 1b - Enabled.

25.3.1.2 SENT CHCTRL Register

Offset: 4h



Yuntu Microelectronics

Table 25.7: SENT CHCTRL Register Description

Field	Function
3 CH3EN	**Channel 3 Enable** Sent channel 3 enable control. 0b - Disabled. 1b - Enabled.
2 CH2EN	**Channel 2 Enable** Sent channel 2 enable control. 0b - Disabled. 1b - Enabled.
1 CH1EN	**Channel 1 Enable** Sent channel 1 enable control. 0b - Disabled. 1b - Enabled.
0 CH0EN	**Channel 0 Enable** Sent channel 0 enable control. 0b - Disabled. 1b - Enabled.

25.3.1.3 SENT GSTS Register

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R													CH3ERR	CH2ERR	CH1ERR	CH0ERR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													WUF	FMFO	FMDU	SMDU
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 25.8: SENT GSTS Register Description

Field	Function
19 CH3ERR	**Channel 3 Error** Indicate channel 3 receive error status. 0b - No error detected. 1b - Any error detected on channel 3.
18 CH2ERR	**Channel 2 Error** Indicate channel 2 receive error status. 0b - No error detected. 1b - Any error detected on channel 2.
17 CH1ERR	**Channel 1 Error** Indicate channel 1 receive error status. 0b - No error detected. 1b - Any error detected on channel 1.
16 CH0ERR	**Channel 0 Error** Indicate channel 0 receive error status. 0b - No error detected. 1b - Any error detected on channel 0.
3 WUF	**Wake-Up Flag** Indicate received valid data from PAD. This flag can be used to wake up chip from low power mode. Write 1 to clear this bit. 0b - No valid input was detected. 1b - Valid input was detected.
2 FMFO	**Fast Message FIFO Overflow** Indicate fast message is received on any channel when FIFO is full. This bit is valid only when FIFO is enabled. Write 1 to clear this bit. 0b - No overflow. 1b - Overflow occurred.
1 FMDU	**Fast Message DMA Underflow** Indicate read when Fast Message DMA buffer or FIFO is empty. Write 1 to clear this bit. 0b - No underflow. 1b - Underflow occurred.
0 SMDU	**Slow Serial Message DMA Underflow** Indicate read when Slow Serial Message DMA buffer is empty. Write 1 to clear this bit. 0b - No underflow. 1b - Underflow occurred.

Yuntu Microelectronics

25.3.1.4 SENT GINTE Register

Offset: ch

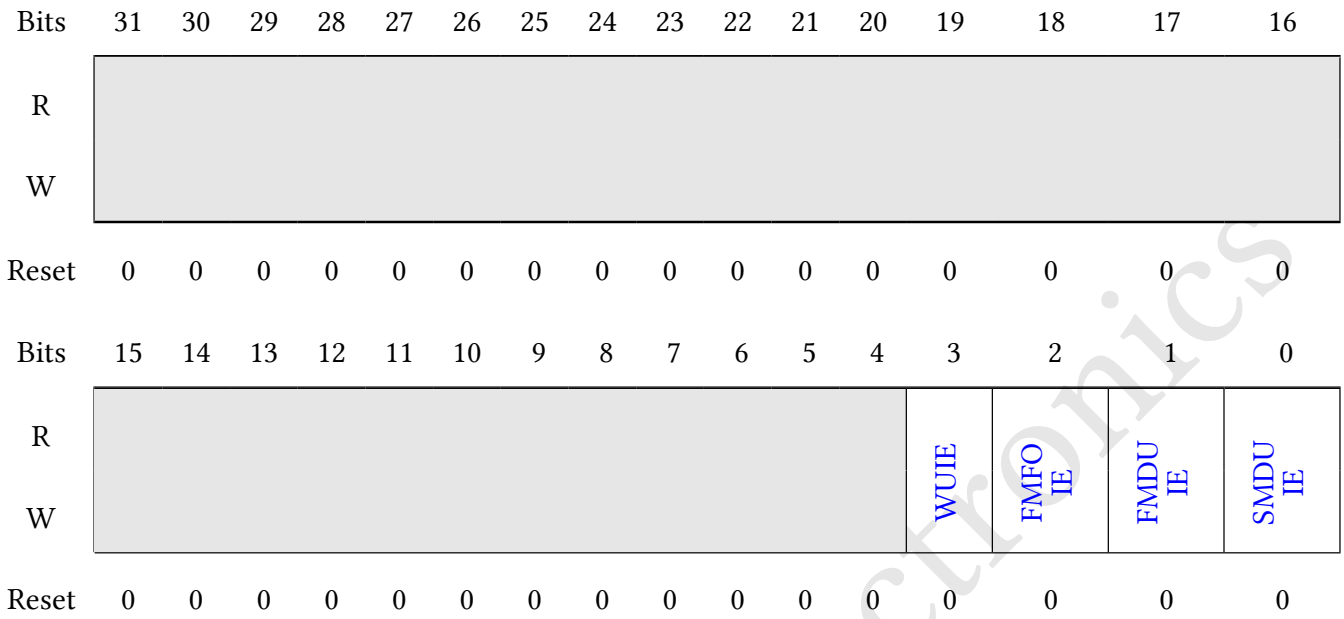


Table 25.9: SENT GINTE Register Description

Field	Function
3 WUIE	**Wake-up Interrupt Enable** Wake-up interrupt enable control. 0b - Disabled. 1b - Enabled.
2 FMFOIE	**Fast Message FIFO Overflow Interrupt Enable** Fast Message FIFO overflow interrupt enable control. 0b - Disabled. 1b - Enabled.
1 FMDUIE	**Fast Message DMA Underflow Interrupt Enable** Fast Message DMA buffer or FIFO underflow interrupt enable control. 0b - Disabled. 1b - Enabled.
0 SMDUIE	**Slow Serial Message DMA Underflow Interrupt Enable** Slow Serial Message DMA buffer underflow interrupt enable control. 0b - Disabled. 1b - Enabled.

25.3.1.5 SENT FRDY Register

Offset: 10h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													CH3FRDY	CH2FRDY	CH1FRDY	CH0FRDY
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 25.10: SENT FRDY Register Description

Field	Function
3 CH3FRDY	**Channel 3 Fast Message Ready** Indicate channel 3 Fast Message receive ready. 0b - No Fast Message is available to be read out. 1b - Fast Message is available to be read out.
2 CH2FRDY	**Channel 2 Fast Message Ready** Indicate channel 2 Fast Message receive ready. 0b - No Fast Message is available to be read out. 1b - Fast Message is available to be read out.
1 CH1FRDY	**Channel 1 Fast Message Ready** Indicate channel 1 Fast Message receive ready. 0b - No Fast Message is available to be read out. 1b - Fast Message is available to be read out.
0 CH0FRDY	**Channel 0 Fast Message Ready** Indicate channel 0 Fast Message receive ready. 0b - No Fast Message is available to be read out. 1b - Fast Message is available to be read out.

25.3.1.6 SENT SRDY Register

Offset: 14h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R													CH3SRDY	CH2SRDY	CH1SRDY	CH0SRDY
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 25.11: SENT SRDY Register Description

Field	Function
3 CH3SRDY	**Channel 3 Slow Serial Message Ready** Indicate channel 3 Slow Serial Message receive ready. 0b - No Slow Serial Message is available to be read out. 1b - Slow Serial Message is available to be read out.
2 CH2SRDY	**Channel 2 Slow Serial Message Ready** Indicate channel 2 Slow Serial Message receive ready. 0b - No Slow Serial Message is available to be read out. 1b - Slow Serial Message is available to be read out.
1 CH1SRDY	**Channel 1 Slow Serial Message Ready** Indicate channel 1 Slow Serial Message receive ready. 0b - No Slow Serial Message is available to be read out. 1b - Slow Serial Message is available to be read out.
0 CH0SRDY	**Channel 0 Slow Serial Message Ready** Indicate channel 0 Slow Serial Message receive ready. 0b - No Slow Serial Message is available to be read out. 1b - Slow Serial Message is available to be read out.

25.3.1.7 SENT FINTE Register

Offset: 18h

Yuntu Microelectronics

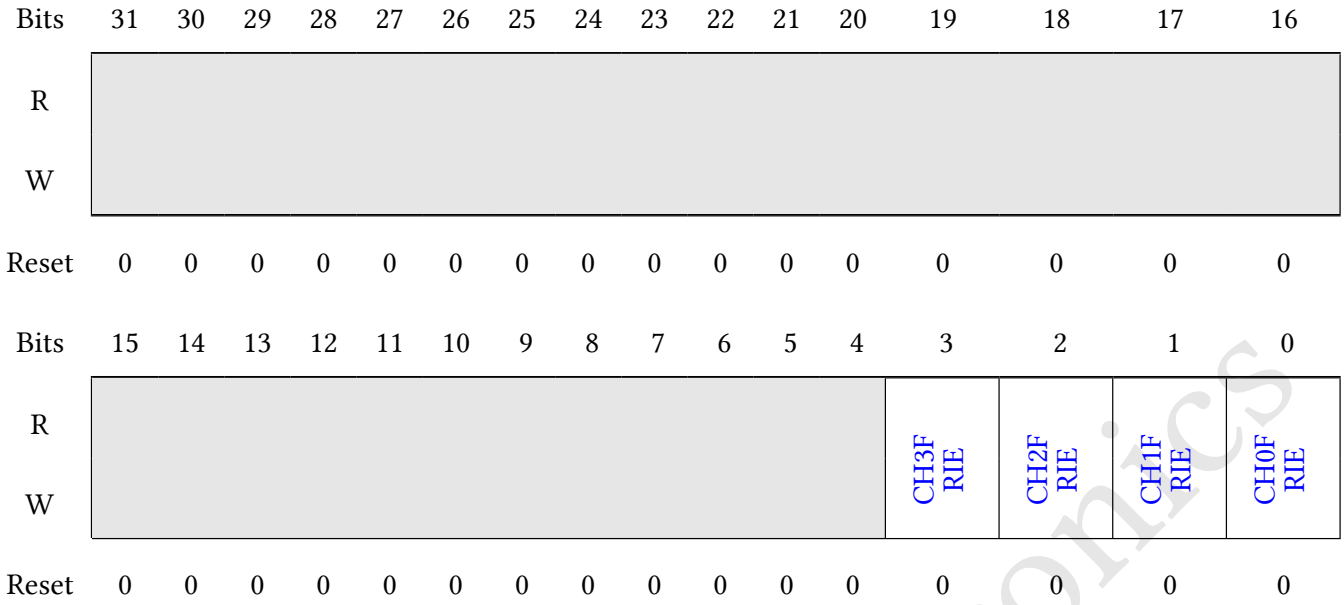


Table 25.12: SENT FINTE Register Description

Field	Function
3 CH3FRIE	**Channel 3 Fast Message Ready Interrupt Enable** Channel 3 Fast Message ready interrupt enable control. 0b - Disabled. 1b - Enable.
2 CH2FRIE	**Channel 2 Fast Message Ready Interrupt Enable** Channel 2 Fast Message ready interrupt enable control. 0b - Disabled. 1b - Enable.
1 CH1FRIE	**Channel 1 Fast Message Ready Interrupt Enable** Channel 1 Fast Message ready interrupt enable control. 0b - Disabled. 1b - Enable.
0 CH0FRIE	**Channel 0 Fast Message Ready Interrupt Enable** Channel 0 Fast Message ready interrupt enable control. 0b - Disabled. 1b - Enable.

Yuntu Microelectronics

25.3.1.8 SENT SINTE Register

Offset: 1ch

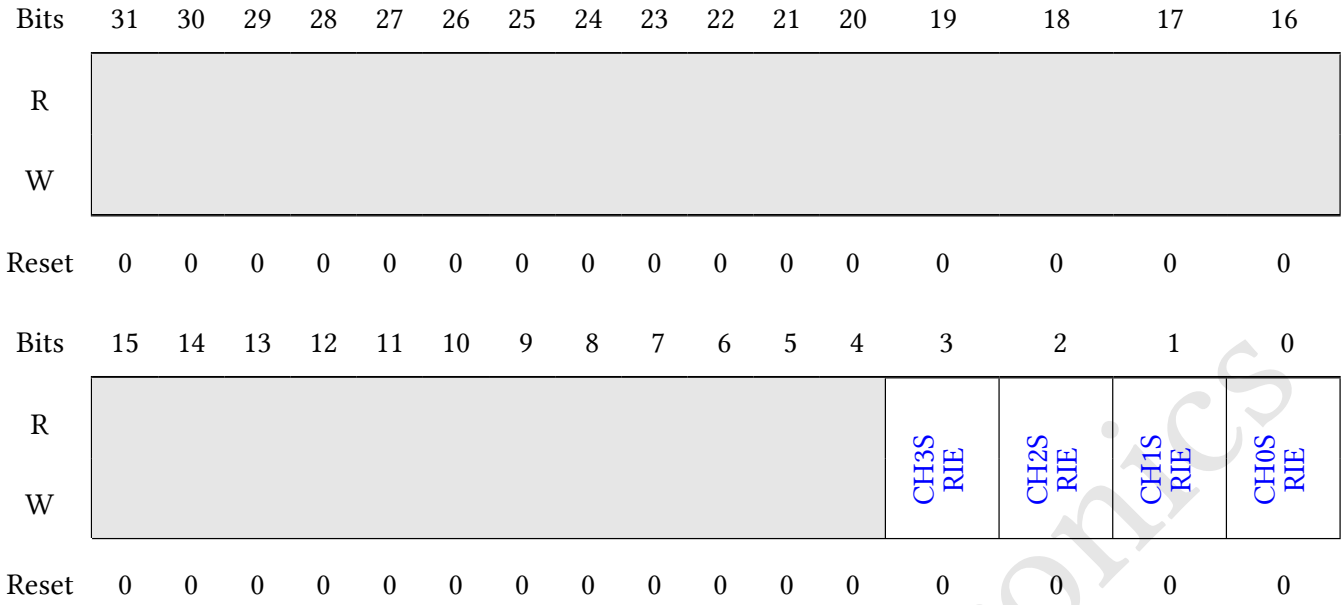


Table 25.13: SENT SINTE Register Description

Field	Function
3 CH3SRIE	**Channel 3 Slow Serial Message Ready Interrupt Enable** Channel 3 Slow Serial Message ready interrupt enable control. 0b - Disabled. 1b - Enabled.
2 CH2SRIE	**Channel 2 Slow Serial Message Ready Interrupt Enable** Channel 2 Slow Serial Message ready interrupt enable control. 0b - Disabled. 1b - Enabled.
1 CH1SRIE	**Channel 1 Slow Serial Message Ready Interrupt Enable** Channel 1 Slow Serial Message ready interrupt enable control. 0b - Disabled. 1b - Enabled.
0 CH0SRIE	**Channel 0 Slow Serial Message Ready Interrupt Enable** Channel 0 Slow Serial Message ready interrupt enable control. 0b - Disabled. 1b - Enabled.

Yuntu Microelectronics

25.3.1.9 SENT DMACTRL Register

Offset: 20h

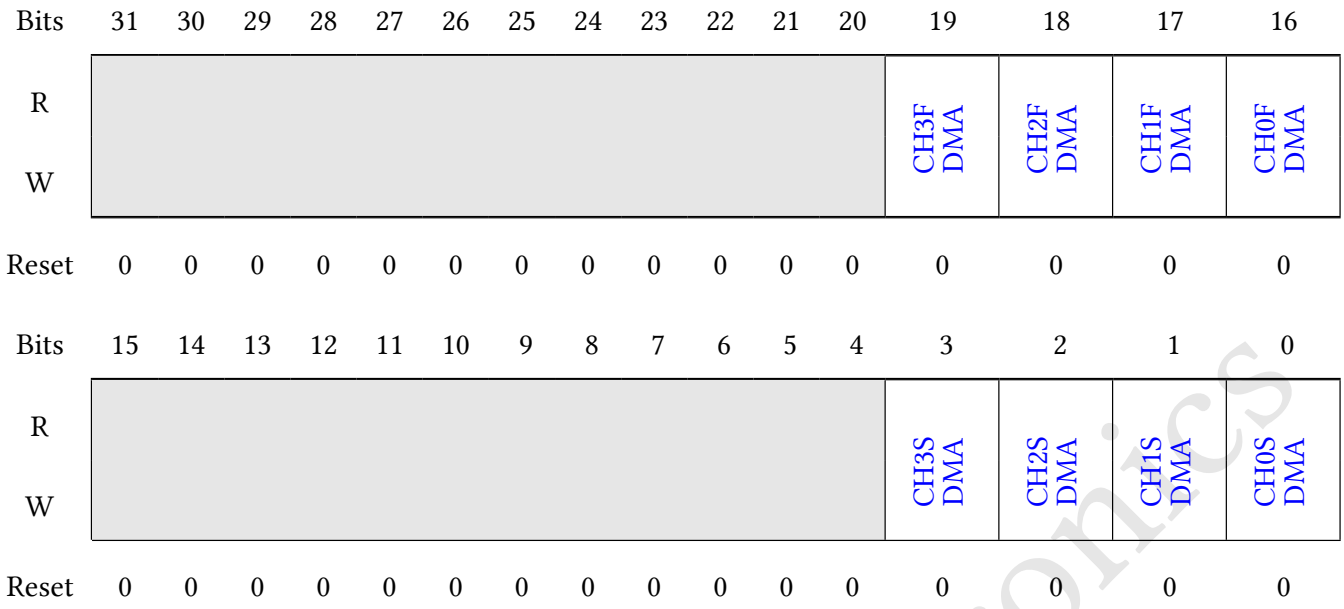


Table 25.14: SENT DMACTRL Register Description

Field	Function
19 CH3FDMA	**Channel 3 Fast Message DMA Enable** DMA enable control for Fast Message on channel 3. 0b - DMA for Fast Message is disabled. 1b - DMA for Fast Message is enabled.
18 CH2FDMA	**Channel 2 Fast Message DMA Enable** DMA enable control for Fast Message on channel 2. 0b - DMA for Fast Message is disabled. 1b - DMA for Fast Message is enabled.
17 CH1FDMA	**Channel 1 Fast Message DMA Enable** DMA enable control for Fast Message on channel 1. 0b - DMA for Fast Message is disabled. 1b - DMA for Fast Message is enabled.
16 CH0FDMA	**Channel 0 Fast Message DMA Enable** DMA enable control for Fast Message on channel 0. 0b - DMA for Fast Message is disabled. 1b - DMA for Fast Message is enabled.
3 CH3SDMA	**Channel 3 Slow Serial Message DMA Enable** DMA enable control for Slow Serial Message on channel 3. 0b - DMA for Slow Serial Message is disabled. 1b - DMA for Slow Serial Message is enabled.
2	**Channel 2 Slow Serial Message DMA Enable**

Yuntu Microelectronics

Field	Function
CH2SDMA	DMA enable control for Slow Serial Message on channel 2. 0b - DMA for Slow Serial Message is disabled. 1b - DMA for Slow Serial Message is enabled.
1 CH1SDMA	**Channel 1 Slow Serial Message DMA Enable** DMA enable control for Slow Serial Message on channel 1. 0b - DMA for Slow Serial Message is disabled. 1b - DMA for Slow Serial Message is enabled.
0 CH0SDMA	**Channel 0 Slow Serial Message DMA Enable** DMA enable control for Slow Serial Message on channel 0. 0b - DMA for Slow Serial Message is disabled. 1b - DMA for Slow Serial Message is enabled.

25.3.1.10 SENT CFG Register

Offset: 24h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TSPRSC											FIFOWM				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W												OVW	FIFO EN	FAST CLR	SLPA CK	DBGF RZ
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

Table 25.15: SENT CFG Register Description

Field	Function
31 - 24 TSPRSC	**Time Stamp Prescaler Value** Configure prescaler for internal time stamp clock. Time stamp clock = (Functional Clock)/(value+1).
20 - 16 FIFOWM	**FIFO Watermark** DMA read FIFO watermark level. Indicate the number of Fast Message to be stored in FIFO before a DMA request is asserted.

Yuntu Microelectronics

Field	Function
4 OVW	**Overwrite** Select store mode of new data when FIFO is full. Discard new data or overwrite the old data with it. 0b - Overwrite is disabled, FIFO keeps old data, new data received will be discarded. 1b - Overwrite is enabled, FIFO receives new data and overwrite the old data.
3 FIFOEN	**FIFO Enable** Internal FIFO enable control. 0b - FIFO disabled, Fast Message is read from buffer only. 1b - FIFO enabled, Fast Message can be read from FIFO.
2 FASTCLR	**Fast Clearing** Fast clearing enable control. Enable clearing of ready status bits automatically when corresponding message is read. 0b - Fast clearing is disabled. 1b - Fast clearing is enabled.
1 SLPACK	**Sleep Acknowledge** Select acknowledge the low power mode request immediately or wait until internal state clean. 0b - Wait for several Functional Clock cycles until back to the idle state, then acknowledge the low power mode request. 1b - Acknowledge the low power mode request immediately.
0 DBGFRZ	**Debug Freeze** Debug mode freeze control. Set this bit to disable module receiving under debug mode. 0b - Sent module stills work under debug mode. 1b - Sent module is disabled under debug mode.

Yuntu Microelectronics

25.3.1.11 SENT NIBCFG Register

Offset: 28h

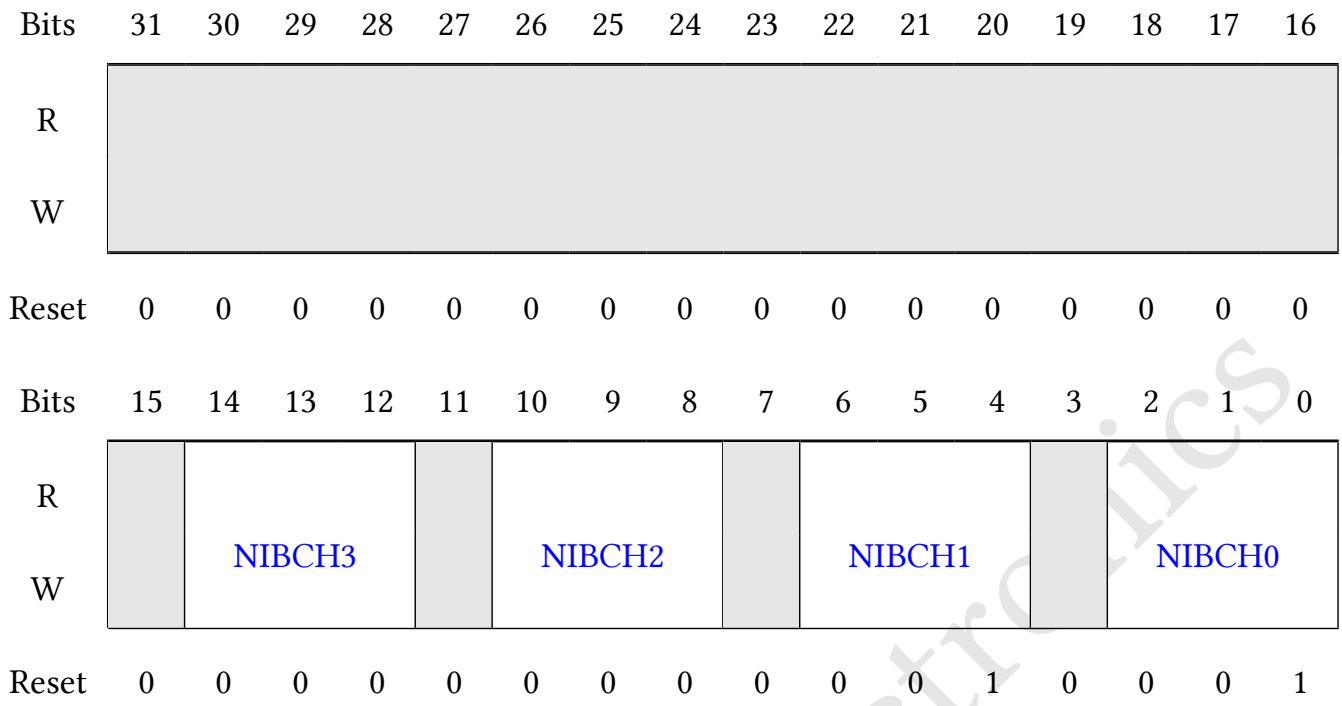


Table 25.16: SENT NIBCFG Register Description

Field	Function
14 - 12 NIBCH3	**Channel 3 Nibble Number** Number of data nibbles supported in channel 3.
10 - 8 NIBCH2	**Channel 2 Nibble Number** Number of data nibbles supported in channel 2.
6 - 4 NIBCH1	**Channel 1 Nibble Number** Number of data nibbles supported in channel 1.
2 - 0 NIBCH0	**Channel 0 Nibble Number** Number of data nibbles supported in channel 0.

Yuntu Microelectronics

25.3.1.12 SENT FMDATA Register

FIFO data output, DAM accesses register

Offset: 30h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SCNIB				DNIB0				DNIB1				DNIB2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DNIB3				DNIB4				DNIB5				FCRC			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 25.17: SENT FMDATA Register Description

Field	Function
31 - 28 SCNIB	**Status and Communication Nibble** Status and communication nibble of received Fast Message.
27 - 24 DNIB0	**Data Nibble 0** Data 0 nibble of received Fast Message.
23 - 20 DNIB1	**Data Nibble 1** Data 1 nibble of received Fast Message.
19 - 16 DNIB2	**Data Nibble 2** Data 2 nibble of received Fast Message.
15 - 12 DNIB3	**Data Nibble 3** Data 3 nibble of received Fast Message.
11 - 8 DNIB4	**Data Nibble 4** Data 4 nibble of received Fast Message.
7 - 4 DNIB5	**Data Nibble 5** Data 0 nibble of received Fast Message.
3 - 0 FCRC	**Fast Message CRC** 4-bit CRC of received Fast Message.

Yuntu Microelectronics

25.3.1.13 SENT FMCHNUM Register

FIFO data output, DAM accesses register

Offset: 34h

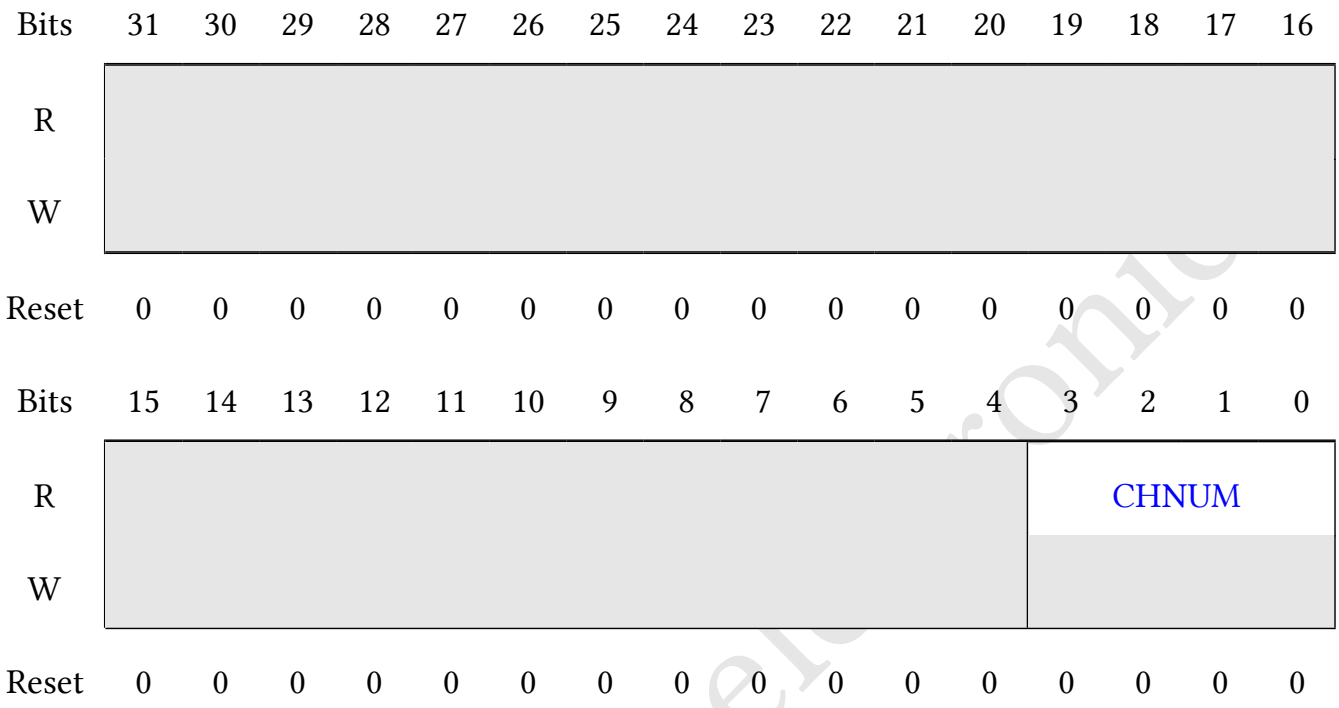


Table 25.18: SENT FMCHNUM Register Description

Field	Function
3 - 0 CHNUM	**Channel Number** Indicate the channel number of the received Fast Message.

25.3.1.14 SENT FMTS Register

FIFO data output, DAM accesses register

Offset: 38h

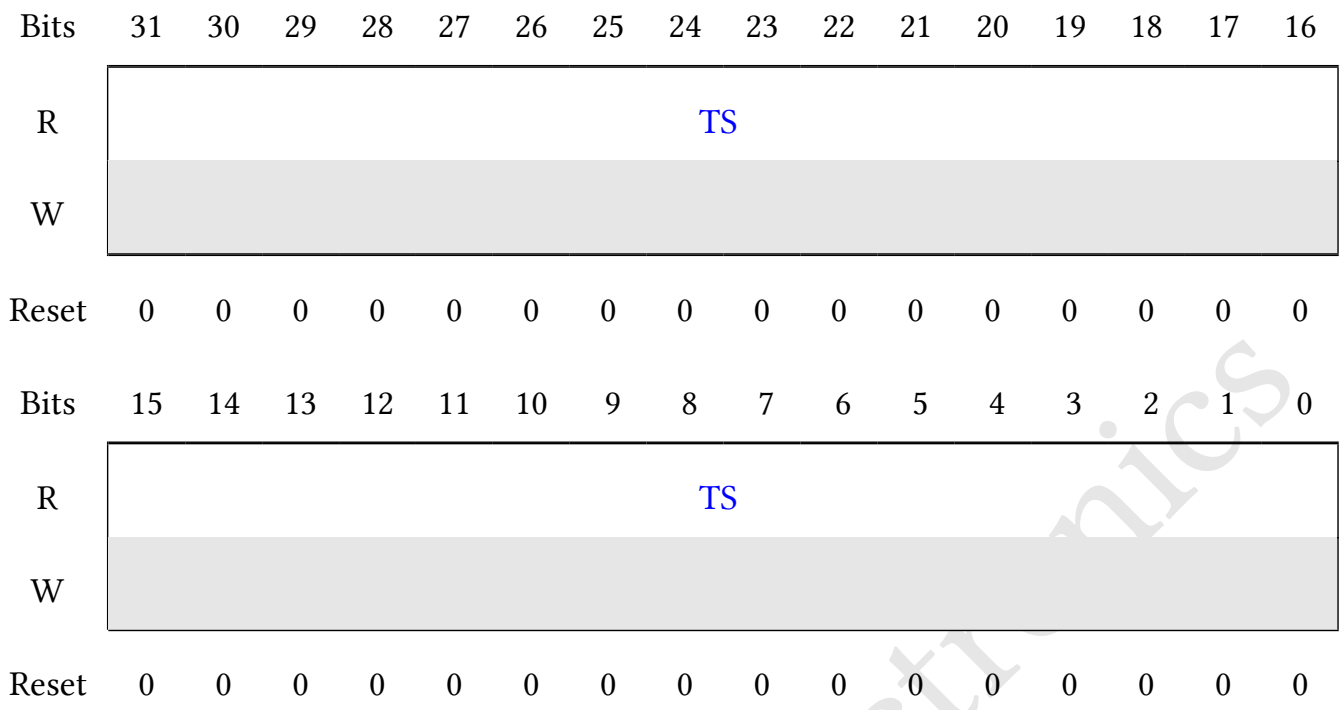


Table 25.19: SENT FMTS Register Description

Field	Function
31 - 0 TS	**Time Stamp** Indicate the time stamp of the received Fast Message.

Yuntu Microelectronics

25.3.1.15 SENT SMBIT3 Register

DAM accesses register

Offset: 3ch

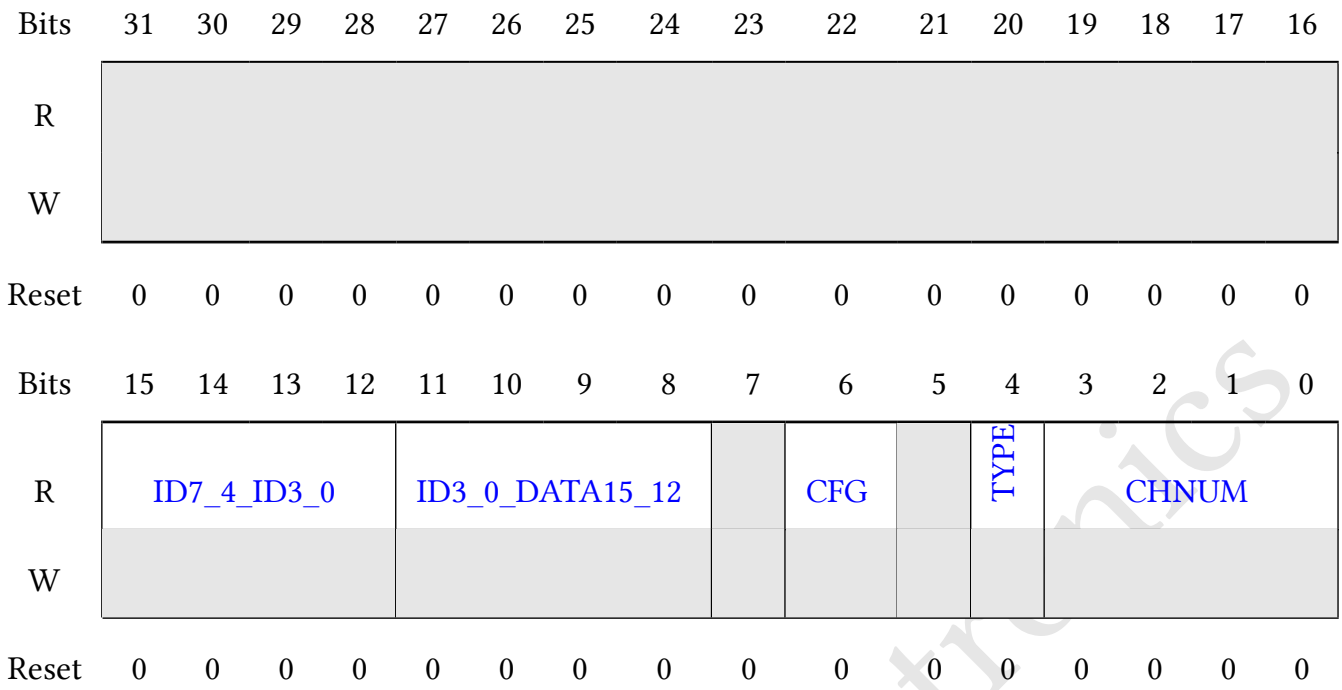


Table 25.20: SENT SMBIT3 Register Description

Field	Function
15 - 12 ID7_4_ID 3_0	**ID[7:4] or ID[3:0]** ID field. Value depends on 'C' bit. If 'C'=0, this field is ID[7:4]; If 'C'=1, this field is ID[3:0].
11 - 8 ID3_0_DA TA15_12	**ID[3:0] or DATA[15:12]** ID or Data field. Value depends on 'C' bit. If 'C'=0, this field is ID[3:0]; If 'C'=1, this field is DATA[15:12].
6 CFG	**Configuration** Configuration bit 'C'. Indicate the type of Enhances Serial Message. 0b - Enhanced Serial Message with 8-bit ID field. 1b - Enhanced Serial Message with 4-bit ID field.
4 TYPE	**Serial Message Type** Serial message type. Indicate the type of received Slow Serial Message. 0b - Short Serial Message. 1b - Enhanced Serial Message.
3 - 0 CHNUM	**Channel Number** Indicate the channel number of the received Slow Serial Message.

Yuntu Microelectronics

25.3.1.16 SENT SMBIT2 Register

DAM accesses register

Offset: 40h

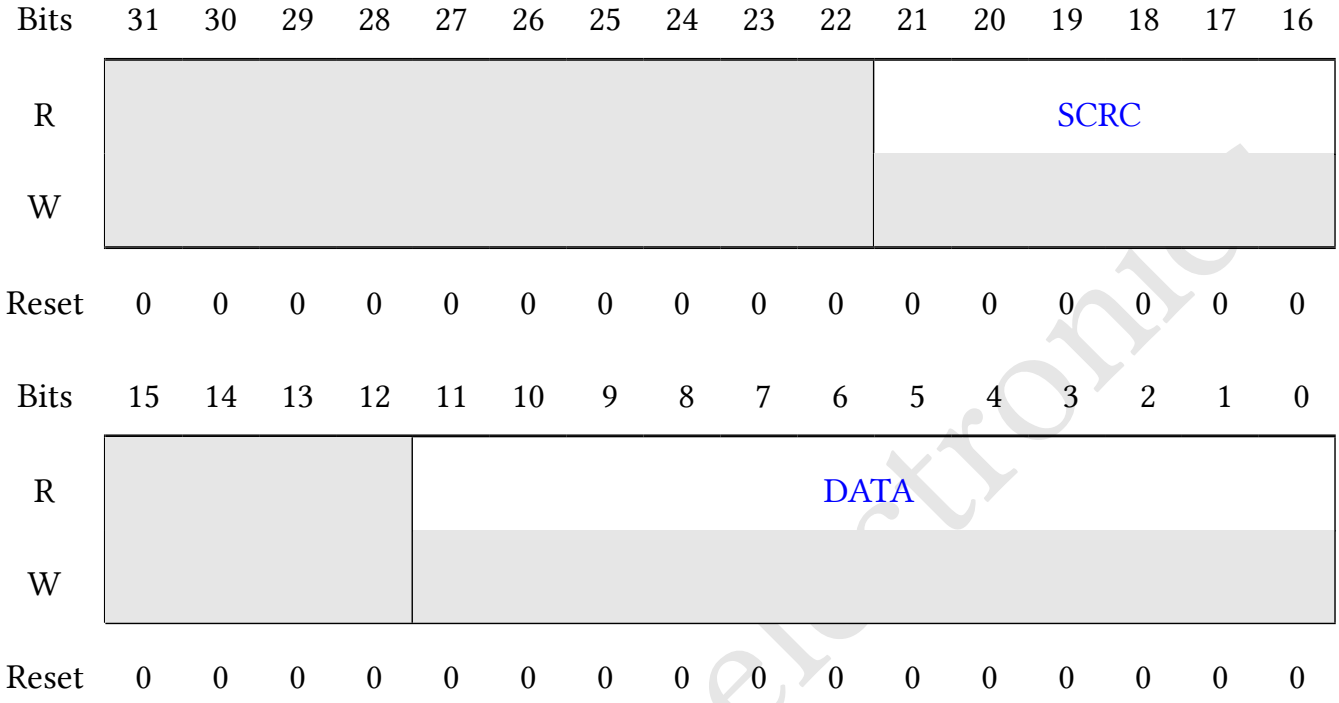


Table 25.21: SENT SMBIT2 Register Description

Field	Function
21 - 16 SCRC	**Slow Serial Message CRC** CRC value of the Slow Serial Message. When TYPE=0, this CRC is 4-bit for Short Serial Message; When TYPE=1, this CRC is 6-bit for Enhanced Serial Message.
11 - 0 DATA	**DATA[11:0]** Data field.

25.3.1.17 SENT SMTS Register

DAM accesses register

Offset: 44h

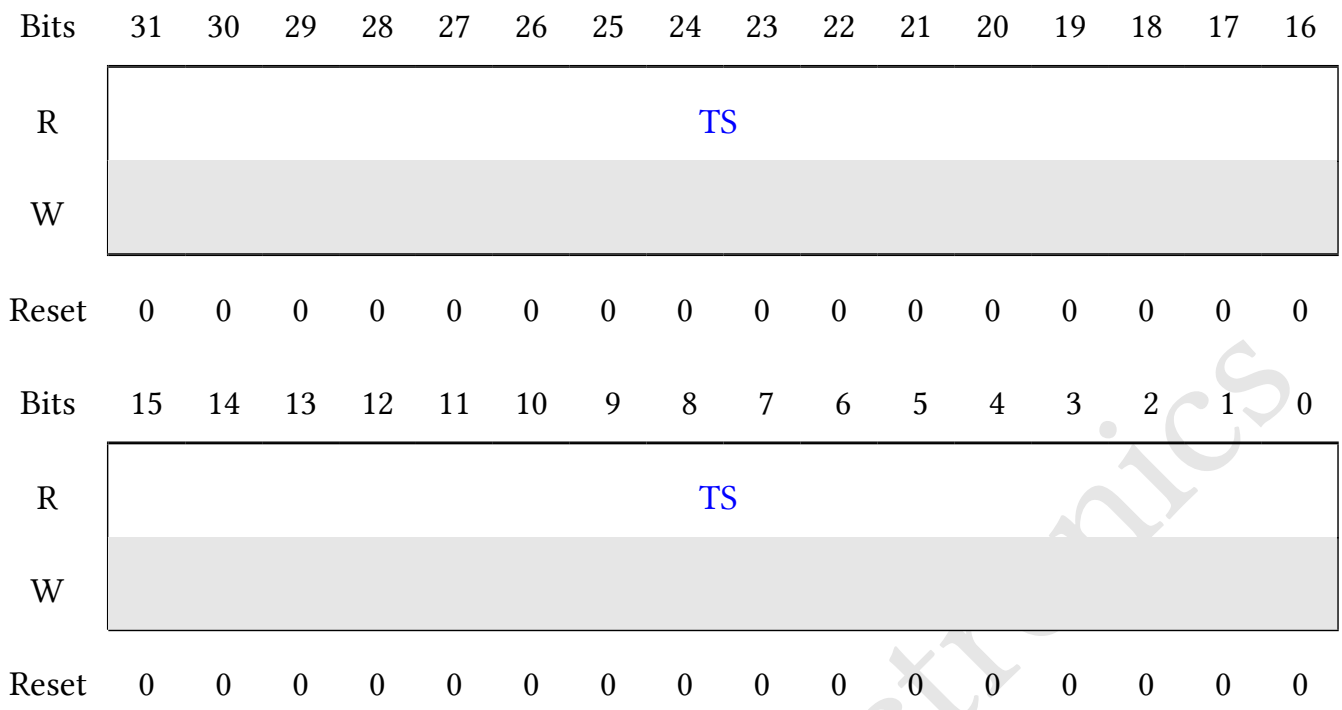


Table 25.22: SENT SMTS Register Description

Field	Function
31 - 0 TS	**Time Stamp** Indicate the time stamp of the received Slow Serial Message.

Yuntu Microelectronics

25.3.1.18 SENT CH_CLK Register

n=0,1,2,3

Offset: 50h + n * 30h

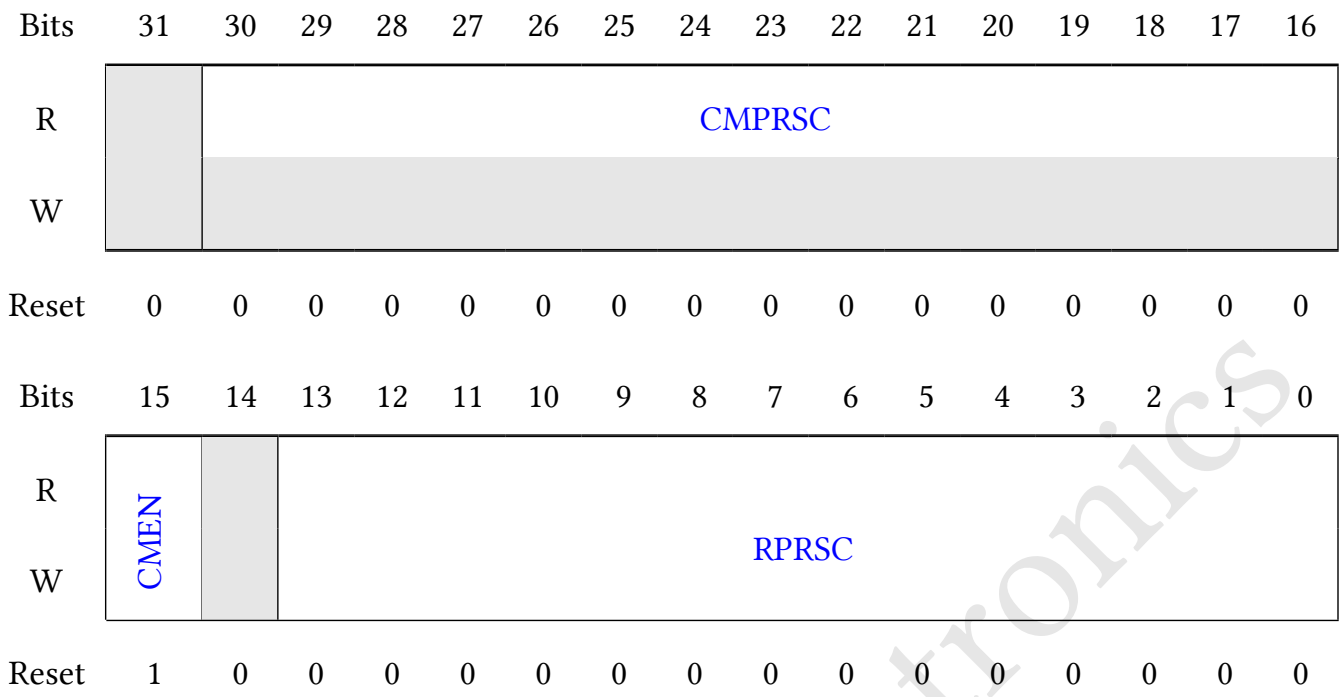


Table 25.23: SENT CH_CLK Register Description

Field	Function
30 - 16 CMPRSC	**Compensated Prescaler Value** This value is used to compensate the Tx clock variation, it's calculated by internal clock compensation logic. '= Receive Prescaler Value' No variation, no compensation or compensation is disabled '< Receive Prescaler Value' Tx clock is faster than the programmed receive clock '> Receive Prescaler Value' Tx clock is slower than the programmed receive clock
15 CMEN	**Compensation Enable** Compensation enable control. 0b - Compensation disabled. 1b - Compensation enabled.
13 - 0 RPRSC	**Receive Prescaler Value** This factor is used to generate required receive clock tick from the Functional Clock. It can only be written when channel enable bit is 0. Write 0 to bypass the prescaler, should not be used. RPRSC=(receive clock tick)/(functional clock period).

25.3.1.19 SENT CH_STS Register

n=0,1,2,3

Offset: 54h + n * 30h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R		ENUMERR	FCRCERR	SCRCERR	NVERR	CDERR	CLERR	PPDERR	FMOF	SMOF	CAL_20_25	CALRESYNC				IDLE
W		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 25.24: SENT CH_STS Register Description

Field	Function
14 ENUMERR	**Edge Number Error** Indicate the edge number error state. This error will only set when ”Option 1” for successive calibration check is selected. Write 1 to clear this bit. 0b - No error detected. 1b - Not the expected number of negative edges between calibration pulse.
13 FCRCERR	**Fast Message CRC Error** Indicate checksum error in Fast Message. Write 1 to clear this bit. 0b - No error detected. 1b - Not the expected CRC value of received Fast Message.
12 SCRCERR	**Slow Serial Message CRC Error** Indicate checksum error in Slow Serial Message. Write 1 to clear this bit. 0b - No error detected. 1b - Not the expected CRC value of received Slow Serial Message.
11 NVERR	**Nibble Value Error** Indicate nibble value error. Write 1 to clear this bit. 0b - No error detected. 1b - Invalid nibble value that < 0 or > 15 detected.
10 CDERR	**Calibration Diagnose Error** Indicate error about successive calibration pulses differ. Write 1 to clear this bit. 0b - No error detected. 1b - Successive calibration pulses differ by > +1.5625% or < -1.5625%.
9	**Calibration Length Error**

Yuntu Microelectronics

Field	Function
CLERR	Indicate calibration pulse length error. Write 1 to clear this bit. 0b - No error detected. 1b - Calibration pulse is more than 56 ticks ±25%.
8 PPDERR	**Pause Pulse Diagnose Error** Indicate the ratio of calibration pulse to overall message length error. This check is valid only for message with pause pulse. Write 1 to clear this bit. 0b - No error detected. 1b - The ratio of calibration pulse length to overall message length (with pause pulse) is more than ±1.5625%.
7 FMOF	**Fast Message Overflow** Indicate Fast Message overflow occured on this channel. Write 1 to clear this bit. 0b - No error detected. 1b - Overflow occured.
6 SMOF	**Slow Serial Message Overflow** Indicate Slow Serial Message overflow occured on this channel. Write 1 to clear this bit. 0b - No error detected. 1b - Overflow occured.
5 CAL_20_25	**Calibration Variation 20%-25%** Indicate the calibration pulse variation information. Write 1 to clear this bit. 0b - Calibration pulse variation is less than 56 ticks ±20% or more than 56 ticks ±25%. 1b - Calibration pulse variation is between ±20% to ±25% from 56 ticks.
4 CALRESYNC	**Calibration Resynchronized Error** Indicate the resynchronized error. Write 1 to clear this bit. 0b - No error detected. 1b - Successive calibration check has failed three times and assume 3rd message to be correct and will be received.
0 IDLE	**Bus Idle** Indicate channel receive idle state. 0b - Channel is busy. 1b - Channel is idle.

25.3.1.20 SENT CH_INTE Register

n=0,1,2,3

Offset: 58h + n * 30h

Yuntu Microelectronics

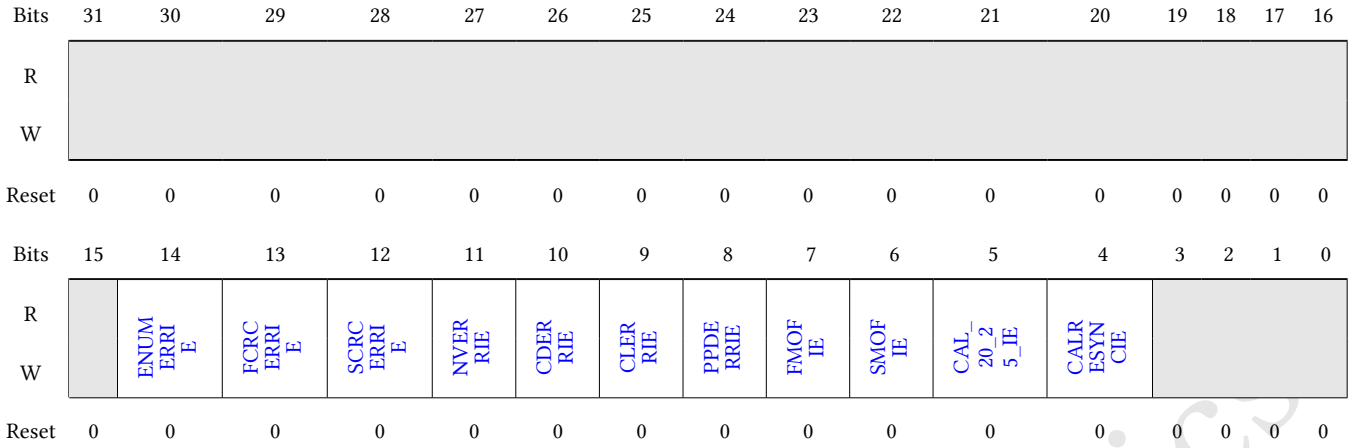


Table 25.25: SENT CH_INTE Register Description

Field	Function
14 ENUMERRIE	**Edge Number Error Interrupt Enable** Edge number error interrupt enable control. 0b - Disabled. 1b - Enabled.
13 FCRCERRIE	**Fast Message CRC Error Interrupt Enable** Fast Message checksum error interrupt enable control. 0b - Disabled. 1b - Enabled.
12 SCRCERRIE	**Slow Serial Message CRC Error Interrupt Enable** Slow Serial Message checksum error interrupt enable control. 0b - Disabled. 1b - Enabled.
11 NVERRIE	**Nibble Value Error Interrupt Enable** Nibble value error interrupt enable control. 0b - Disabled. 1b - Enabled.
10 CDERRIE	**Calibration Diagnose Error Interrupt Enable** Calibration diagnose error interrupt enable control. 0b - Disabled. 1b - Enabled.
9 CLERRIE	**Calibration Length Error Interrupt Enable** Calibration pulse length error interrupt enable control. 0b - Disabled. 1b - Enabled.
8	**Pause Pulse Diagnose Error Interrupt Enable**

Yuntu Microelectronics

Field	Function
PPDERRIE	Pause pulse diagnose error interrupt enable control. 0b - Disabled. 1b - Enabled.
7 FMOFIE	**Fast Message Overflow Interrupt Enable** Fast Message overflow interrupt enable control. 0b - Disabled. 1b - Enabled.
6 SMOFIE	**Slow Serial Message Overflow Interrupt Enable** Slow Serial Message overflow interrupt enable control. 0b - Disabled. 1b - Enabled.
5 CAL_20_25_IE	**Calibration Variation 20%-25% Interrupt Enable** Calibration variation 20%-25% interrupt enable control. 0b - Disabled. 1b - Enabled.
4 CALRESYNCIE	**Calibration Resynchronized Error Interrupt Enable** Resynchronized error interrupt enable control. 0b - Disabled. 1b - Enabled.

25.3.1.21 SENT CH_CFG Register

n=0,1,2,3

Offset: 5ch + n * 30h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W								FCRCCHK	SCCHK	PPEN	SCRCTYPE	FCRCSCEN	FCRCTYPE	PPCHKSEL	CALRNG	DCHGEN
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FILCNT												IDLECNT			
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 25.26: SENT CH_CFG Register Description

Field	Function
24 FCRCCHK	**Fast Message CRC Check** Control Fast Message CRC check on/off. If enable the check, Fast Message CRC error will break the current reception. 0b - Check is enabled. 1b - Check is disabled.
23 SCCHK	**Successive Calibration Check** Select successive calibration check type. 0b - Select option 2 (Low Latency Option) as SAE Specification. 1b - Select option 1 (Preferred but High Latency Option) as SAE Specification.
22 PPEN	**Pause Pulse Enable** Pause pulse detection enable control. 0b - Detection of pause pulse is disabled. 1b - Detection of pause pulse is enabled.
21 SCRCTYPE	**Slow Serial Message CRC Type** Select Slow Serial Message CRC check type. 0b - Recommended implementation (additional 0 data nibble XORed with the rest of the nibbles). 1b - Legacy implementation (no additional 0 data nibble XORed).
20 FCRCSCEN	**Fast Message CRC Status and Communication Nibble Enable** Select the status and communication nibble used for CRC calculation or not. 0b - Status and communication nibble not used for CRC calculation. 1b - Status and communication nibble is used for CRC calculation.
19 FCRCTYPE	**Fast Message CRC Type** Select Fast Message CRC check type. 0b - Recommended implementation (additional 0 data nibble XORed with the rest of the nibbles). 1b - Legacy implementation (no additional 0 data nibble XORed).
18 PPCHKSEL	**Pause Pulse Check Selection** Select pause pulse check type. 0b - Both successive calibration pulse check and pause pulse diagnostic are run. 1b - Only pause pulse diagnostic is run, successive calibration pulse check is disabled.
17 CALRNG	**Calibration Pulse Range** Select calibration pulse range to be check. 0b - 20% variation is acceptable. 1b - 25% variation is acceptable.
16	**Data Change Enable**

Yuntu Microelectronics

Field	Function
DCHGEN	Select if only different Fast Message data can be stored into buffer and generate an interrupt or DMA request. 0b - All Fast Messages will be received even if data nibbles are same. 1b - Only Fast Messages with different values of data nibbles will be received.
15 - 8 FILCNT	**Filter Count** Input filter sample count. Used to filter input glitch. Write non-zero value to enable the filter. Only the lowest non-zero bit is valid in this field. 00000001b (0x01) - The input data is valid only when the keep time is larger than 2^1 functional clock cycles. 00000010b (0x02) - The input data is valid only when the keep time is larger than 2^2 functional clock cycles. 00000100b (0x04) - The input data is valid only when the keep time is larger than 2^3 functional clock cycles. 00001000b (0x08) - The input data is valid only when the keep time is larger than 2^4 functional clock cycles. 00010000b (0x10) - The input data is valid only when the keep time is larger than 2^5 functional clock cycles. 00100000b (0x20) - The input data is valid only when the keep time is larger than 2^6 functional clock cycles. 01000000b (0x40) - The input data is valid only when the keep time is larger than 2^7 functional clock cycles. 10000000b (0x80) - The input data is valid only when the keep time is larger than 2^8 functional clock cycles.
3 - 0 IDLECNT	**Idle Count** Bus idle count. Used to define the maximum allowable idle period on the sensor interface of this channel. 0000b - Disabled. 0001b - 127*2 receiver clock tick counts. 0010b - 127*4 receiver clock tick counts. 0100b - 127*8 receiver clock tick counts. 1000b - 127*16 receiver clock tick counts.

Yuntu Microelectronics

25.3.1.22 SENT CH_FMDATA Register

n=0,1,2,3

Offset: 60h + n * 30h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SCNIB				DNIB0				DNIB1				DNIB2			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DNIB3				DNIB4				DNIB5				CRC4			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 25.27: SENT CH_FMDATA Register Description

Field	Function
31 - 28 SCNIB	**Status and Communication Nibble** Status and communication nibble of received Fast Message.
27 - 24 DNIB0	**Data Nibble 0** Data 0 nibble of received Fast Message.
23 - 20 DNIB1	**Data Nibble 1** Data 1 nibble of received Fast Message.
19 - 16 DNIB2	**Data Nibble 2** Data 2 nibble of received Fast Message.
15 - 12 DNIB3	**Data Nibble 3** Data 3 nibble of received Fast Message.
11 - 8 DNIB4	**Data Nibble 4** Data 4 nibble of received Fast Message.
7 - 4 DNIB5	**Data Nibble 5** Data 0 nibble of received Fast Message.
3 - 0 CRC4	**CRC-4** 4-bit CRC of received Fast Message.

Yuntu Microelectronics

25.3.1.23 SENT CH_FMCHNUM Register

n=0,1,2,3

Offset: 64h + n * 30h

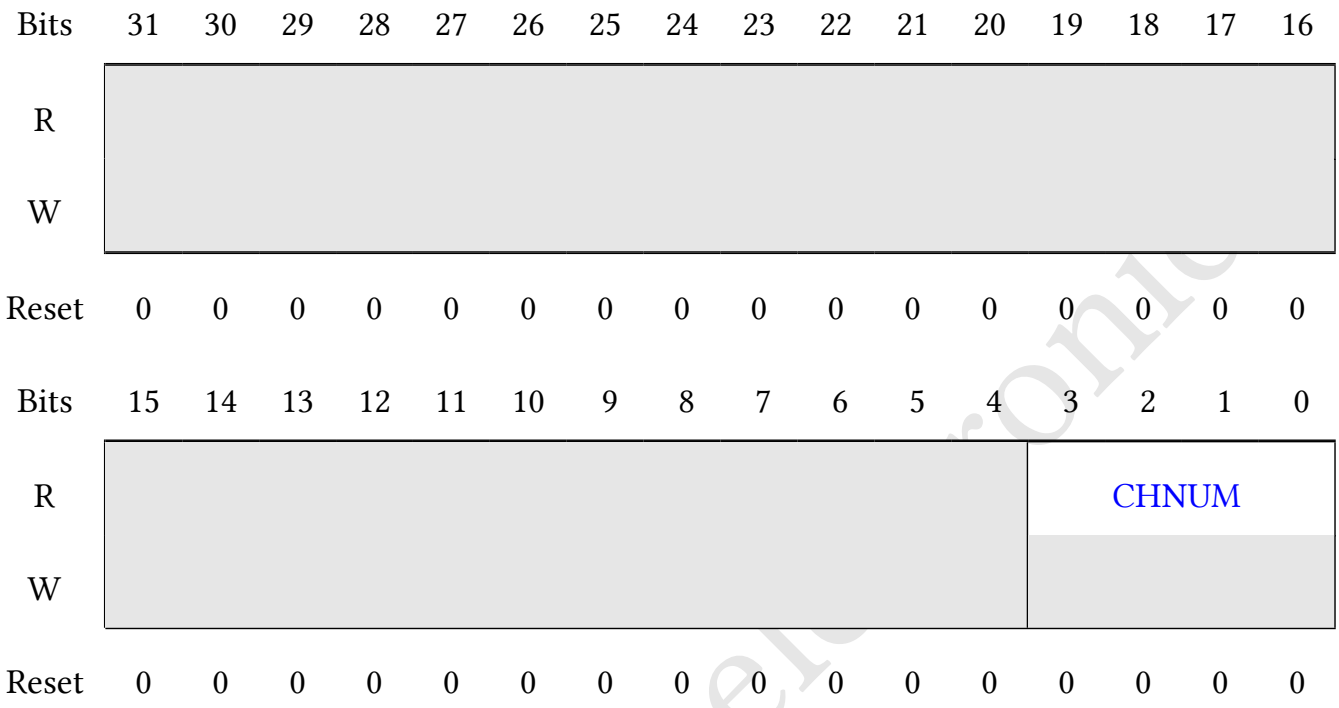


Table 25.28: SENT CH_FMCHNUM Register Description

Field	Function
3 - 0 CHNUM	**Channel Number** Indicate the channel number of the received Fast Message.

25.3.1.24 SENT CH_FMTS Register

n=0,1,2,3

Offset: 68h + n * 30h

Yuntu Microelectronics

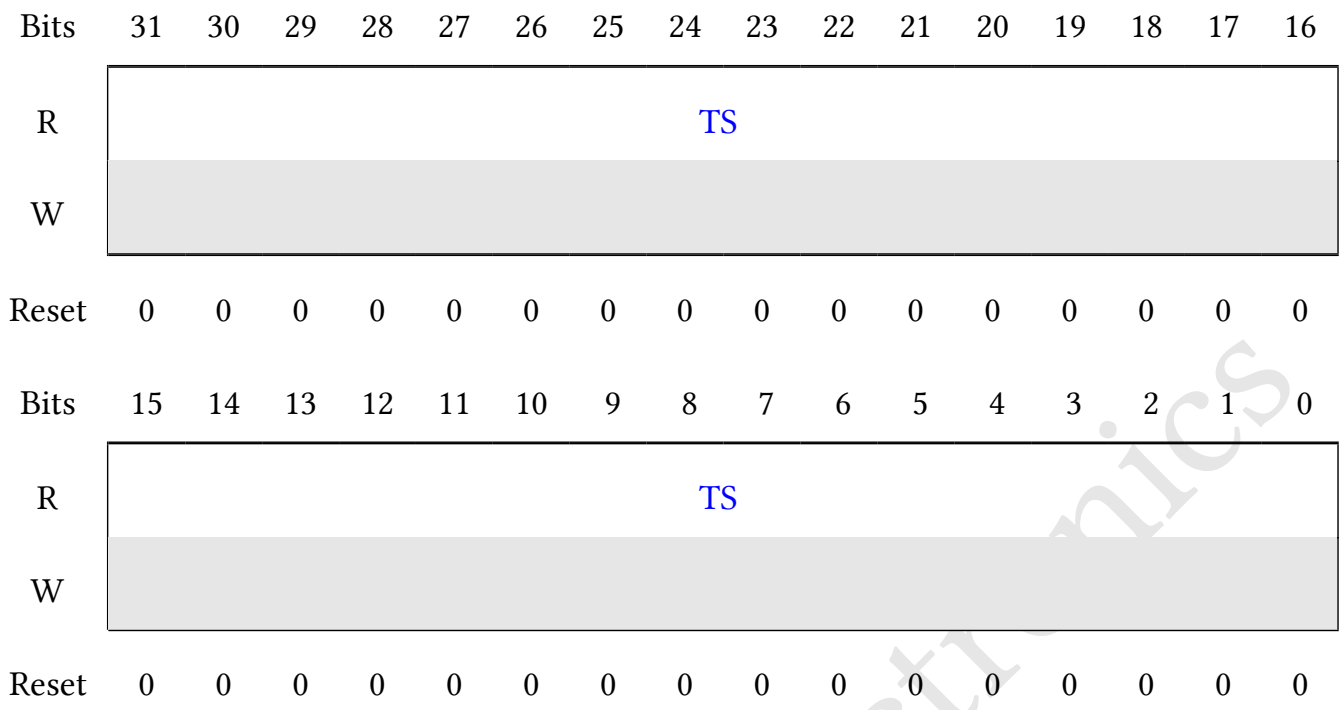


Table 25.29: SENT CH_FMTS Register Description

Field	Function
31 - 0 TS	**Time Stamp** Indicate the time stamp of the received Fast Message.

Yuntu Microelectronics

25.3.1.25 SENT CH_SMBIT3 Register

n=0,1,2,3

Offset: 6ch + n * 30h

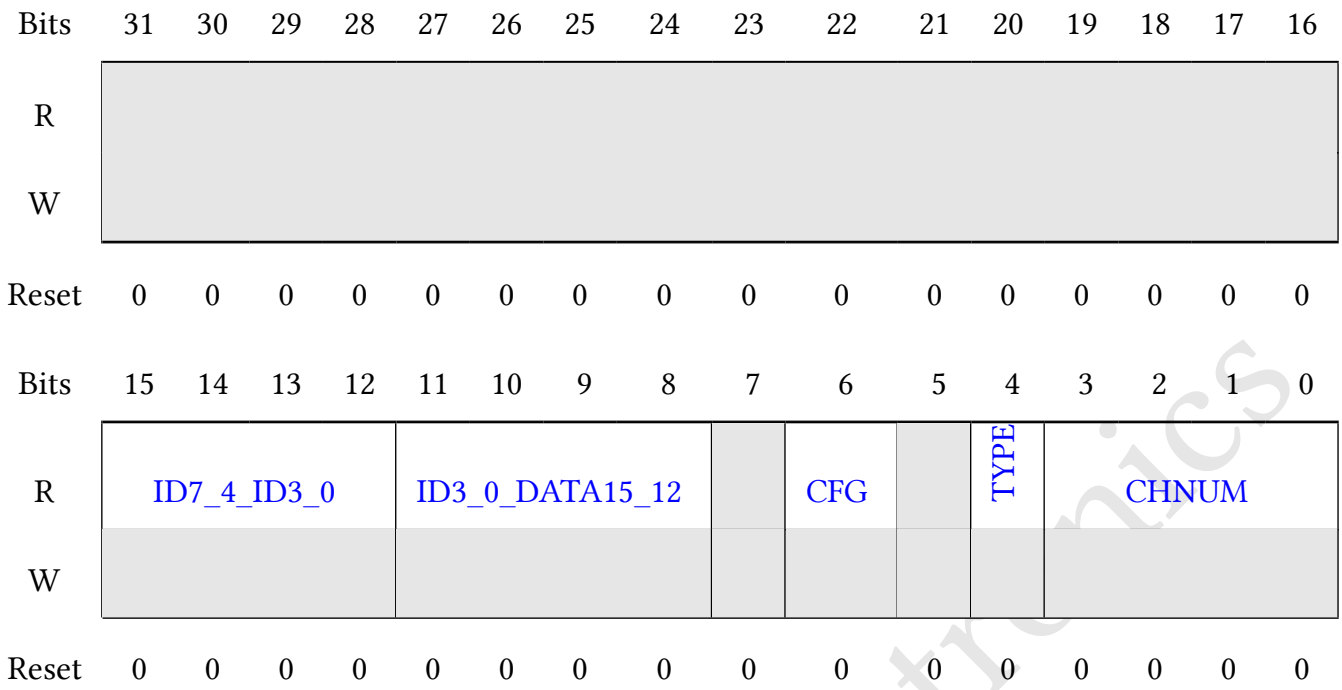


Table 25.30: SENT CH_SMBIT3 Register Description

Field	Function
15 - 12 ID7_4_ID 3_0	**ID[7:4] or ID[3:0]** ID field. Value depends on 'C' bit. If 'C'=0, this field is ID[7:4]; If 'C'=1, this field is ID[3:0].
11 - 8 ID3_0_DA TA15_12	**ID[3:0] or DATA[15:12]** ID or Data field. Value depends on 'C' bit. If 'C'=0, this field is ID[3:0]; If 'C'=1, this field is DATA[15:12].
6 CFG	**Configuration Bit** Configuration bit 'C'. Indicate the type of Enhances Serial Message. 0b - Enhanced Serial Message with 8-bit ID field. 1b - Enhanced Serial Message with 4-bit ID field.
4 TYPE	**Serial Message Type** Serial message type. Indicate the type of received Slow Serial Message. 0b - Short Serial Message. 1b - Enhanced Serial Message.
3 - 0 CHNUM	**Channel Number** Indicate the channel number of the received Slow Serial Message.

Yuntu Microelectronics

25.3.1.26 SENT CH_SMBIT2 Register

n=0,1,2,3

Offset: 70h + n * 30h

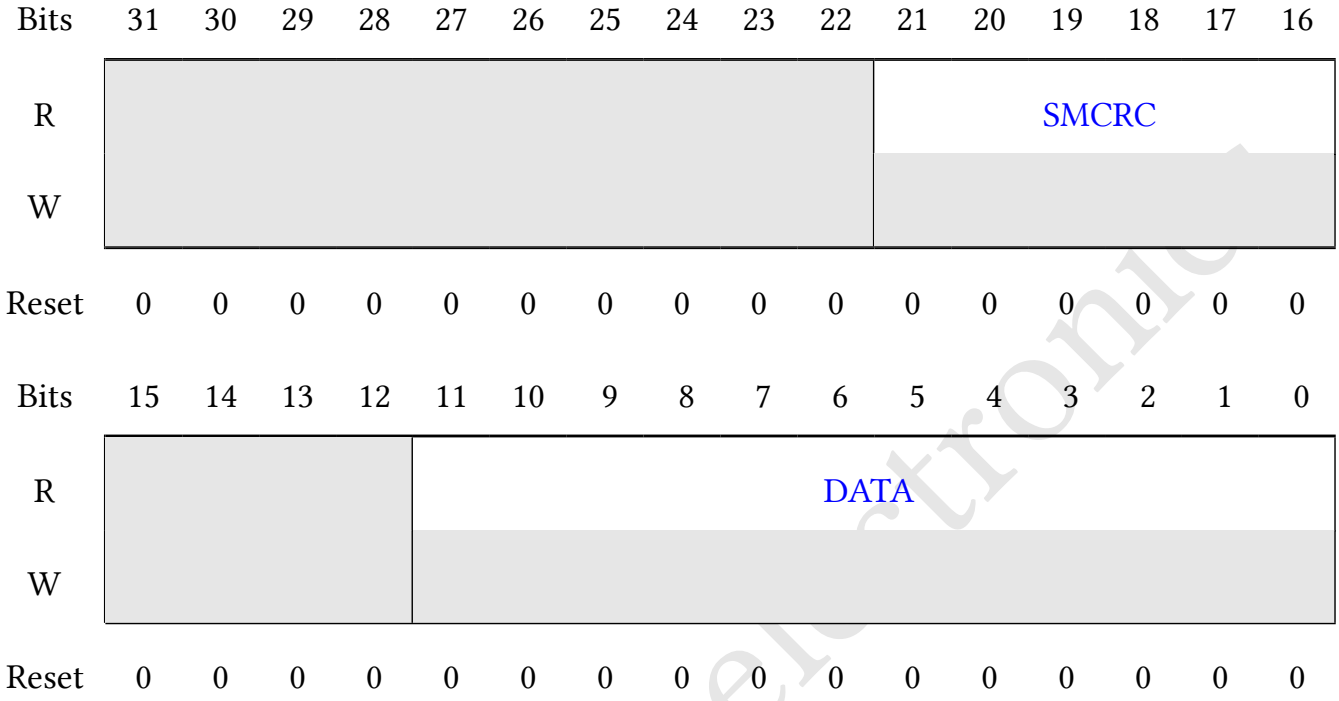


Table 25.31: SENT CH_SMBIT2 Register Description

Field	Function
21 - 16 SMCRC	**Slow Serial Message CRC** CRC value of the Slow Serial Message. When TYPE=0, this CRC is 4-bit for Short Serial Message; When TYPE=1, this CRC is 6-bit for Enhanced Serial Message.
11 - 0 DATA	**DATA[11:0]** Data field.

25.3.1.27 SENT CH_SMTS Register

n=0,1,2,3

Offset: 74h + n * 30h

Yuntu Microelectronics

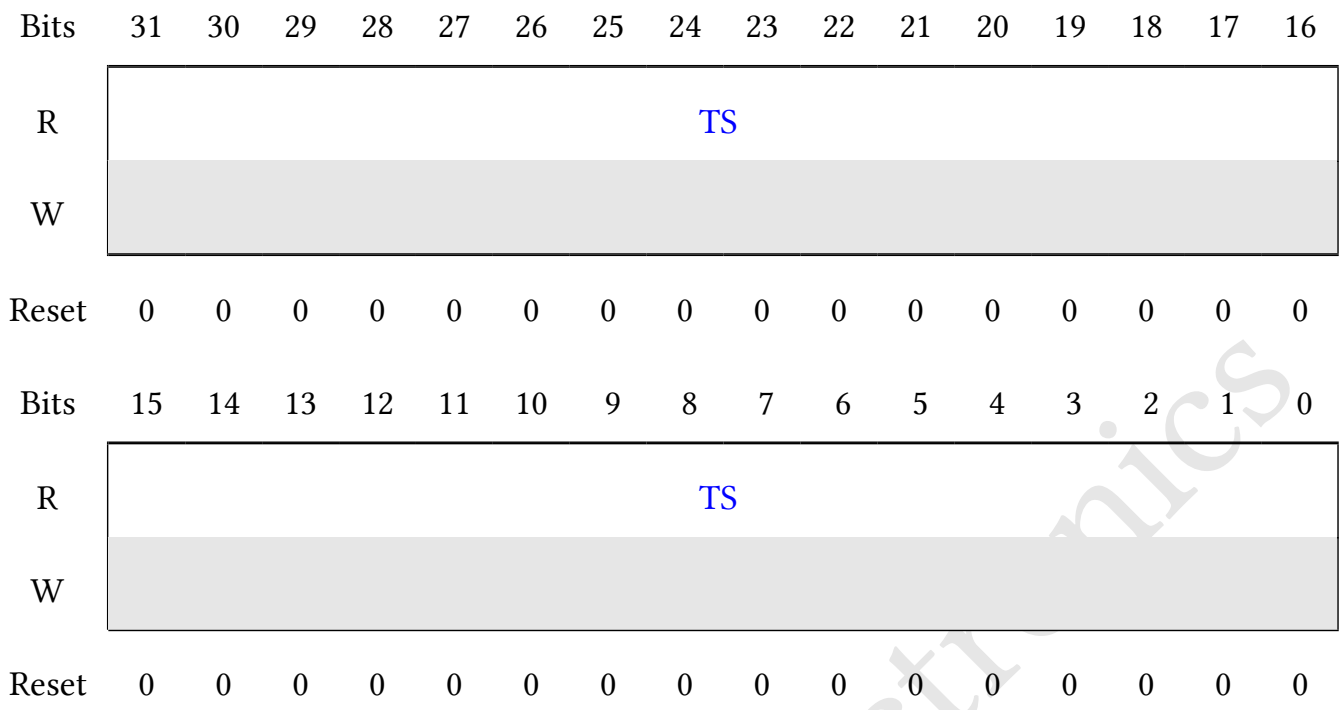


Table 25.32: SENT CH_SMTS Register Description

Field	Function
31 - 0 TS	**Time Stamp** Indicate the time stamp of the received Slow Serial Message.

25.4 Functional Description

The SENT module is a unified receiver module comprising several receiver channels, and the messages from each channel are accessible for DMA and register space. It's flexible to receive Fast Message and Slow Serial Message and store message data into separate buffers, and FIFO is configurable enabled but only for Fast Message storing.

Fast Message indicates the single message that consist of 154-270 clock ticks, Slow Serial Message includes the Short Serial Message and the Enhanced Serial Message.

25.4.1 Data Receive

Each channel receives messages independently, and it's permitted that apply different receive configurations to each channel. Configure CHCTRL[CHnEN] to enable or disable corresponding channel.

25.4.1.1 Input Filter

The input filter can be programmed to ensure that only valid inputs from pin can be received by SENT module, and each channel has a separate input filter. The input filter works based on the Functional Clock, and pin inputs is synchronized by Functional Clock firstly. Program non-zero value into

Yuntu Microelectronics

CHnCFG[FILCNT] to select the input data keep time, any input that less than the keep time will be filtered. The filter is bypassed when CHnCFG[FILCNT] = 0.

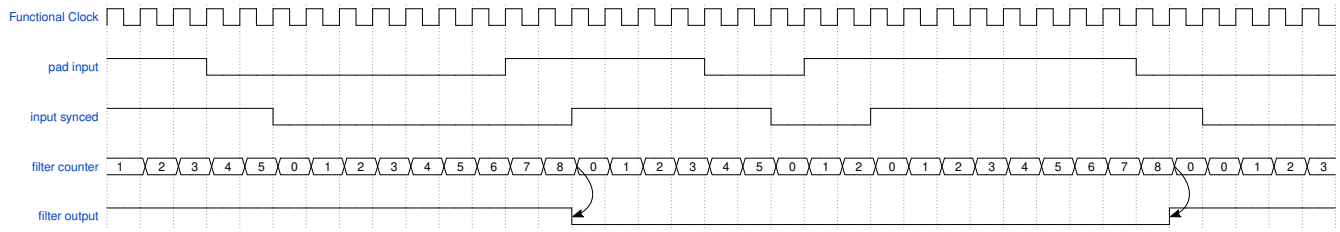


Figure 25.2: SENT Input Filter Timing Diagram

25.4.1.2 Rx Clock Generation and Adjustment

Each channel of the SENT module is composed of a clock generation and adjustment block to generate the receiver clock and adjust it for frequency variation in the transmitter clock. User needs to configure the receive clock tick that correspond with the transmitter before starting data receiving, and the compensation logic can adjust the variation of transmit clock tick. Configure CHnCLK[RPRSC] to decide the initial receive clock tick, and the adjusted receive clock tick is available from CHnCLK[CMPRSC]. The compensation logic is enabled by setting CHnCLK[CMEN].

25.4.1.3 Received Message Diagnostics

The received data can only be stored after various diagnostics. The following checks are conducted by the receiver in each channel,

- Calibration pulse length < 56 clock ticks − 25% or > 56 clock ticks + 25%.
- Not the expected number of falling edges between calibration pulses.
- Checksum error.
- Any nibble data values measured as < 0 or > 15.
- Successive calibration pulses differ by > +1.5625% (1/64) or < -1.5625%.
- For message with pause pulse, ratio of calibration pulse to message length varies by >1.5625% or <-1.5625%.

If any of the above errors detected, the receiver rejects the current data until a new calibration pulse is detected. The error status will display on the CHnSTS registers.

25.4.1.3.1 Calibration Pulse Length Check

When the channel receive a calibration pulse of length between 42 and 70 tick counts, this check will set the CHnSTS[CLERR] bit.

25.4.1.3.2 Number of Falling Edge Check

The number of negative edges between two successive calibration pulses is counted and compared with expected number of edges. The expected number of nibbles is the sum of following:

- Status and Communication nibble
- Data nibbles (programmable)
- CRC nibble
- Pause Pulse (configurable)

Yuntu Microelectronics

When the channel detects unexpected falling edge number between two successive calibration pulses, this check will set the CHnSTS[ENUMERR] bit. The check does not run when low latency of successive calibration pulse check is selected (CHnCFG[SCCHK]=0). The data nibble number of each channel is configured by NIBCFG[NIBCHn].

25.4.1.3.3 Checksum Check

The checksum check will compute both 4-bit and 6-bit CRC based on previous received data and compare the result will the received CRC value. Configure CHnCFG[FCRCTYPE] and CHnCFG[SCRCTYPE] to select the type of the Fast Message and the Slow Serial Message CRC computation (Legacy Implementation or Recommended Implementation). The status and communication nibble will be included in the 4-bit CRC calculation for Fast Serial Messages when CHnCFG[FCRCSCEN] is set.

When CRC error detected, this check will set CHnSTS[FCRCERR] or CHnSTS[SCRCERR] for Fast Message or Slow Serial Message. The checksum check of Fast Message can be disabled by clearing CHnCFG[FCRCCHK].

25.4.1.3.4 Nibble Value Check

This check is used for the status and communication nibble, all data nibbles and CRC nibble. When the channel receive a nibble which length is < 12 ticks or > 27 ticks, this check will set the CHnSTS[NVERR] bit.

25.4.1.3.5 Successive Calibration Pulse Check

The successive calibration pulse check checks if successive calibration pulses differ by > +1.5625% (1/64) or < -1.5625%. The accuracy of this check will depend on the frequency of the Functional Clock used. If check fail, this check will set the CHnSTS[CDERR] bit. When Option 2 of successive calibration pulse check is enabled, the CHnSTS[CALRESYNC] bit will be asserted on every third successive error detected by this check.

25.4.1.3.6 Pause Pulse Diagnostic

The pause pulse diagnostic checks whether the ratio of calibration pulse length to full message length (including pause pulse) varies from one message to another by less than 1.5625%. When this check fails, the status bit CHnSTS[PPERR] will be asserted. This diagnostic will not discard current message. The successive calibration pulse check can be configured to operate along with this check, setting CHnCFG[PPCHK] to enable it. This function is added to avoid conflicting results from the two checks.

Setting CHnCFG[PPEN] to enable the reception of message with pause pulse.

25.4.1.4 Time Stamp Logic

An internal 32-bit counter maintains the time stamp for all channels which can be used by the software to determine the relative time of arrival of the messages for each channel. It's an always on free-running counter after module enabled (CTRL[EN]=1), and it only stops under debug mode or low power mode. The clock for this counter is generated from the Functional Clock, and a 8-bit prescaler is available for the clock (CFG[TSPRSC]).

Yuntu Microelectronics

25.4.1.4.1 Limitation

Since the same counter is used for all channels, it may happen that more than one message is received within the same time stamp across different channels. Hence the software would not be able to determine the sequence of arrival of messages which have the same time stamp value.

25.4.1.5 Data Store and Access

These messages that pass the diagnostics will be stored into the buffers of each channel. The Fast Massage information are stored into CHnFMDATA, CHnFMCHNUM and CHnFMTS registers. The Slow Serial Message information are stored into CHnSMBIT3, CHnSMBIT2 and CHnSMTS registers. CHNUM field in CHnFMCHNUM or CHnSMCHNUM is the channel number information of the receive message. The TS field in CHnFMST or CHnSMST is the time stamp value when the status and communication nibble received. The channel received data ready flag (FRDY[CHnFRDY] or SRDY[CHnSRDY]) will be set at the same time that data information stored into buffers. If new data received and previous data hasn't been read, overflow flag (CHnSTS[FMOF] or CHnSTS[SMOF]) will be set.

If FIFO logic is enabled (CFG[FIFOEN]=1), the Fast Message information are stored into both FIFO and buffers. The FIFO data can be read by accessing FMDATA, FMCHNUM and FMTS registers. The FIFO output data pointer moves to the next data only after reading FMTS register. The FIFO overflow flag (GSTS[FMFO]) will be set if receive new data received when FIFO is full. There is also a control bit (CFG[OVW]) which can select the store way of new data when FIFO is full.

Channel buffer registers (CHnFMDATA, CHnFMCHNUM, CHnFMTS, CHnSMBIT3, CHnSMBIT2, CHnSMTS) will update when new data are received by corresponding channel. Common buffer registers (FMDATA, FMCHNUM, FMTS, SMBIT3, SMBIT2, SMTS) are only used together with DMA, and the data only come from channels which enable the DMA function can be stored into the FIFO. See "DMA Usage" for details.

25.4.2 Modes of Operation

25.4.2.1 Debug Mode

In debug mode, message reception will be halted (without waiting for the current message receive completed), and all existing status and messages will be retained. CFG[DBGFRZ] bit is required to be set for this mode. After debug exiting, a calibration length error (STS[CLERR]) can be reported and the channels will start receiving messages from a valid calibration pulse.

25.4.2.2 Low Power Mode

The SENT module behavior is affected by chip-specific low power modes (Deepsleep/Standby/Powerdown). These low power modes may gate-off both the Bus Clock and the Functional Clock, depending on the application. SENt module is disabled and can not receive new data under these low power mode.

When the Bus Clock is gated, writing access to registers will be ignored and registers' status will not update.

When the Functional Clock is gated, the module will not acknowledge the low power mode request correctly when CFG[SLPACK] = 0, this bit is required to be set in this case.

The SENT module is configurable to wake up chip from low power mode, it's required to set GINTE[WUIE] for this function, then any valid negative edge from input pin will generate a interrupt,

Yuntu Microelectronics

25.4.3 Interrupts

This module provide various interrupts but they share the same interrupt number.

GINTE register controls the enable/disable of wake up interrupt, overflow interrupt and underflow interrupt.

FINTE register controls the enable/disable of Fast Message data ready interrupt.

SINTE register controls the enable/disable of Slow Serial Message data ready interrupt.

CHnINTE registers control the enable/disable of channels' error interrupt.

25.4.4 DMA Usage

Users can access the received data by DMA, and each channel has it's own DMA enable control bit. DMA[CHnFDMA] bits are used to enable/disable the Fast Message DMA request, and DMA[CHnSDMA] bits are used to enable/disable the Slow Serial Message DMA request. DMA accessible registers are FMDATA, FMCHNUM, FMTS, SMBIT3, SMBIT2 and SMTS.

When FIFO logic is enabled (CFG[FIFOEN]=1), the watermark (CFG[FIFOWM]) is used to trigger DMA request. FIFO triggers DMA request is disabled when CFG[FIFOWM]=0. When FIFO logic is disabled (CFG[FIFOEN]=0), the channel data ready flag is used to trigger DMA request.

25.4.4.1 FIFO Access

A single DMA request is generated for all Fast Message received on any channel. There is an internal FIFO to store the message from all channels and allow them to be read by DMA. The fixed round robin sequencing logic is applied to search the channel buffer one by one and store them into the FIFO. The first round robin starts from channel 0, and the sequence order is 0, 1, ..., n-1, n. Each time a new data received, the round robin starts from the next channel and check buffer one by one until getting the buffer data.

For example, if the first received data from channel 2 (assume total channel number is 4), then the round robin sequencing logic will search channel 0 buffer, channel 1 buffer and channel 2 buffer, channel 2 buffer is not empty and data will be stored into the FIFO. If the next received data from channel 1, then the round robin sequencing logic will search channel 3 buffer, channel 0 buffer and channel 1 buffer, channel 1 buffer is not empty and data will be stored into the FIFO. If data received during the round robin sequencing logic searching and data from the previous channel, the later data will be stored into FIFO first.

25.4.4.2 Buffer Access

When FIFO logic is disabled (CFG[FIFOEN]=0), DMA accesses to FMDATA, FMCHNUM and FMTS will get the buffer data. Each time a new data received, the data will keep until DMA read it. If several channels received data before DMA reading, the first data will keep until being read. After reading the first data, the round robin sequencing logic will search the next buffer based on the sequence order.

25.4.4.3 Error Access

Reading access to FMDATA, FMCHNUM, FMTS registers when no data received, underflow flag (GSTS[FMDU]) will be set. Reading access to SMBIT3, SMBIT2 and SMTS registers when no data received, underflow flag (GSTS[SMDU]) will be set.

Yuntu Microelectronics

25.4.5 Initialization Flow

1. Select the Functional Clock for the SENT module, configure IPC register
2. Configure the receive clock tick according to the transmitter, configure CHnCLK register
3. Configure the channel nibble number according to the transmitter, configure NIBCFG register
4. Enable FIFO if need, configure CFG register
5. Enable interrupt or DMA if need, configure FINTE, SINTE and DMA registers
6. Enable channel that ready for data receiving, configure CHCTRL
7. Enable module, configure CTRL

25.5 Glossary

CRC Cyclic Redundancy Check

Yuntu Microelectronics

Chapter 26

Serial Audio Interface (SAI)

26.1 Chip-specific Information

26.1.1 Instantiation Information

SAI0 has 4 data lines while SAI1 has 1 data line.

26.1.2 MCLK Sources

Table 26.1: SAI0 MCLK sources table

MCLK	Clock Source	Description
mclk[0]	pclk	Bus clock
mclk[1]	fxosc	Fast external oscillator clock
mclk[2]	pll	PLL clock
mclk[3]	sai1_mclk_ind	SAI1 MCLK input from PAD

Table 26.2: SAI1 MCLK sources table

MCLK	Clock Source	Description
mclk[0]	pclk	Bus clock
mclk[1]	fxosc	Fast external oscillator clock
mclk[2]	pll	PLL clock
mclk[3]	sai0_mclk_ind	SAI0 MCLK input from PAD

26.1.3 DMA Description

Table 26.3: SAI DMA Table

Interrupt Source	Vector	IRQn	Wakeup
SAI0 interrupt	226	210	Y
SAI1 interrupt	227	211	Y

Yuntu Microelectronics

26.1.4 Interrupt Description

Table 26.4: SAI Interrupt Table

DMA Request Source	DMA Channel	DMA Request Description
SAI0_RX_CH0	116	SAI0 Channel 0 DMA receive request
SAI0_RX_CH1	117	SAI0 Channel 1 DMA receive request
SAI0_RX_CH2	118	SAI0 Channel 2 DMA receive request
SAI0_RX_CH3	119	SAI0 Channel 3 DMA receive request
SAI0_TX_CH0	120	SAI0 Channel 0 DMA transmit request
SAI0_TX_CH1	121	SAI0 Channel 1 DMA transmit request
SAI0_TX_CH2	122	SAI0 Channel 2 DMA transmit request
SAI0_TX_CH3	123	SAI0 Channel 3 DMA transmit request
SAI1_RX_CH0	124	SAI1 Channel 0 DMA receive request
SAI1_TX_CH0	125	SAI1 Channel 0 DMA transmit request

26.1.5 Power Mode Description

Table 26.5: SAI Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Keeps working when function clock available
Deepsleep	Keeps working when function clock available
Standby	Function disabled
Powerdown	Powered off

26.2 Introduction

26.2.1 Overview

The SAI (Serial Audio Interface) has a half-duplex serial interface with various frame synchronization formats. It supports many stereo or mono audio applications like I2S(both LSB and MSB-justified), AC'97, TDM and PCM/DSP.

26.2.2 Features

- Transmitter and receiver have independent bit clock and frame sync.
- Transmitter and receiver support up to 4 data lines.
- The word size is from 8 to 32 bits. The first word size can be set separately from other words.
- The maximum frame size is 16 words.
- Each transmit and receive data line has an 8 x 32 bit FIFO.
  - Automatic graceful restart after FIFO error.
  - Support packing 8-bit and 16-bit data into 32-bit FIFO words.
  - Support combining FIFOs of multiple data lines into a single data line FIFO
- Asynchronous interrupt for both transmitter and receiver, these interrupt sources are supported:
  - Word start interrupt
  - Sync error interrupt

Yuntu Microelectronics

  - FIFO error, warning and request interrupt
- Support DMA
  - Each data line has independent DMA enable and DMA request.
  - Two DMA request sources, FIFO warning and FIFO request.

26.2.3 Block Diagram

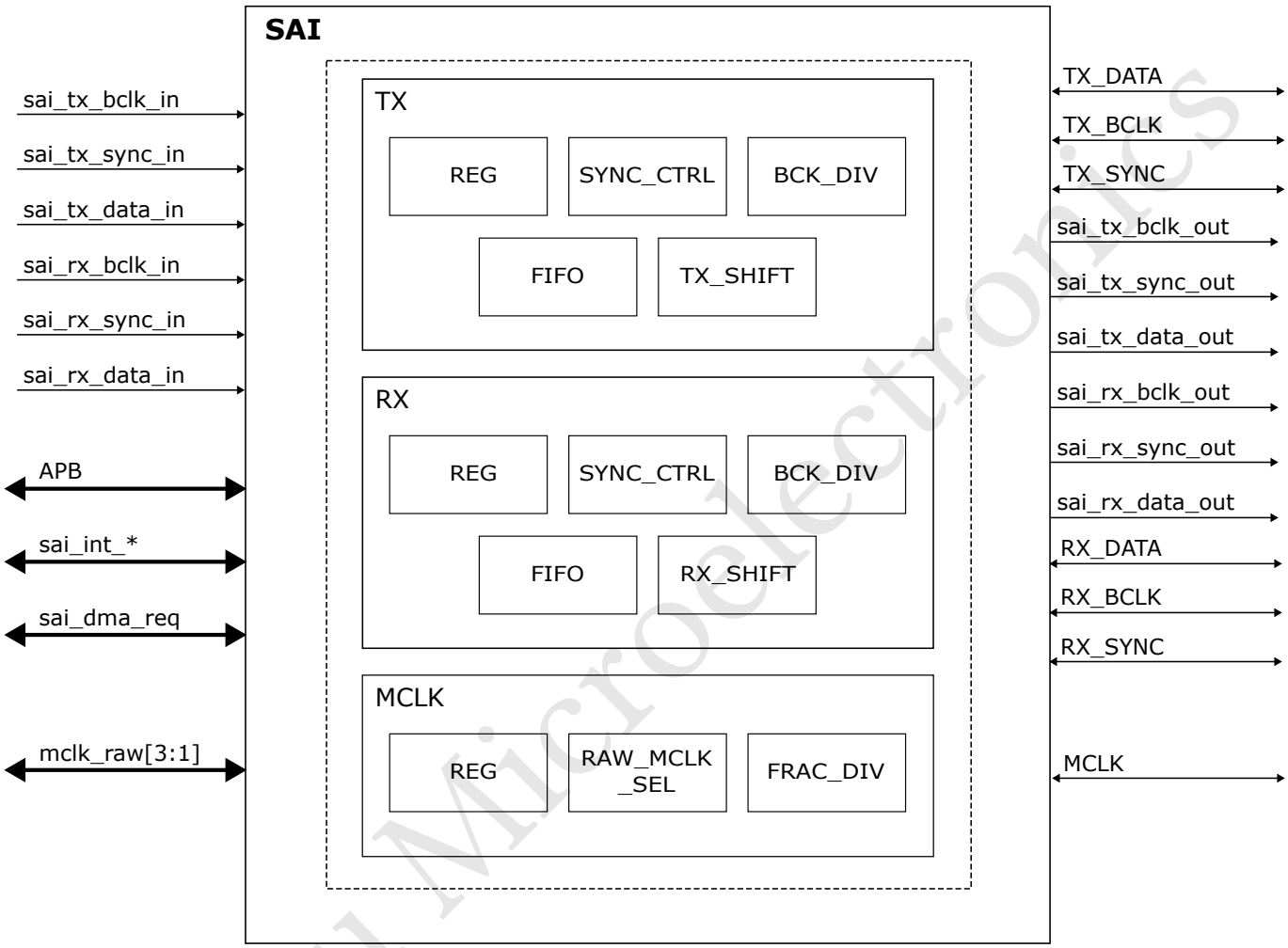


Figure 26.1: Block Diagram of SAI

26.2.4 Signal Description

26.3 Register Definition

26.3.1 SAI Memory Map

SAI0 base address 401cc000h
SAI1 base address 401d0000h

Table 26.6: SAI Memory Map

Offset	Register	Width	Access	Reset Value
0h	Transmit Control Register (TCTRL)	32	RW	00000000h
4h	Transmit Interrupt Enable Register (TINTE)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
8h	Transmit Status Register (TSTS)	32	RW	00000000h
ch	Transmit FIFO Control Register (TFCR)	32	RW	00000000h
10h	Transmit Configuration 0 Register (TCFG0)	32	RW	00000000h
14h	Transmit Configuration 1 Register (TCFG1)	32	RW	00000000h
18h	Transmit Configuration 2 Register (TCFG2)	32	RW	00000000h
1ch	Transmit Configuration 3 Register (TCFG3)	32	RW	00000000h
20h	Transmit Data Register (TDATA0)	32	RW	00000000h
24h	Transmit Data Register (TDATA1)	32	RW	00000000h
28h	Transmit Data Register (TDATA2)	32	RW	00000000h
2ch	Transmit Data Register (TDATA3)	32	RW	00000000h
40h	Transmit FIFO Register (TFR0)	32	RW	00000000h
44h	Transmit FIFO Register (TFR1)	32	RW	00000000h
48h	Transmit FIFO Register (TFR2)	32	RW	00000000h
4ch	Transmit FIFO Register (TFR3)	32	RW	00000000h
60h	Transmit Word Mask (TMR)	32	RW	00000000h
80h	Receive Control Register (RCTRL)	32	RW	00000000h
84h	Receive Interrupt Enable Register (RINTE)	32	RW	00000000h
88h	Receive Status Register (RSTS)	32	RW	00000000h
8ch	Receive FIFO Control Register (RFCR)	32	RW	00000000h
90h	Receive Configuration 0 Register (RCFG0)	32	RW	00000000h
94h	Receive Configuration 1 Register (RCFG1)	32	RW	00000000h
98h	Receive Configuration 2 Register (RCFG2)	32	RW	00000000h
9ch	Receive Configuration 3 Register (RCFG3)	32	RW	00000000h
a0h	Receive Data Register (RDATA0)	32	RW	00000000h
a4h	Receive Data Register (RDATA1)	32	RW	00000000h
a8h	Receive Data Register (RDATA2)	32	RW	00000000h
ach	Receive Data Register (RDATA3)	32	RW	00000000h
c0h	Receive FIFO Register (RFR0)	32	RW	00000000h
c4h	Receive FIFO Register (RFR1)	32	RW	00000000h
c8h	Receive FIFO Register (RFR2)	32	RW	00000000h
cch	Receive FIFO Register (RFR3)	32	RW	00000000h
e0h	Receive Word Mask (RMR)	32	RW	00000000h
100h	MCLK Control Register (MCLK_CTRL)	32	RW	00000000h
104h	MCLK Fractional Divider Register (MCLK_FRAC_DIV)	32	RW	00000000h

Yuntu Microelectronics

26.3.1.1 SAI TCTRL Register

Offset: 0h

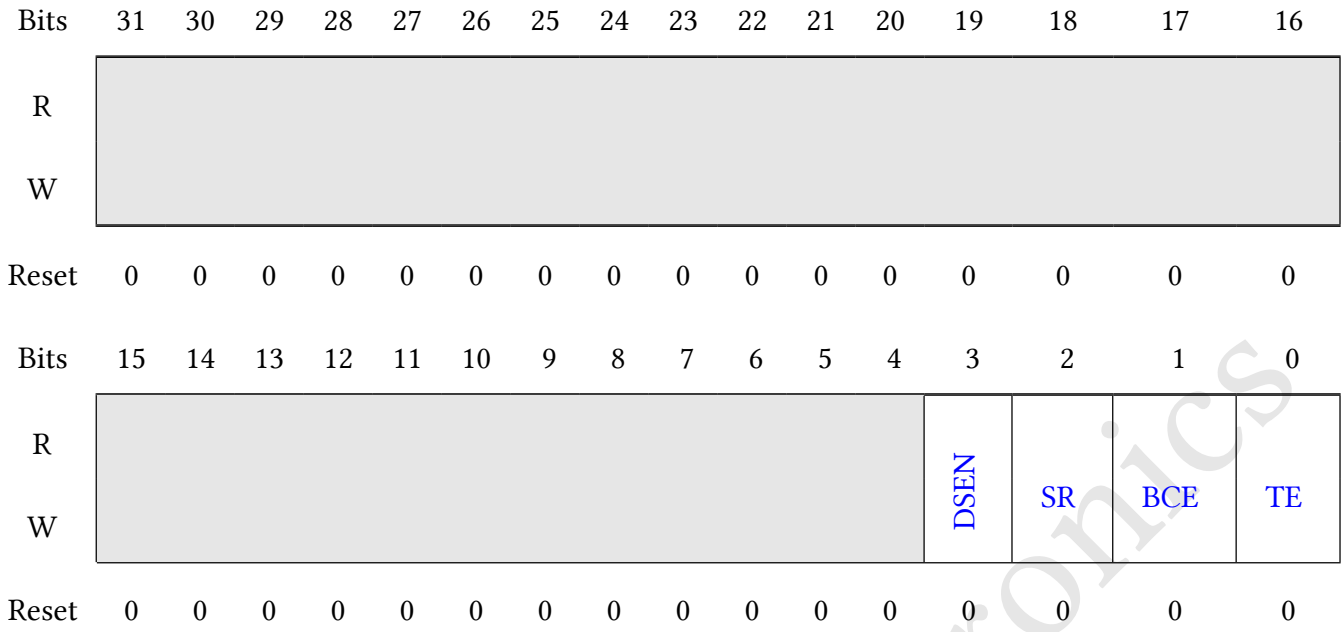


Table 26.7: SAI TCTRL Register Description

Field	Function
3 DSEN	**Deepsleep Enable** Configure the operation state of the transmitter in Deepsleep. 0b - Transmitter disable in Deepsleep 1b - Transmitter enable in Deepsleep
2 SR	**Software Reset** When set, it will reset FIFO and the internal transmitter logic. 0b - No effect. 1b - Software reset.
1 BCE	**Bit Clock Enable** Enables the transmit bit clock, separately from the TE. 0b - Transmit bit clock is disabled when TE is cleared. 1b - Transmit bit clock is enabled.
0 TE	**Transmit Enable** Enable bit for the transmitter. After this bit is write 0, it remains set until the end of the current frame. 0b - Transmitter is disabled. 1b - Transmitter is enabled.

Yuntu Microelectronics

26.3.1.2 SAI TINTE Register

Offset: 4h

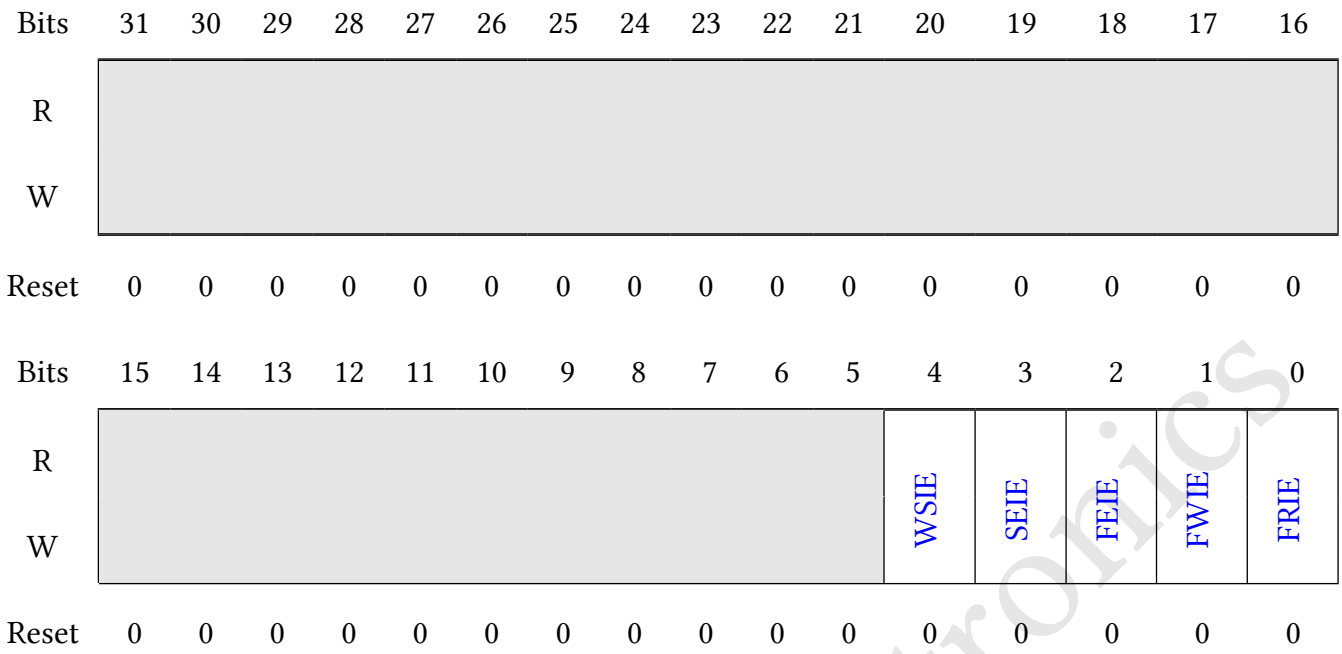


Table 26.8: SAI TINTE Register Description

Field	Function
4 WSIE	**Word Start Interrupt Enable** Enable bit for word start interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
3 SEIE	**Sync Error Interrupt Enable** Enable bit for sync error interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
2 FEIE	**FIFO Error Interrupt Enable** Enable bit for FIFO error interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
1 FWIE	**FIFO Warning Interrupt Enable** Enable bit for FIFO warning interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
0 FRIE	**FIFO request Interrupt Enable** Enable bit for FIFO request interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.

Yuntu Microelectronics

26.3.1.3 SAI TSTS Register

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R												WSF	SEF	FEF	FWF	FRF
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 26.9: SAI TSTS Register Description

Field	Function
4 WSF	**Word Start Flag** Indicates the start of the word.
3 SEF	**Sync Error Flag** Indicates an error is detected on the frame sync which generated externally.
2 FEF	**FIFO Error Flag** Indicates an FIFO underrun has been detected on enabled FIFOs.
1 FWF	**FIFO Warning Flag** Indicates at least one of the enabled FIFOs is empty.
0 FRF	**FIFO request Flag** Indicates at least one of the enabled FIFOs has words less or equal to the FIFO watermark.

26.3.1.4 SAI TFCR Register

Offset: ch

Yuntu Microelectronics

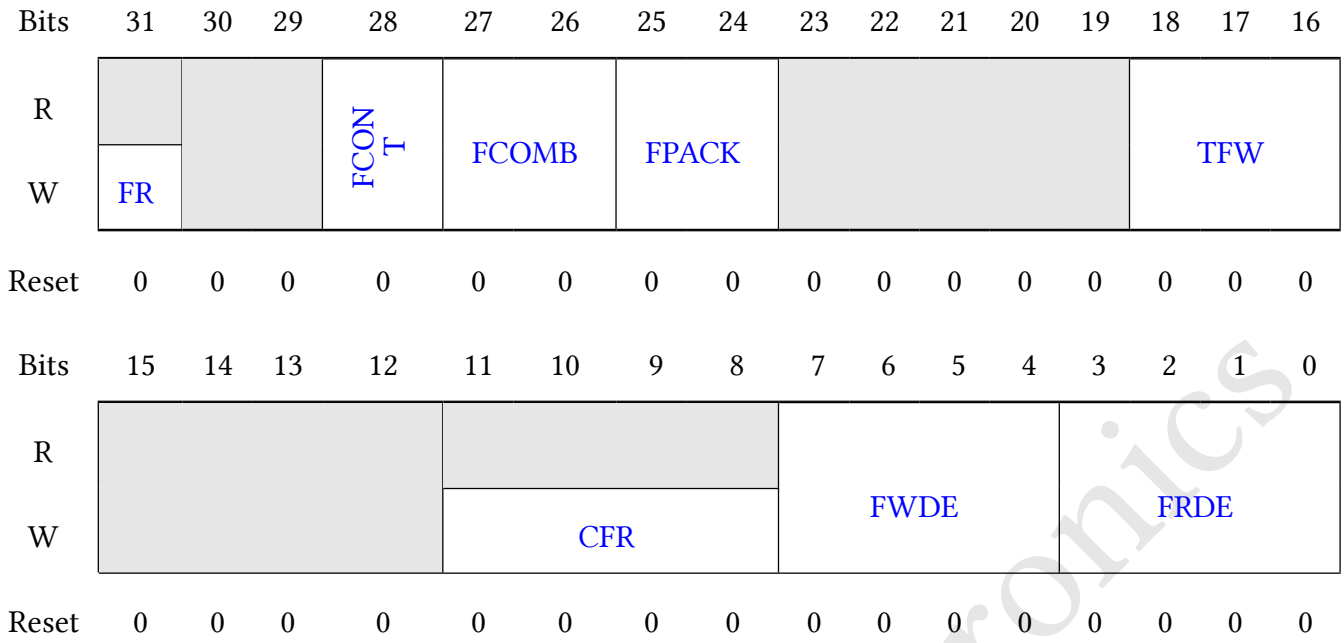


Table 26.10: SAI TFCR Register Description

Field	Function
31 FR	**FIFO Reset** FIFO software reset. Clears the data in the FIFO and sets the FIFO read pointer and write pointer to the same value. When the transmitter is disabled or the FIFO error occurs, this bit can be set.
28 FCONT	**FIFO Continue After Error** When configured, the SAI will continue transmitting when a FIFO error has been detected. 0b - After FIFO error occurs and the FIFO error flag is cleared, the SAI continues from the start of the next frame. 1b - After FIFO error occurs and the FIFO error flag is cleared, the SAI continues from the same word that caused the FIFO error.
27 - 26	**FIFO Combine Mode**

Yuntu Microelectronics

Field	Function
FCOMB	When FIFO combine mode enabled on FIFO reads, the transmit data will alternate the read between the enabled FIFOs. The first word will be transmitted from the first enabled FIFO and the second word will be transmitted from the second FIFO. The first word of the frame is always transmitted from the first enabled FIFO. When FIFO combine mode enabled on FIFO writes, soft writing to any FIFO data will alternate the write between the enabled FIFOs. The first write will store data in the first enabled FIFO and the second write will store data in the second enabled FIFO. The first channel should be always enabled and the combined FIFOs should be continuous. 00b - FIFO combine mode disabled. 01b - FIFO combine mode enabled on FIFO reads. 10b - FIFO combine mode enabled on FIFO writes. 11b - FIFO combine mode enabled on FIFO reads and FIFO writes.
25 - 24 FPACK	**FIFO Packing Mode** Enables packing of 8-bit data or 16-bit data into 32-bit FIFO word. The first word in each frame always start with a new 32-bit FIFO word. 00b - FIFO packing mode disabled. 01b - Reserved. 10b - 8-bit FIFO packing mode enabled. 11b - 16-bit FIFO packing mode enabled.
18 - 16 TFW	**Transmit FIFO Watermark** Configures the watermark level for the enabled FIFOs
11 - 8 CFR	**Channel FIFO Reset** Resets the FIFO pointers for the specific channel. It should only be set when a channel is disabled or the FIFO error flag is set.
7 - 4 FWDE	**FIFO Warning DMA Enable** Enable the DMA request for the specific channel.
3 - 0 FRDE	**FIFO Request DMA Enable** Enable the DMA request for the specific channel.

Yuntu Microelectronics

26.3.1.5 SAI TCFG0 Register

Offset: 10h

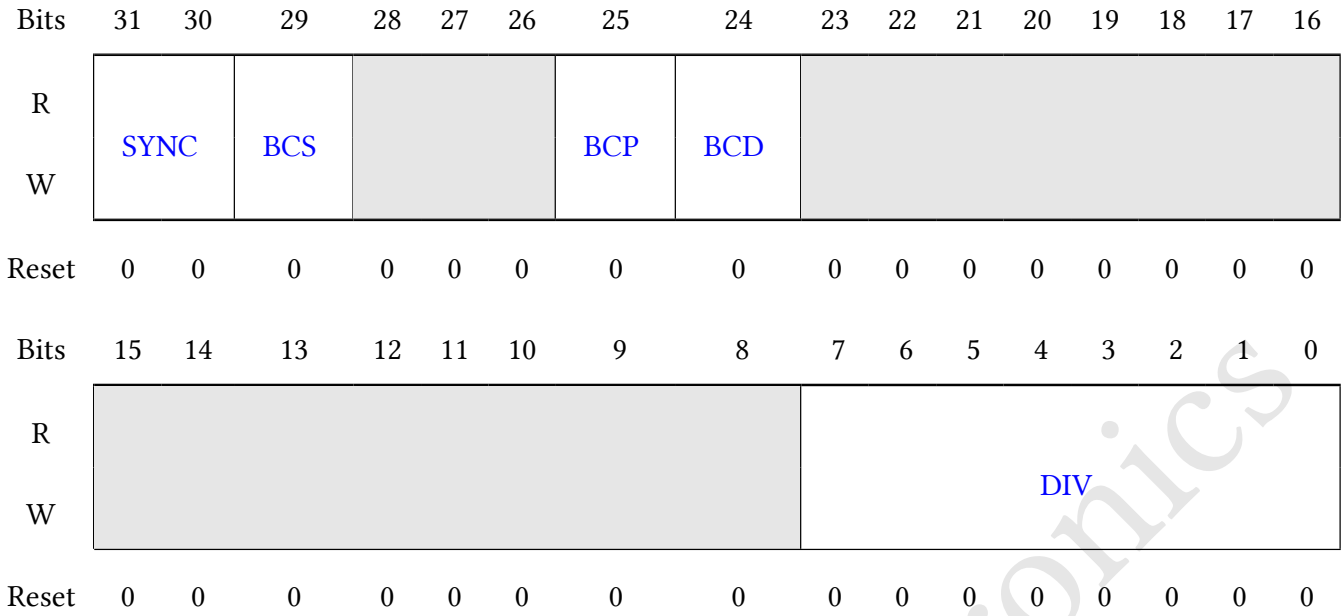


Table 26.11: SAI TCFG0 Register Description

Field	Function
31 - 30 SYNC	**Synchronous Mode** Configures between asynchronous and synchronous modes of operation. In the synchronous mode, it operate with the synchronous bit clock and frame sync. 00b - Asynchronous mode. 01b - Synchronous with the receiver(it should be in asynchronous mode). 10b - Synchronous with another SAI transmitter(it should be in asynchronous mode). 11b - Synchronous with another SAI receiver(it should be in asynchronous mode).
29 BCS	**Bit Clock Swap** In asynchronous mode, swaps the bit clock with the receiver. This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync. In synchronous mode(SYNC=01b), the transmitter and the receiver should be set to the same value. When both set, the transmitter and the receiver both use the transmitter bit clock. 0b - Use the normal bit clock source. 1b - Swap the bit clock source.
25 BCP	**Bit Clock Polarity** Configures the polarity of the bit clock. 0b - Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1b - Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24	**Bit Clock Direction**

Yuntu Microelectronics

Field	Function
BCD	Configures the direction of the bit clock. 0b - Bit clock is generated externally in slave mode. 1b - Bit clock is generated internally in master mode.
7 - 0 DIV	**Bit Clock Divide** Divide the audio master clock to generate the bit clock. The division value is (DIV+1)*2

26.3.1.6 SAI TCFG1 Register

Offset: 14h

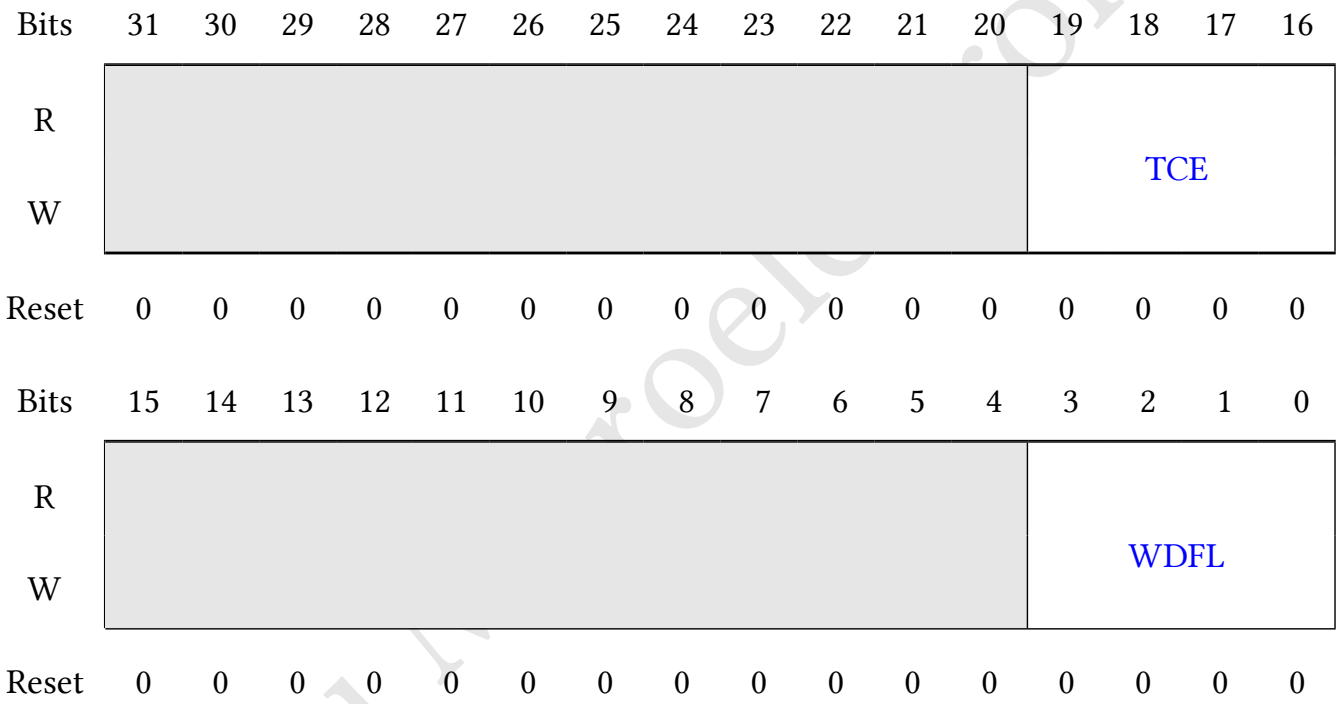


Table 26.12: SAI TCFG1 Register Description

Field	Function
19 - 16 TCE	**Transmit Channel Enable** Enable the corresponding data channels for transmit. Changing this field take effect immediately at the end of the frame.
3 - 0 WDFL	**Configures which word sets the start of word flag. The value of this field must be one less than the word number.**

Yuntu Microelectronics

26.3.1.7 SAI TCFG2 Register

Offset: 18h

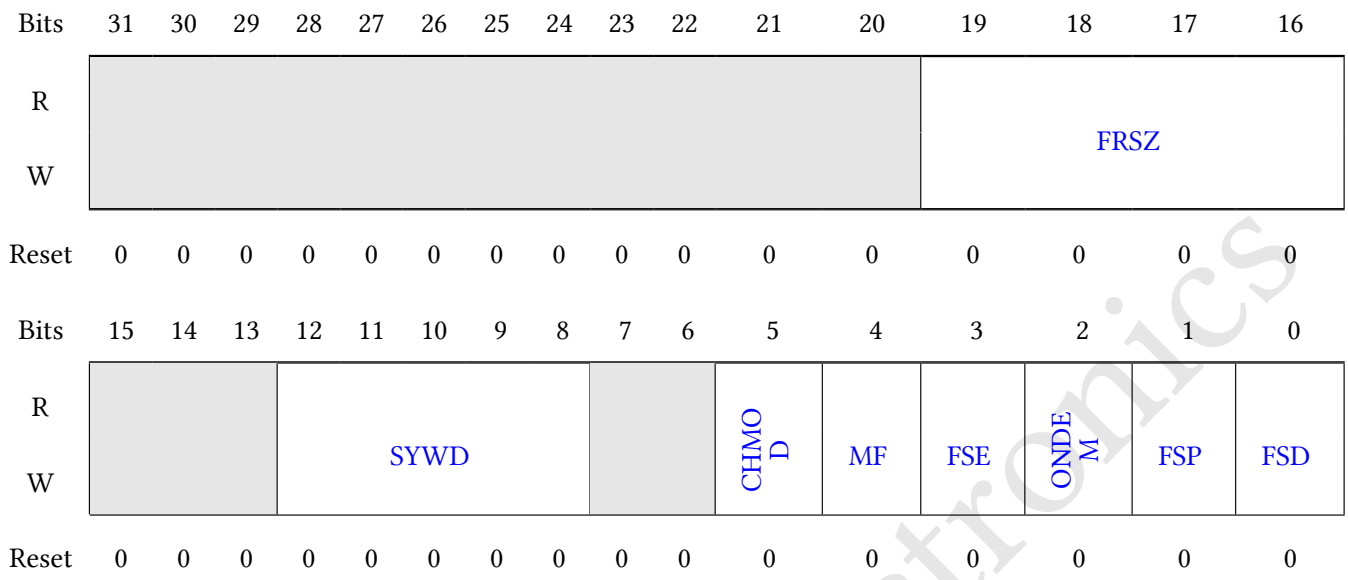


Table 26.13: SAI TCFG2 Register Description

Field	Function
19 - 16 FRSZ	**Frame Size** Configures the number of words in each frame. The number of words in the frame is (FRSZ+1).
12 - 8 SYWD	**Sync Width** Configures the length of the frame sync in number of bit clock. The length is (SYWD+1) bit clock cycles.
5 CHMOD	**Channel Mode** Configures if the transmit data pins are TDM mode or output mode. 0b - TDM mode. The transmit data pins are tri-stated when slots are masked or channels are disabled. 1b - Output mode. The transmit data pins output zero when slots are masked or channels are disabled.
4 MF	**MSF First** Configures whether the LSB or the MSB is transmitted first. 0b - LSB first. 1b - MSB first.
3 FSE	**Frame Sync Early** 0b - Frame sync asserts with the first bit of the frame. 1b - Frame sync asserts one bit before the first bit of the frame.
2	**On Demand Mode**

Yuntu Microelectronics

Field	Function
ONDEM	Configures that the frame sync is only generated when the FIFO warning flag is cleared and the frame sync is generated internally. 0b - Internal frame sync is generated continuously. 1b - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	**Frame Sync Polarity** Configures the polarity of the frame sync. 0b - Frame sync is active high. 1b - Frame sync is active low.
0 FSD	**Frame Sync Direction** Configures the direction of the frame sync. 0b - Frame sync is generated externally in slave mode. 1b - Frame sync is generated internally in master mode.

26.3.1.8 SAI TCFG3 Register

Offset: 1ch

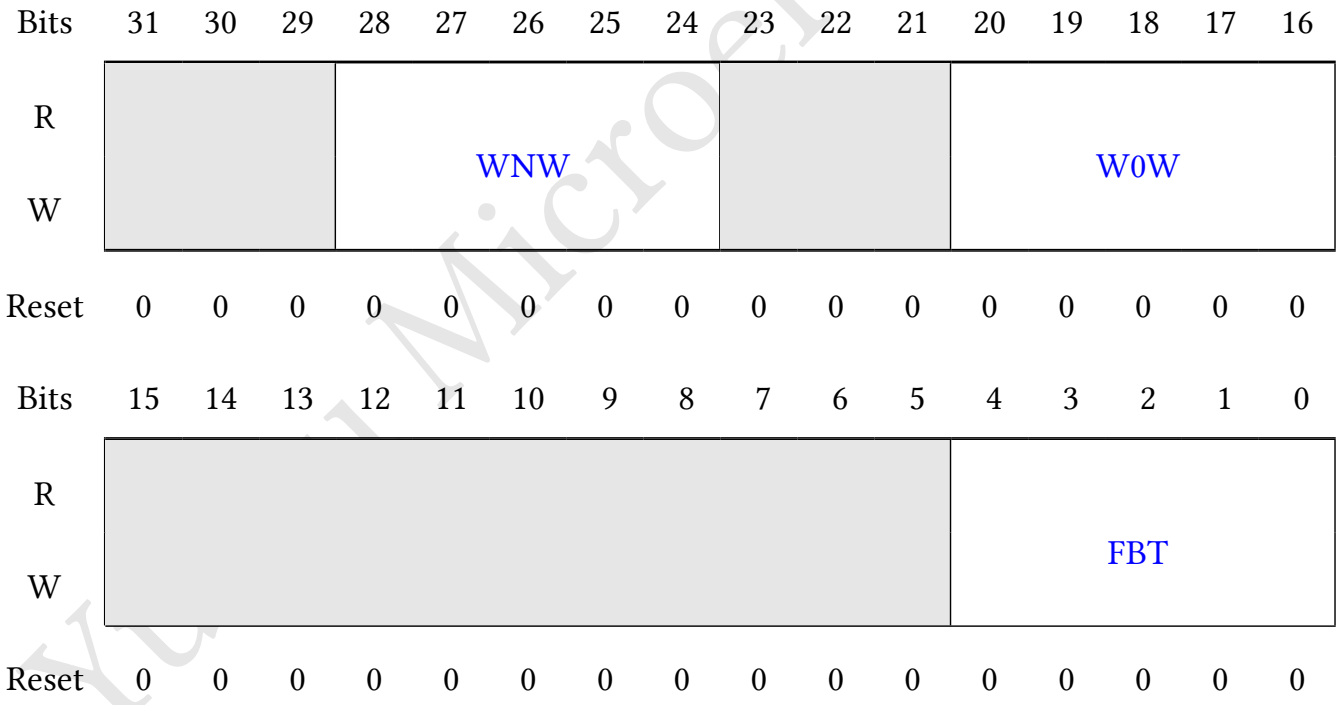


Table 26.14: SAI TCFG3 Register Description

Field	Function
28 - 24 WNW	**Word N Width** Configure the number of bits in each word except the first word in the frame. The number of bits in each word is (WNW+1) and should be no less than 8.

Yuntu Microelectronics

Field	Function
20 - 16 W0W	**Word 0 Width** Configure the number of bits in the first word in each frame. The number of bits in the first word is (W0W+1) and should be no less than 8.
4 - 0 FBT	**First Bit Shifted** Configures the bit index for the first bit transmitted for each word.

26.3.1.9 SAI TDATA Register

Offset: 20h + n * 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	TDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	TDATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 26.15: SAI TDATA Register Description

Field	Function
31 - 0 TDATA	**Transmit Data Register** Writes to this field will push the data into the FIFO when the FIFO is not full.

26.3.1.10 SAI TFR Register

Offset: 40h + n * 4h

Yuntu Microelectronics

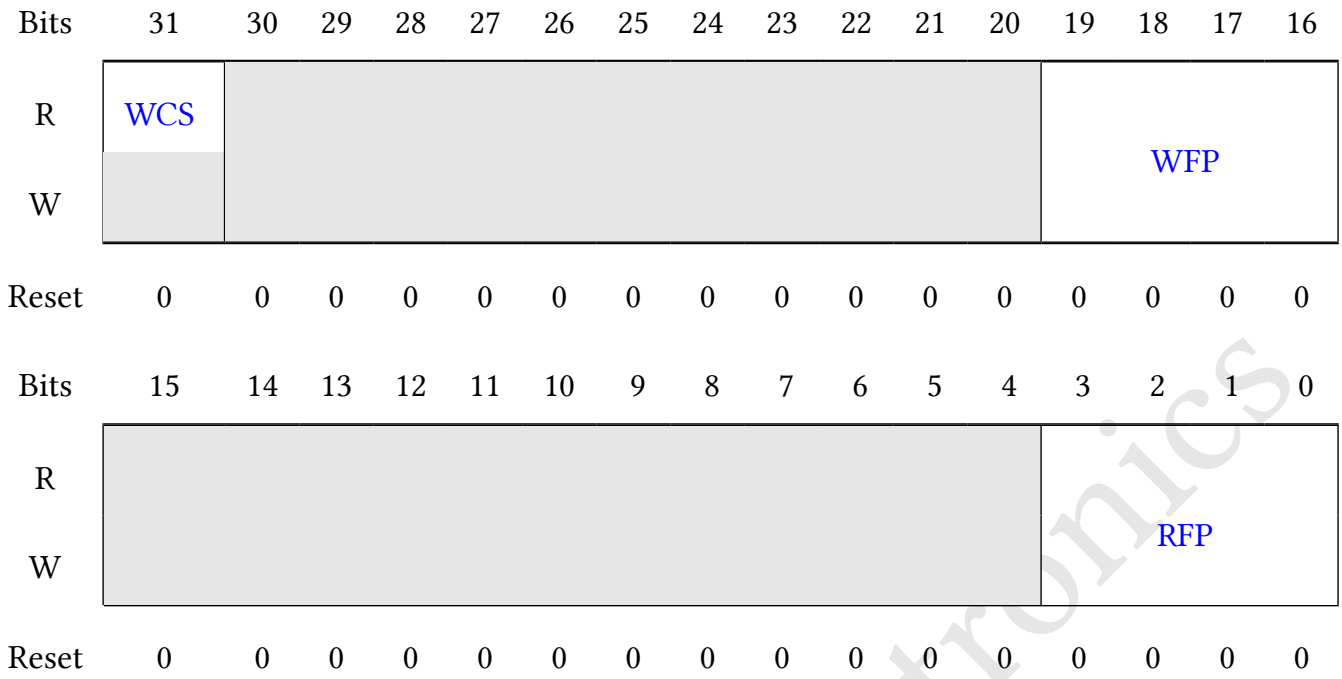


Table 26.16: SAI TFR Register Description

Field	Function
31 WCS	**Write Channel Select** Indicates FIFO of this channel is to be write next during FIFO combine mode.
19 - 16 WFP	**Write FIFO Pointer**
3 - 0 RFP	**Read FIFO Pointer**

Yuntu Microelectronics

26.3.1.11 SAI TMR Register

Offset: 60h

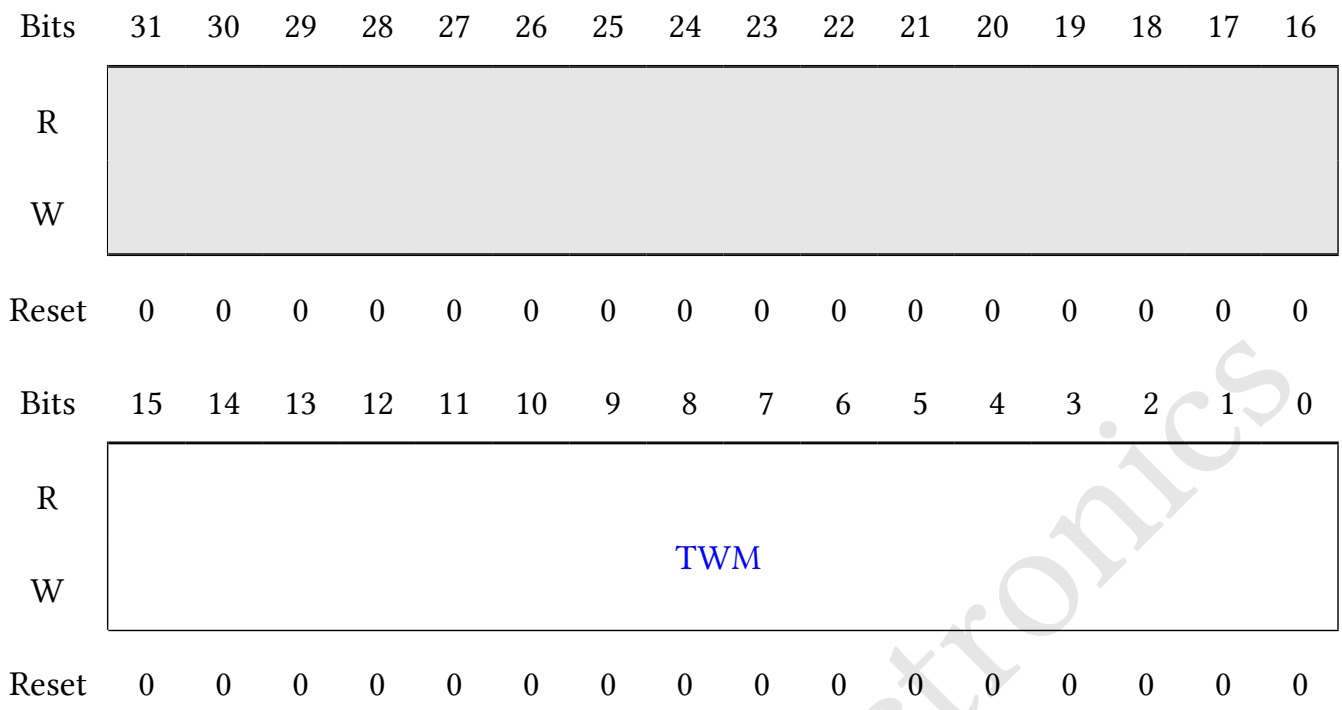


Table 26.17: SAI TMR Register Description

Field	Function
15 - 0 TWM	**Transmit Word Mask** Configure whether the transmit word is masked for the corresponding word in the frame. This register is double-buffered and updates when TCTRL[TE] is set from 0 to 1 or at the end of each frame.

Yuntu Microelectronics

26.3.1.12 SAI RCTRL Register

Offset: 80h

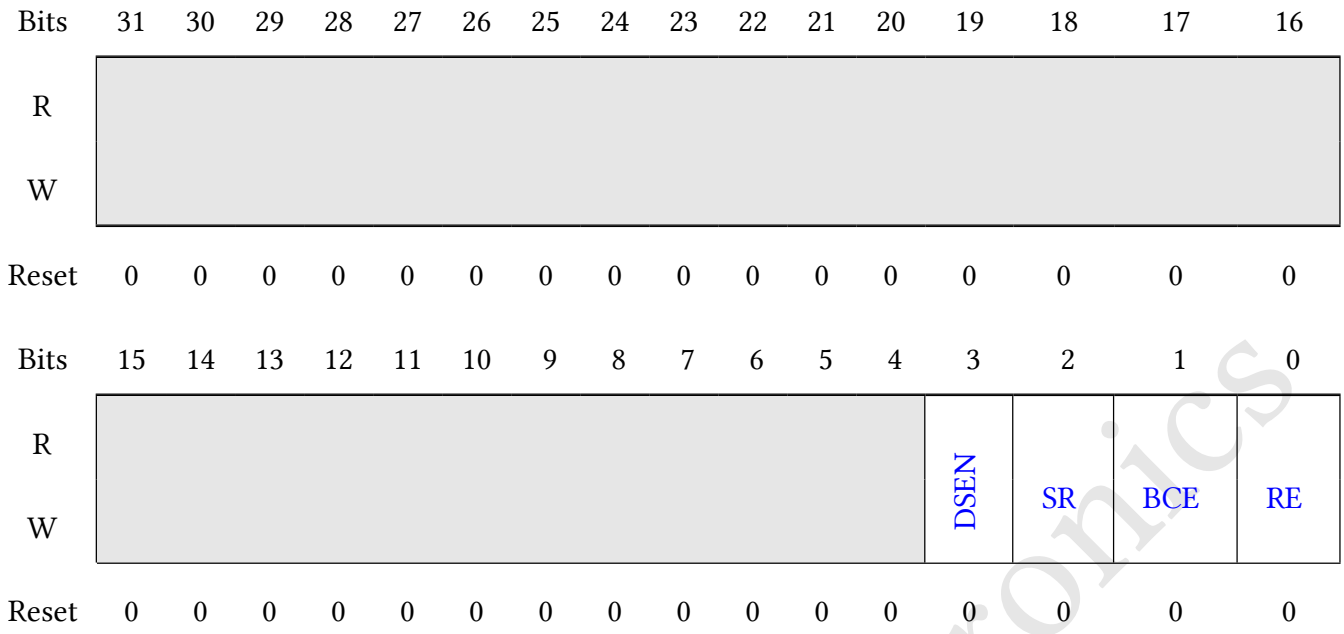


Table 26.18: SAI RCTRL Register Description

Field	Function
3 DSEN	**Deepsleep Enable** Configure the operation state of the receiver in Deepsleep. 0b - Receiver disable in Deepsleep 1b - Receiver enable in Deepsleep
2 SR	**Software Reset** When set, it will reset FIFO and the internal receiver logic. 0b - No effect. 1b - Software reset.
1 BCE	**Bit Clock Enable** Enables the receive bit clock, separately from the RE. 0b - Receive bit clock is disabled when RE is cleared. 1b - Receive bit clock is enabled.
0 RE	**Receive Enable** Enable bit for the receiver. After this bit is write 0, it remains set until the end of the current frame. 0b - Receiver is disabled. 1b - Receiver is enabled.

Yuntu Microelectronics

26.3.1.13 SAI RINTE Register

Offset: 84h

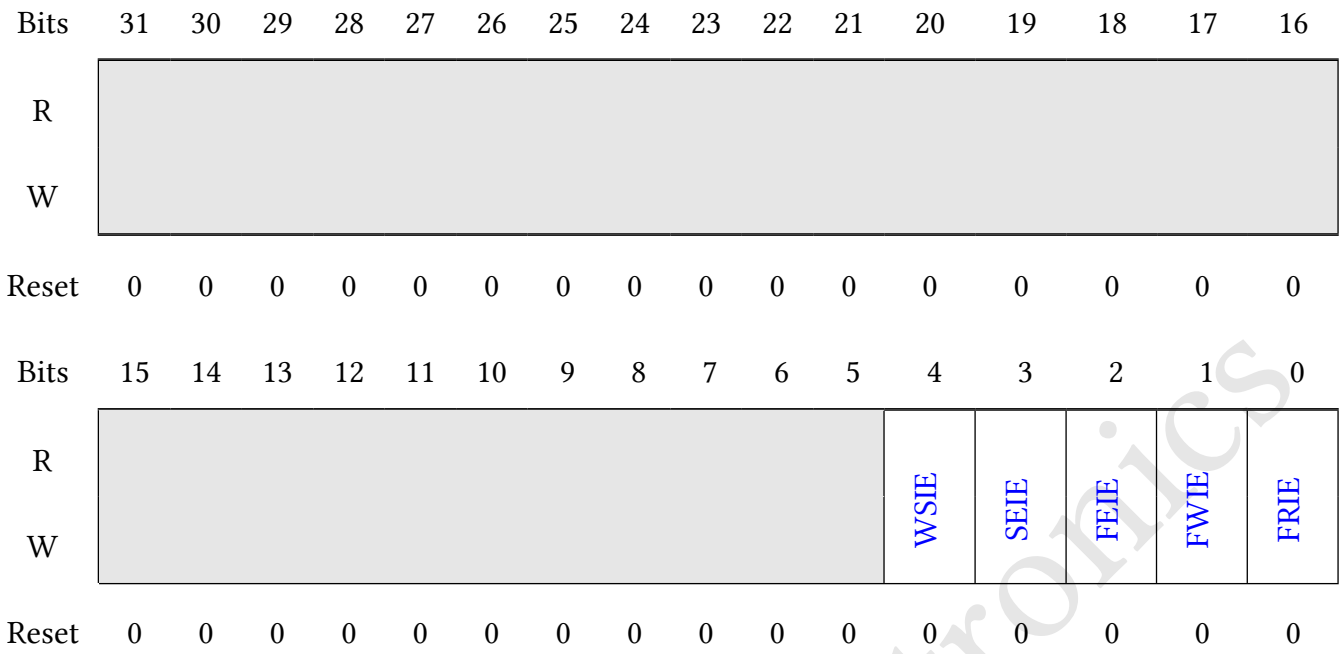


Table 26.19: SAI RINTE Register Description

Field	Function
4 WSIE	**Word Start Interrupt Enable** Enable bit for word start interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
3 SEIE	**Sync Error Interrupt Enable** Enable bit for sync error interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
2 FEIE	**FIFO Error Interrupt Enable** Enable bit for FIFO error interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
1 FWIE	**FIFO Warning Interrupt Enable** Enable bit for FIFO warning interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.
0 FRIE	**FIFO request Interrupt Enable** Enable bit for FIFO request interrupt. 0b - Disable the interrupt. 1b - Enable the interrupt.

Yuntu Microelectronics

26.3.1.14 SAI RSTS Register

Offset: 88h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R												WSF	SEF	FEF	FWF	FRF
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 26.20: SAI RSTS Register Description

Field	Function
4 WSF	**Word Start Flag** Indicates the start of the word.
3 SEF	**Sync Error Flag** Indicates an error is detected on the frame sync which generated externally.
2 FEF	**FIFO Error Flag** Indicates an FIFO overrun has been detected on enabled FIFOs.
1 FWF	**FIFO Warning Flag** Indicates at least one of the enabled FIFOs is full.
0 FRF	**FIFO request Flag** Indicates at least one of the enabled FIFOs has words greater or than the FIFO watermark.

26.3.1.15 SAI RFCR Register

Offset: 8ch

Yuntu Microelectronics

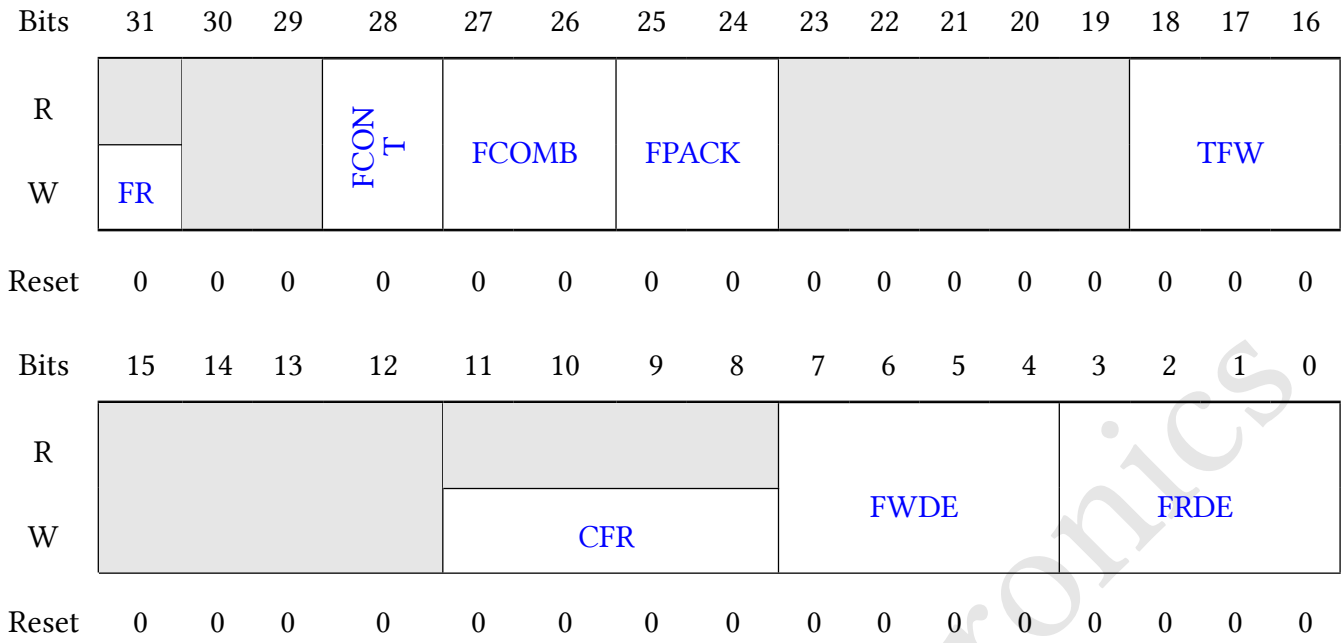


Table 26.21: SAI RFCR Register Description

Field	Function
31 FR	**FIFO Reset** FIFO software reset. Clears the data in the FIFO and sets the FIFO read pointer and write pointer to the same value. When the receiver is disabled or the FIFO error occurs, this bit can be set.
28 FCONT	**FIFO Continue After Error** When configured, the SAI will continue transmitting when a FIFO error has been detected. 0b - After FIFO error occurs and the FIFO error flag is cleared, the SAI continues from the start of the next frame. 1b - After FIFO error occurs and the FIFO error flag is cleared, the SAI continues from the same word that caused the FIFO error.
27 - 26	**FIFO Combine Mode**

Yuntu Microelectronics

Field	Function
FCOMB	When FIFO combine mode enabled on FIFO write, the receive data will alternate the write between the enabled FIFOs. The first word will be stored to the first enabled FIFO and the second word will be stored to the second FIFO. The first word of the frame is always stored to the first enabled FIFO. When FIFO combine mode enabled on FIFO reads, soft reading to any FIFO data will alternate the read between the enabled FIFOs. The first read will get data from the first enabled FIFO and the second read will get data from the second enabled FIFO. The first channel should be always enabled and the combined FIFOs should be continuous. 00b - FIFO combine mode disabled. 01b - FIFO combine mode enabled on FIFO writes. 10b - FIFO combine mode enabled on FIFO reads. 11b - FIFO combine mode enabled on FIFO reads and FIFO writes.
25 - 24 FPACK	**FIFO Packing Mode** Enables packing of 8-bit data or 16-bit data into 32-bit FIFO word. The first word in each frame always start with a new 32-bit FIFO word. 00b - FIFO packing mode disabled. 01b - Reserved. 01b - 8-bit FIFO packing mode enabled. 01b - 16-bit FIFO packing mode enabled.
18 - 16 TFW	**Receive FIFO Watermark** Configures the watermark level for the enabled FIFOs
11 - 8 CFR	**Channel FIFO Reset** Resets the FIFO pointers for the specific channel. It should only be set when a channel is disabled or the FIFO error flag is set.
7 - 4 FWDE	**FIFO Warning DMA Enable** Enable the DMA request for the specific channel.
3 - 0 FRDE	**FIFO Request DMA Enable** Enable the DMA request for the specific channel.

Yuntu Microelectronics

26.3.1.16 SAI RCFG0 Register

Offset: 90h

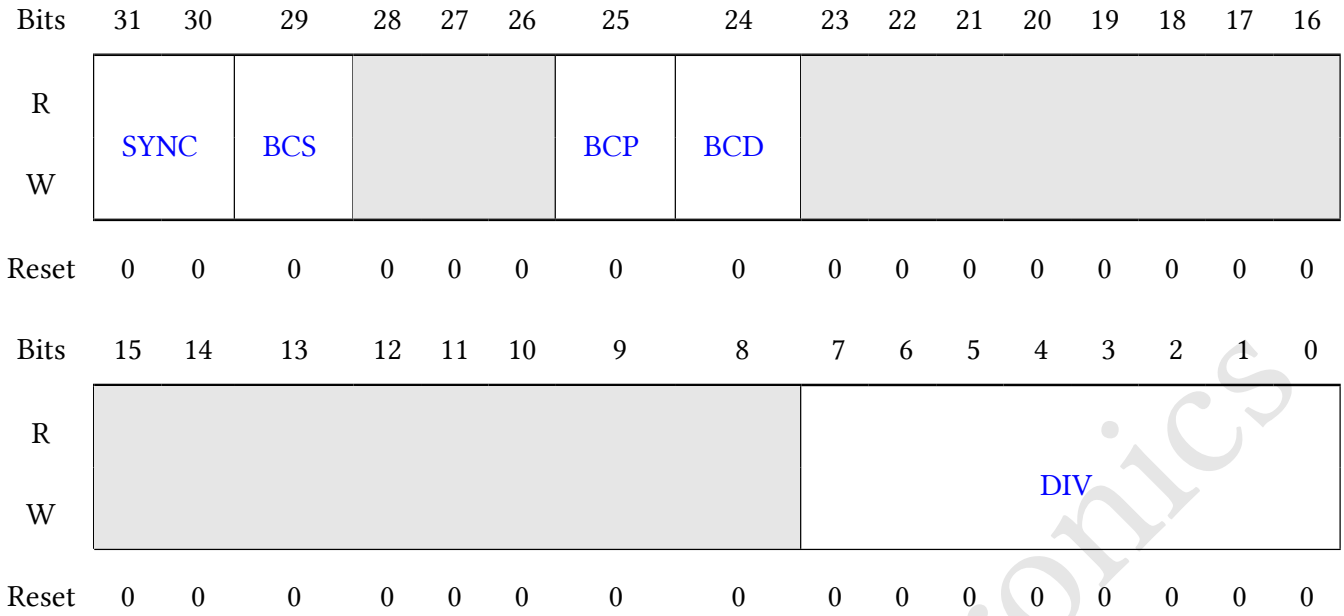


Table 26.22: SAI RCFG0 Register Description

Field	Function
31 - 30 SYNC	**Synchronous Mode** Configures between asynchronous and synchronous modes of operation. In the synchronous mode, it operate with the synchronous bit clock and frame sync. 00b - Asynchronous mode. 01b - Synchronous with the transmitter(it should be in asynchronous mode). 10b - Synchronous with another SAI receiver(it should be in asynchronous mode). 11b - Synchronous with another SAI transmitter(it should be in asynchronous mode).
29 BCS	**Bit Clock Swap** In asynchronous mode, swaps the bit clock with the transmitter. his allows the transmitter and receiver to share the same bit clock, but the receiver continues to use the receive frame sync. In synchronous mode(SYNC=01b), the transmitter and the receiver should be set to the same value. When both set, the transmitter and the receiver both use the receiver bit clock. 0b - Use the normal bit clock source. 1b - Swap the bit clock source.
25 BCP	**Bit Clock Polarity** Configures the polarity of the bit clock. 0b - Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1b - Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24	**Bit Clock Direction**

Yuntu Microelectronics

Field	Function
BCD	Configures the direction of the bit clock. 0b - Bit clock is generated externally in slave mode. 1b - Bit clock is generated internally in master mode.
7 - 0 DIV	**Bit Clock Divide** Divide the audio master clock to generate the bit clock. The division value is (DIV+1)*2

26.3.1.17 SAI RCFG1 Register

Offset: 94h

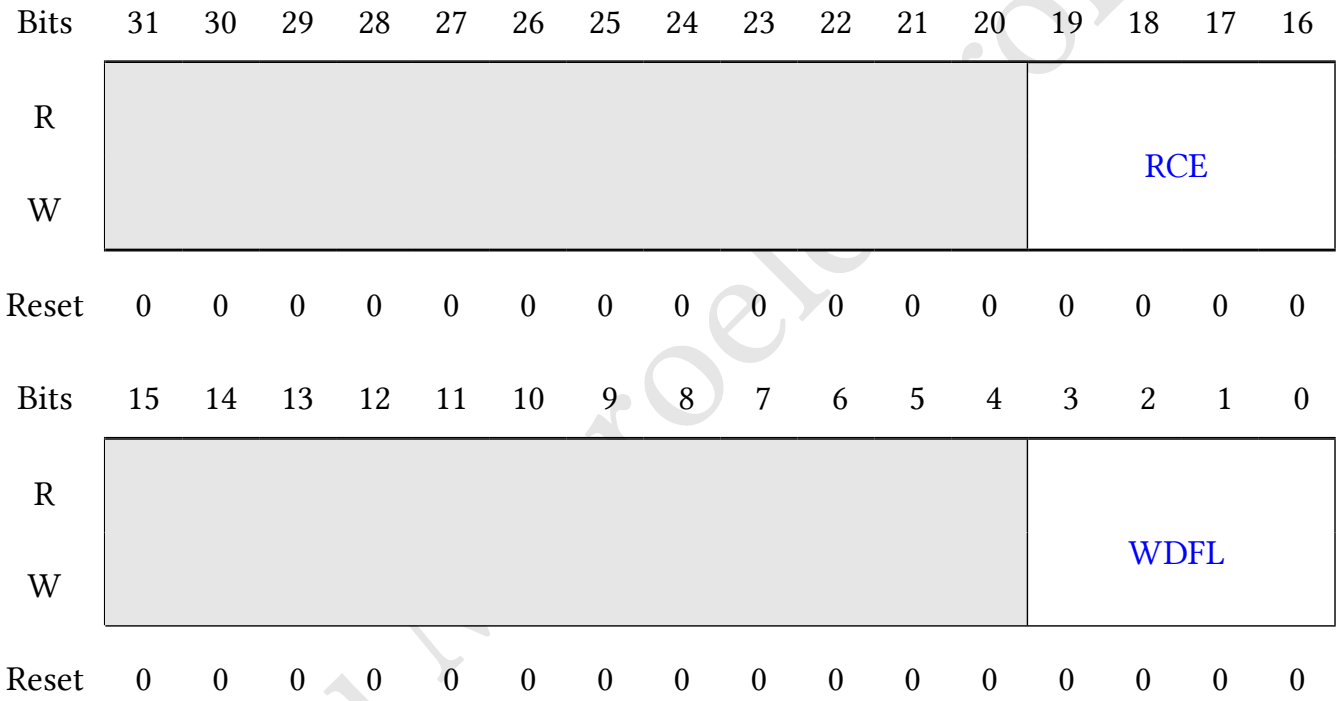


Table 26.23: SAI RCFG1 Register Description

Field	Function
19 - 16 RCE	**Receive Channel Enable** Enable the corresponding data channels for receive. Changing this field take effect immediately at the end of the frame.
3 - 0 WDFL	**Configures which word sets the start of word flag. The value of this field must be one less than the word number.**

Yuntu Microelectronics

26.3.1.18 SAI RCFG2 Register

Offset: 98h

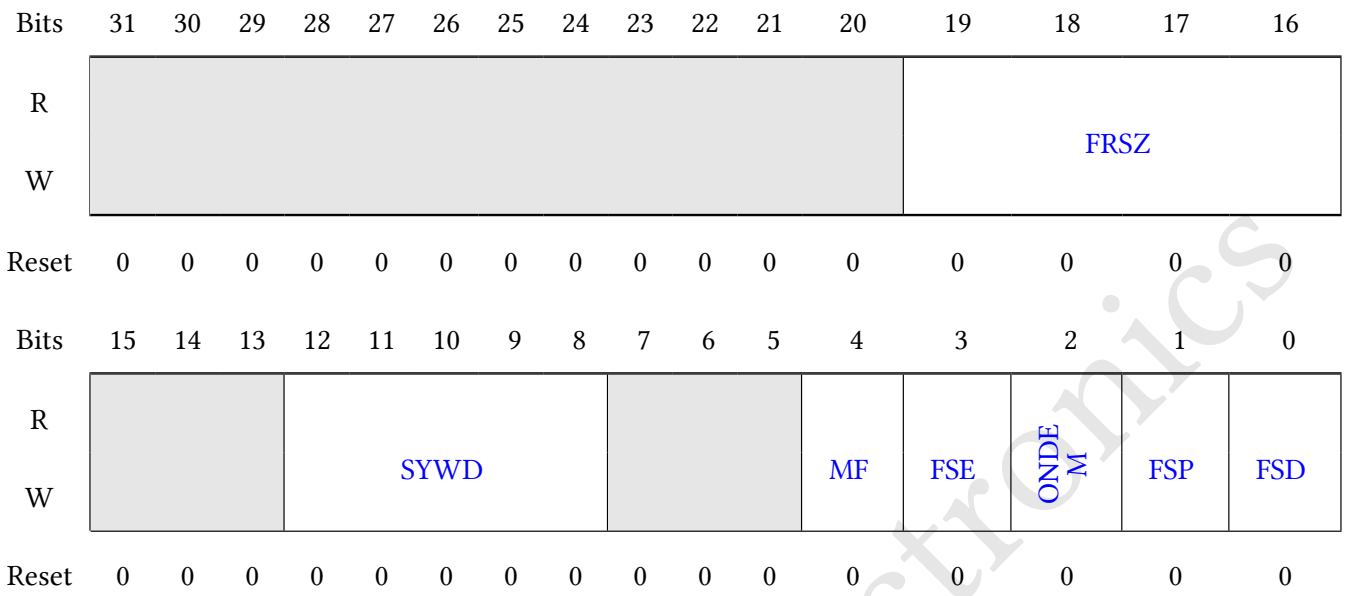


Table 26.24: SAI RCFG2 Register Description

Field	Function
19 - 16 FRSZ	**Frame Size** Configures the number of words in each frame. The number of words in the frame is (FRSZ+1).
12 - 8 SYWD	**Sync Width** Configures the length of the frame sync in number of bit clock. The length is (SYWD+1) bit clock cycles.
4 MF	**MSF First** Configures whether the LSB or the MSB is received first. 0b - LSB first. 1b - MSB first.
3 FSE	**Frame Sync Early** 0b - Frame sync asserts with the first bit of the frame. 1b - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	**On Demand Mode** Configures that the frame sync is only generated when the FIFO warning flag is cleared and the frame sync is generated internally. 0b - Internal frame sync is generated continuously. 1b - Internal frame sync is generated when the FIFO warning flag is clear.
1	**Frame Sync Polarity**

Yuntu Microelectronics

Field	Function
FSP	Configures the polarity of the frame sync. 0b - Frame sync is active high. 1b - Frame sync is active low.
0 FSD	**Frame Sync Direction** Configures the direction of the frame sync. 0b - Frame sync is generated externally in slave mode. 1b - Frame sync is generated internally in master mode.

26.3.1.19 SAI RCFG3 Register

Offset: 9ch

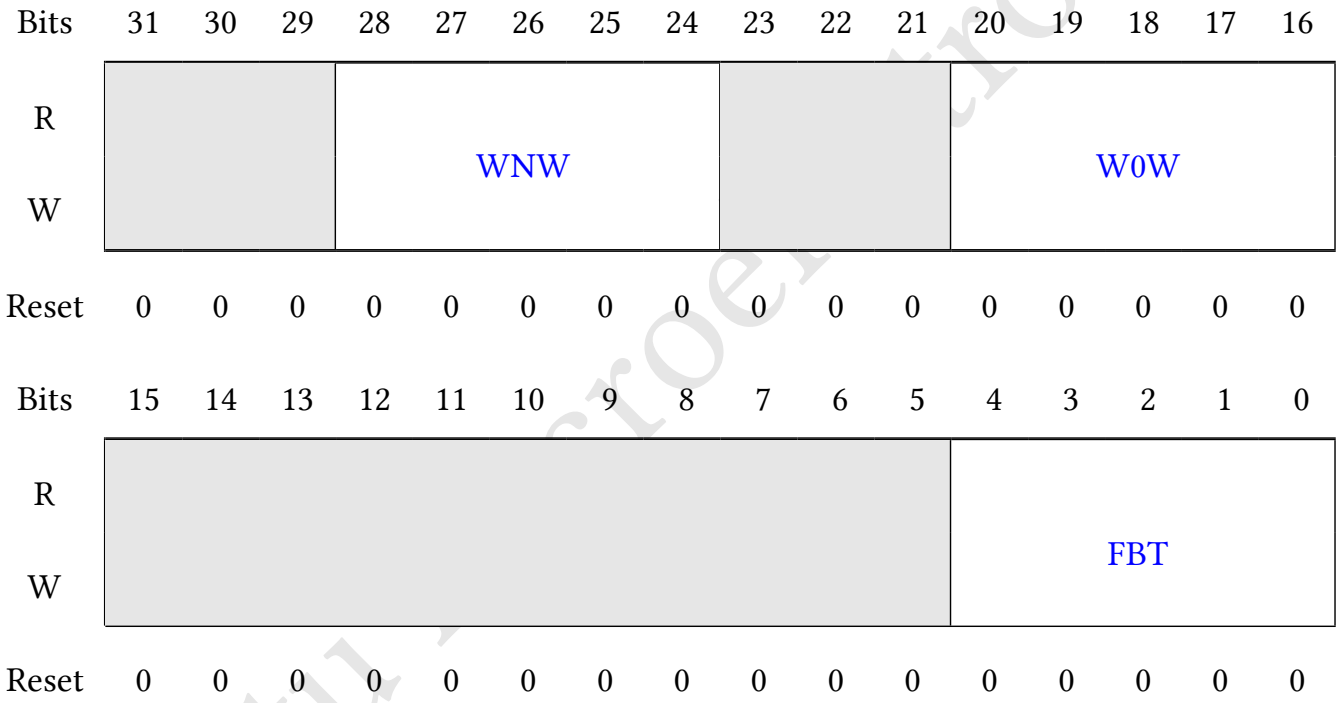


Table 26.25: SAI RCFG3 Register Description

Field	Function
28 - 24 WNW	**Word N Width** Configure the number of bits in each word except the first word in the frame. The number of bits in each word is (WNW+1) and should be no less than 8.
20 - 16 W0W	**Word 0 Width** Configure the number of bits in the first word in each frame. The number of bits in the first word is (W0W+1) and should be no less than 8.
4 - 0 FBT	**First Bit Shifted** Configures the bit index for the first bit received for each word.

Yuntu Microelectronics

26.3.1.20 SAI RDATA Register

Offset: a0h + n * 4h

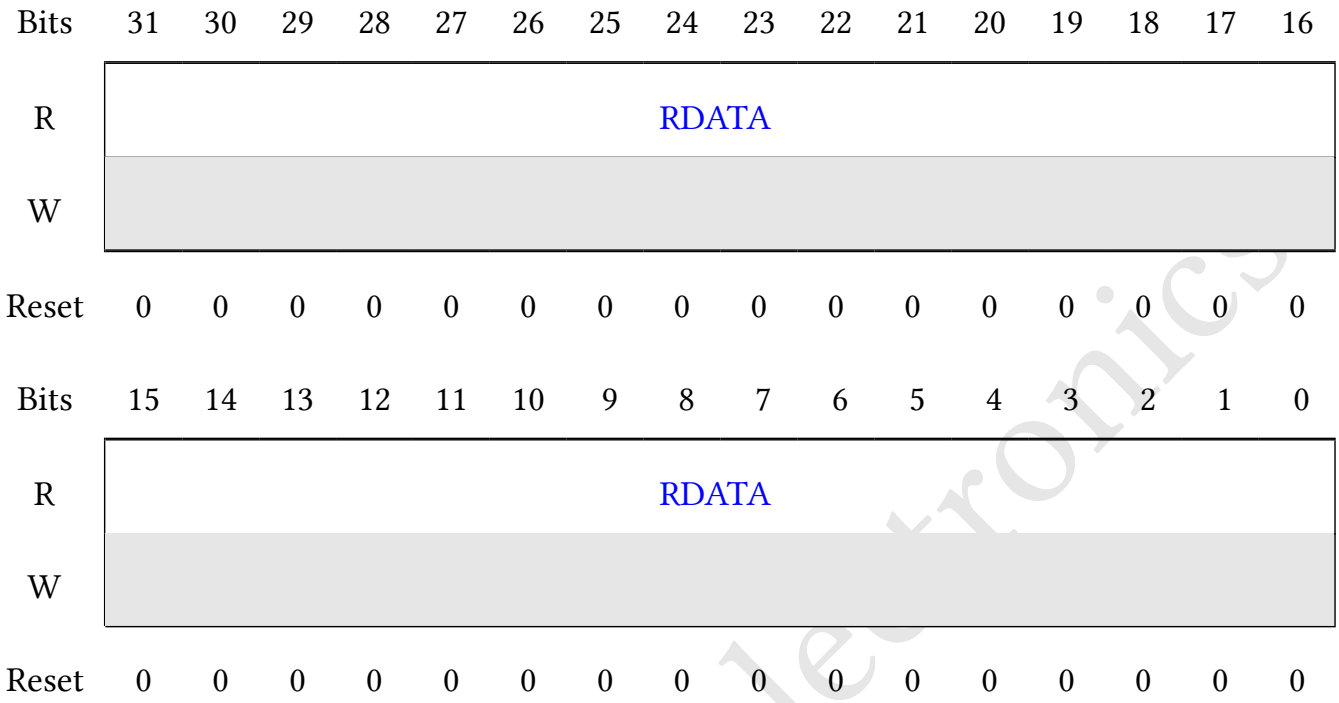


Table 26.26: SAI RDATA Register Description

Field	Function
31 - 0 RDATA	**Receive Data Register** Read this field will return the data from the FIFO when the FIFO is not empty.

Yuntu Microelectronics

26.3.1.21 SAI RFR Register

Offset: c0h + n * 4h

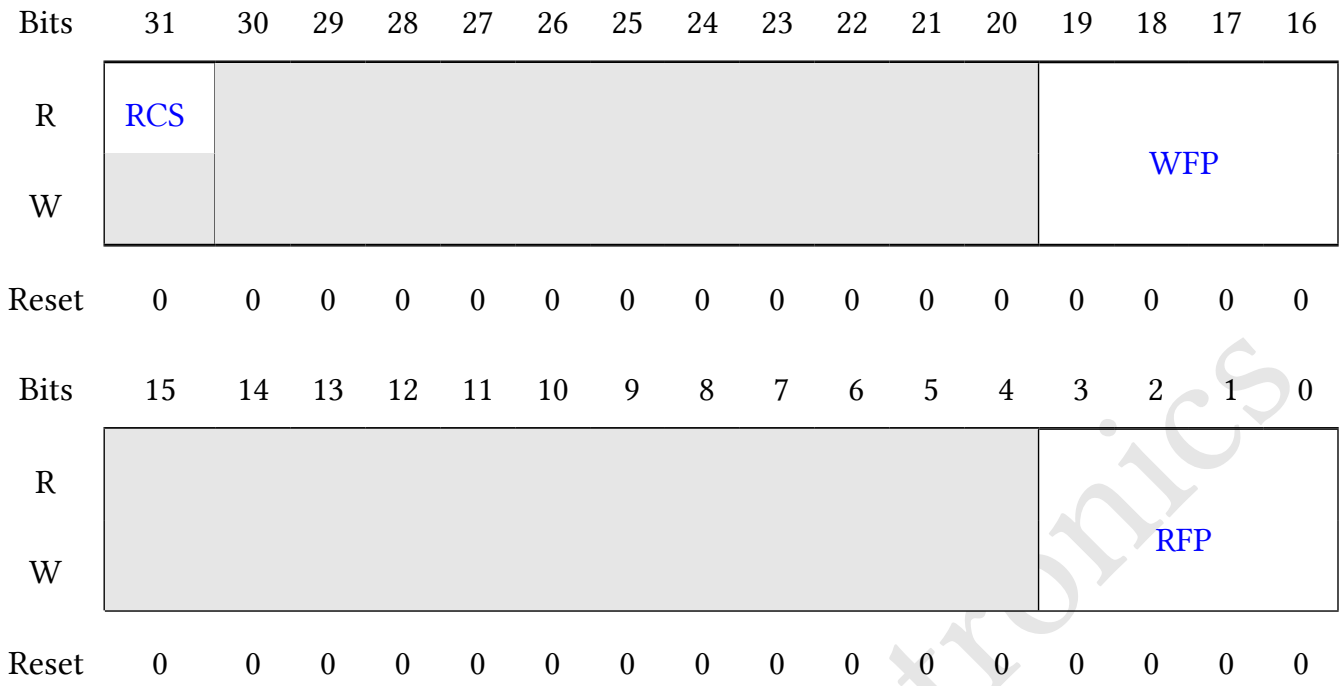


Table 26.27: SAI RFR Register Description

Field	Function
31 RCS	**Read Channel Select** Indicates FIFO of this channel is to be read next during FIFO combine mode.
19 - 16 WFP	**Write FIFO Pointer**
3 - 0 RFP	**Read FIFO Pointer**

Yuntu Microelectronics

26.3.1.22 SAI RMR Register

Offset: e0h

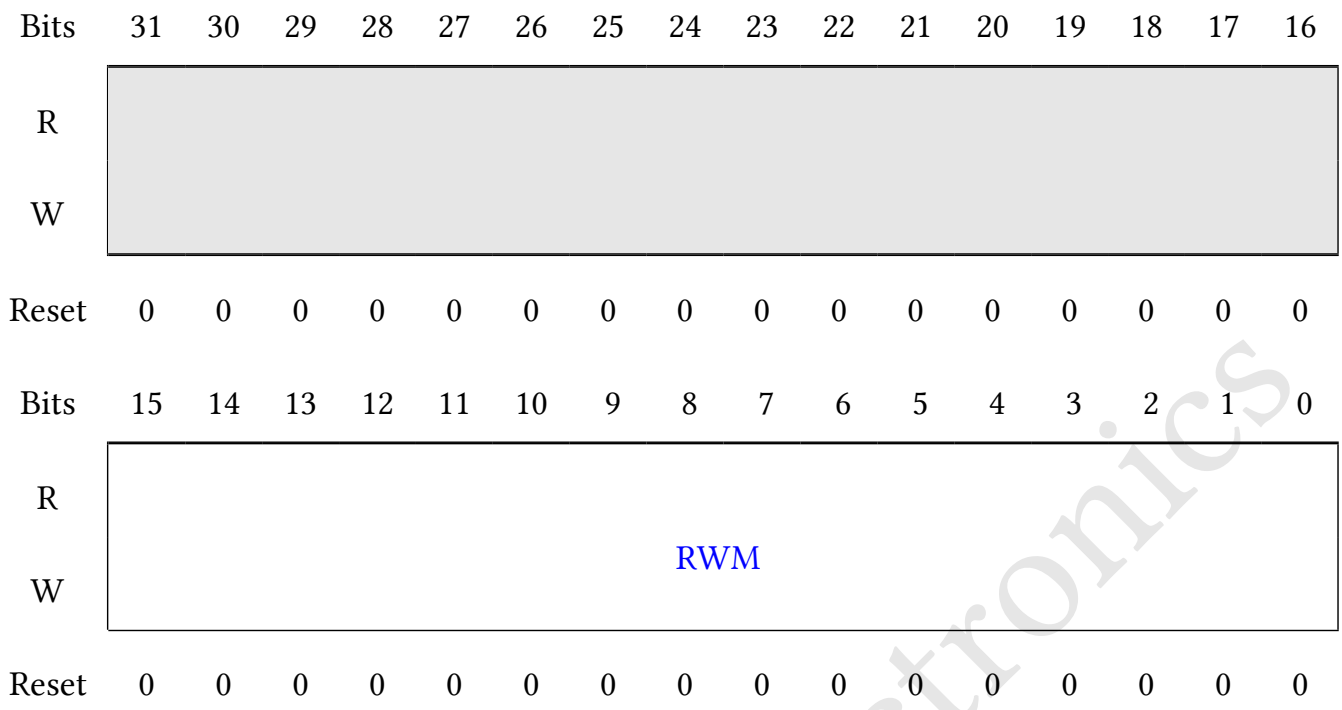


Table 26.28: SAI RMR Register Description

Field	Function
15 - 0 RWM	**Receive Word Mask** Configure whether the receive word is masked for the corresponding word in the frame. This register is double-buffered and updates when RCTRL[RE] is set from 0 to 1 or at the end of each frame.

Yuntu Microelectronics

26.3.1.23 SAI MCLK_CTRL Register

Offset: 100h

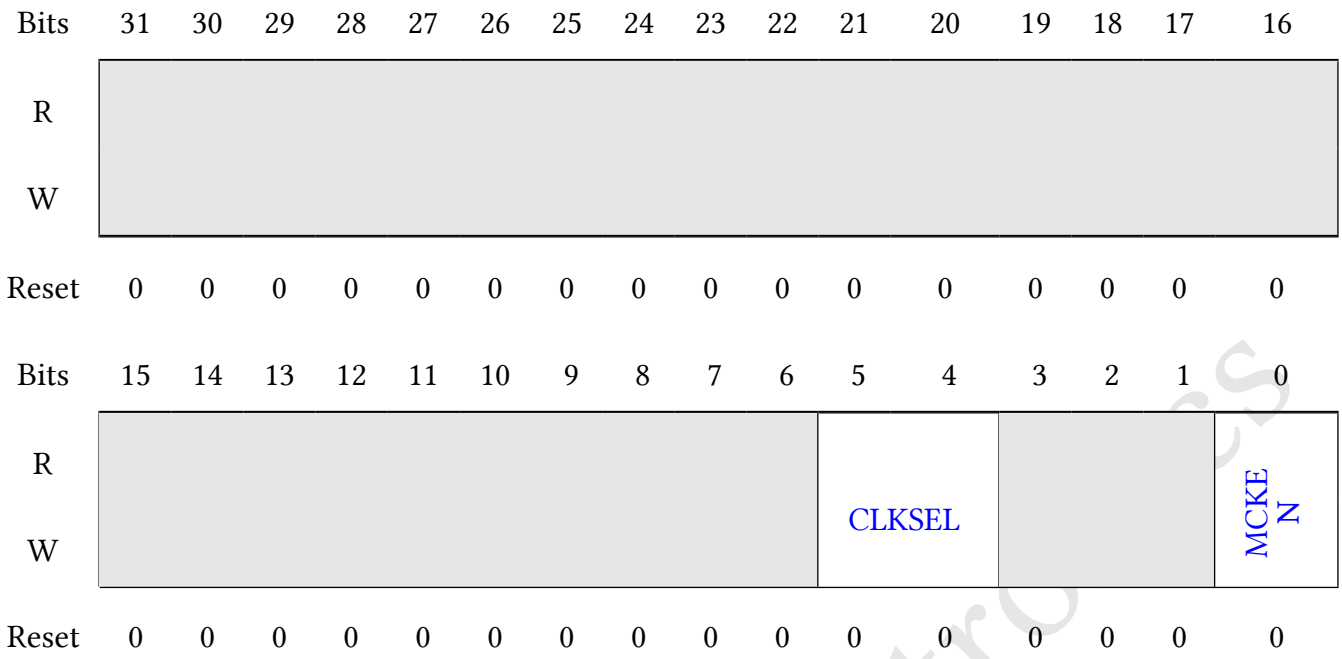


Table 26.29: SAI MCLK_CTRL Register Description

Field	Function
5 - 4 CLKSEL	**MCLK Source Select** Configures the source of MCLK. 00b - Bus Clock selected. 01b - Master Clock 1 selected. 10b - Master Clock 2 selected. 11b - Master Clock 3 selected.
0 MCKEN	**MCLK Enable** Enable the MCLK.

Yuntu Microelectronics

26.3.1.24 SAI MCLK_FRAC_DIV Register

Offset: 104h

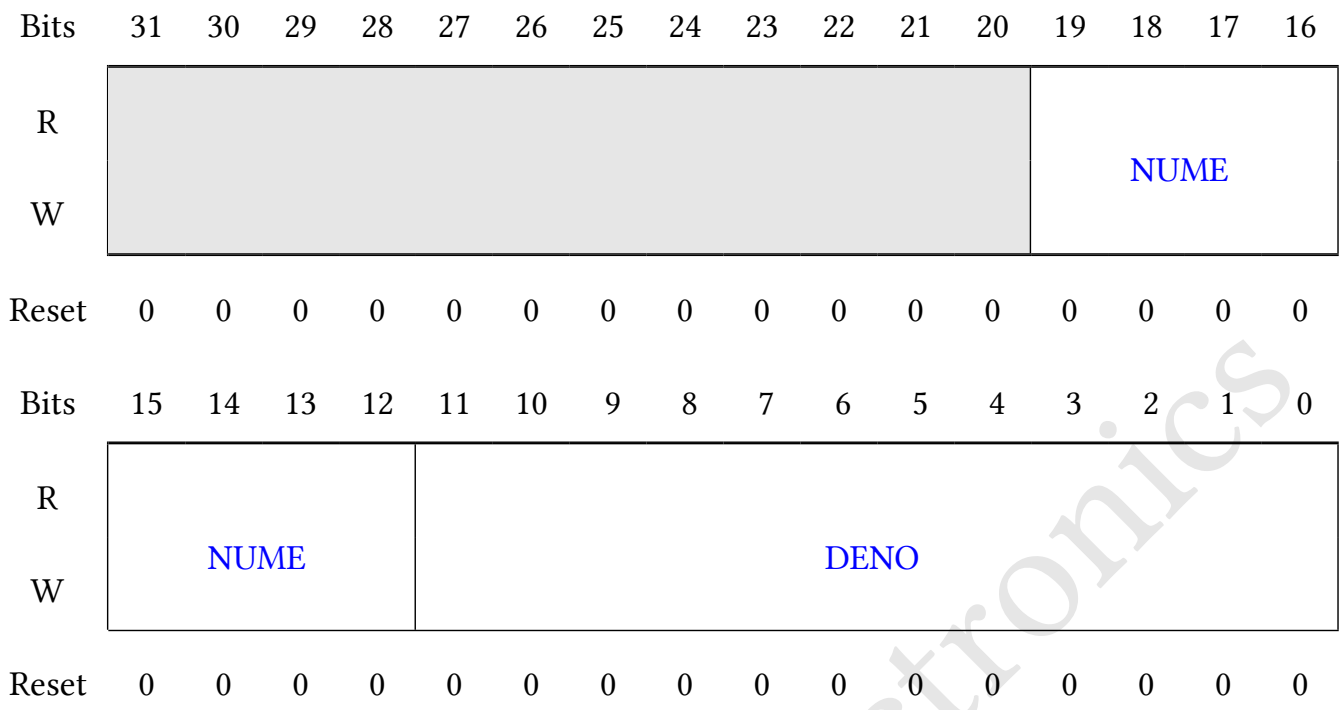


Table 26.30: SAI MCLK_FRAC_DIV Register Description

Field	Function
19 - 12 NUME	**Numerator** Indicate the numerator part of the fractional divider. The numerator is (NUME+1).
11 - 0 DENO	**Denominator** Indicate the denominator part of the fractional divider. The denominator is (DENO+1).

26.4 Function

26.4.1 Master Clock

The master clock is the clock source to generate an internal bit clock for both transmitter and receiver. The master clock can be generated internally or supplied externally.

26.4.1.1 Internal Master Clock

SAI has a fractional clock divider to generate the internal master clock, "see SAI MCLK FRAC DIV Register". The divider has four raw clock sources which can be selected by MCLK_CTRL[CLKSEL]. The denominator should be larger than the numerator.

$$F_{mclk} = \frac{F_{mclk_raw} * (NUME + 1)}{(DEMO + 1)}$$

Yuntu Microelectronics

26.4.1.2 External Master Clock

The external master clock is from the MCLK pin input. When MCLK_CTRL[MCKEN] is disabled, the master clock divider is bypassed and the SAI can use the external master clock directly.

26.4.2 Synchronous Mode

26.4.2.1 Synchronous Internal

The synchronous internal mode is only supported when the SAI has separate master and slave pins. The transmitter and the receiver can operate with synchronous bit clock and frame sync. If both the transmitter and receiver use the transmitter bit clock and frame sync:

- Set TCFG0[SYNC] to 2'b00, set RCFG0[SYNC] to 2'b01.
- Enable the receiver in synchronous mode only after both the transmitter and receiver are enabled.
- Enable the transmitter last and disable the transmitter first.

If both the transmitter and receiver use the receiver bit clock and frame sync:

- Set RCFG0[SYNC] to 2'b00, set TCFG0[SYNC] to 2'b01.
- Enable the transmitter in synchronous mode only after both the receiver and transmitter are enabled.
- Enable the receiver last and disable the receiver first.

Note: Synchronous mode only makes sense when the transmitter and receiver can work at the same time.

26.4.2.2 Synchronous External

The transmitter and the receiver can operate with the bit clock and frame sync from another SAI. Refer to TCFG0[SYNC] and RCFG0[SYNC] for details.

26.4.3 Bit Clock

The SAI can provide asynchronous free-running bit clocks that can be generated internally from a master clock or supplied externally for both transmitter and receiver.

26.4.4 Frame Sync

The SAI transmits or receives the data frame by frame. Each frame consists of a fixed number of words and each word consists of a fixed number of bits. The number of bits can be configured for the first word and the other words independently. Within each frame, any word can be masked (See "SAI_TMR" and "SAI_RMR")so that the transmitter will keep data lines low and the receiver will discard this word.

The frame sync signal indicates the start of each frame. The transmitter and receiver frame sync can be configured independently with any of the following options:

- Generated internally or externally
- Active high or low
- Assert before the first bit in the frame or assert one bit early
- Duration is between 1-bit clock and the first-word length
- Each frame has 1 to 16 words.
- Each word has 8 to 32 bits.

Yuntu Microelectronics

26.4.5 Data Lines

Each SAI can have up to 4 data lines. Each data channel has its own data FIFO. Both transmit data and receive data can be MSB first or LSB first.

26.4.6 Data FIFO

The SAI has an 8 x 32-bit size FIFO for each transmit and receive channel.

26.4.6.1 Data Alignment

Data in the FIFO can be aligned anywhere within the 32-bit wide register.

- The FBT(First Bit Shifted) field selects the bit index of the first bit shifted.
- The MF(MSF First) field configures whether LSB or MSB is selected.

26.4.6.2 FIFO Packing

It supports storing multiple 8-bit data and 16-bit data into a 32-bit word when the FPACK bit is enabled. In FIFO packing mode, the FIFO pointer only changes when a full 32-bit word has been transmitted or received.

When using 16-bit FIFO packing for transmitting, the transmit shift register loads a 32-bit word from FIFO at the start of the frame and after every two unmasked transmit words have been transmitted. The first word transmitted is taken from the 16-bit word at byte offset 0x0. The first bit is selected by FBT and the last bit is selected by FBT and word size, while they must be configured with this 16-bit word. The second word transmitted is taken from the 16-bit word at byte offset 0x2. The first bit and the last bit control is like the first word.

When using 16-bit FIFO packing for receiving, the receive shift register is stored into FIFO after every two unmasked received words and at the end of the frame if the number of the unmasked received words is odd. The first word received is stored in the 16-bit word at byte offset 0x0. The first bit is selected by FBT and the last bit is selected by FBT and word size, while they must be configured with this 16-bit word. The second word received is stored in the 16-bit word at byte offset 0x2. The first bit and the last bit control is like the first word.

When using 8-bit FIFO packing, it is similar to 16-bit FIFO packing. Four words are loaded from or stored into each 32-bit FIFO word. The first word is loaded or stored in byte offset 0x0, while the second in 0x1, the third in 0x2 and the fourth in 0x3.

26.4.6.3 FIFO Combine

It supports allowing the separate FIFOs from multiple data channels to be used as a single FIFO. The enabled data channels must be contiguous and data channel 0 must be enabled when using FIFO Combine mode.

In FIFO combine mode, the software read and write for all the FIFOs of the enabled channels should access the register entry of channel 0 FIFO. The first access to the register entry will access the FIFO of channel 0, and the second access to the register entry will access the FIFO of channel 1. This continues until the last enabled channel FIFO and the pointer resets back to the FIFO of channel 0. FIFO software reset and FIFO combine mode disabled will also reset the pointer.

Combining FIFOs for transmit data channels allows one data channel to use all enabled channel FIFOs. The transmit shift register for all enabled data channels transmit the same data at the same time. They

Yuntu Microelectronics

load data from the start of the frame and after every N(the number of the enabled data channels) unmasked words. The first unmasked word transmitted is loaded from the FIFO of channel 0, the second word is loaded from the FIFO of channel 1 and it continues until the last enabled channel. It should be noted that the first word in each frame always loads from data channel 0.

Combining FIFOs for receive data channels allows data channel 0 to use all enabled channel FIFOs. After receiving shift register of channel 0 receives N(the number of the enabled data channels) unmasked words, they will be stored in the enabled FIFOs. The first word received stores to the first enabled channel FIFO, while the second word received stores to the second enabled channel FIFO and so on. At the end of the frame, even if the data in receive shift register of channel 0 is less than N words, they will be stored in FIFOs.

The combining FIFOs always store and load data at the same time, so that the FIFO warning and request flags assert if it occurs to any of the enabled FIFOs.

26.4.6.4 FIFO Request

The transmit FIFO request flag sets when the number of entries in any of the enabled transmit FIFO is less than or equal to the transmit FIFO watermark configuration and clears when the number of the transmit FIFO is greater than the transmit FIFO watermark configuration. The receive FIFO request flag sets when the number of entries in any of the enabled receive FIFO is greater than the receive FIFO watermark configuration and clears when the number of entries in the receive FIFO is less than or equal to the receive FIFO watermark configuration.

26.4.6.5 FIFO Warning

The transmit warning flag sets when the number of entries in any of the enabled transmit FIFOs is empty and clears when the number of entries in the transmit FIFO is not empty. The receive warning flag sets when the number of entries in any of the enabled receive FIFOs is full and clears when the number of entries in the receive FIFO is not full.

26.4.6.6 FIFO Error

The transmit FIFO error flag TSTS[FEF] sets when any of the enabled transmit FIFOs underrun. After the error flag sets, all enabled transmit channels transmit zero data before TSTS[FEF] clears. The receive FIFO error flag TSTS[REF] sets when any of the enabled receive FIFOs overflow. After the flag is set, all enabled receive channels discard received data until TSTS[REF] clears and the next receive frame starts. Empty all enabled receive FIFOs before RCSR[FEF] clears.

26.4.7 Interrupt

The SAI transmitter and receiver support can generate the asynchronous interrupts separately.

- FIFO request interrupt
- FIFO warning interrupt
- FIFO error interrupt
- Sync error interrupt
- Word start interrupt

Yuntu Microelectronics

26.4.8 DMA

The SAI supports DMA and has two DMA request sources. Each transmit and receive data channel can choose FIFO request or FIFO warning as the DMA request independently.

26.5 Glossary

SAI	Serial Audio Interface
FIFO	First-in first-out
LSB	Least Significant Bit
MSB	Most Significant Bit
AC'97	Audio Codec 1997
TDM	Time Division Multiplexing
PCM	Pulse Code Modulation

Yuntu Microelectronics

Part VII

Timer

Yuntu Microelectronics

Chapter 27

Timer (TMR)

27.1 Chip-specific Information

27.1.1 Instantiation Information

YTM32B1HA0x contains two TMRs.

27.2 Introduction

The Timer Module (TMR) supports commonly required system and application software timing functions. TMR includes a 32-bit count-up timer and four 32-bit compare channels with a separate interrupt source for each channel. The timer is driven by the TMR module clock divided by an 8-bit prescale value (1 to 256).

27.2.1 Features

TMR has the following features:

- One 32-bit count-up timer with an 8-bit prescaler
- Four 32-bit compare channels
- An independent interrupt source for each channel
- Ability to stop the timer in debug mode

Yuntu Microelectronics

27.2.2 Block Diagram

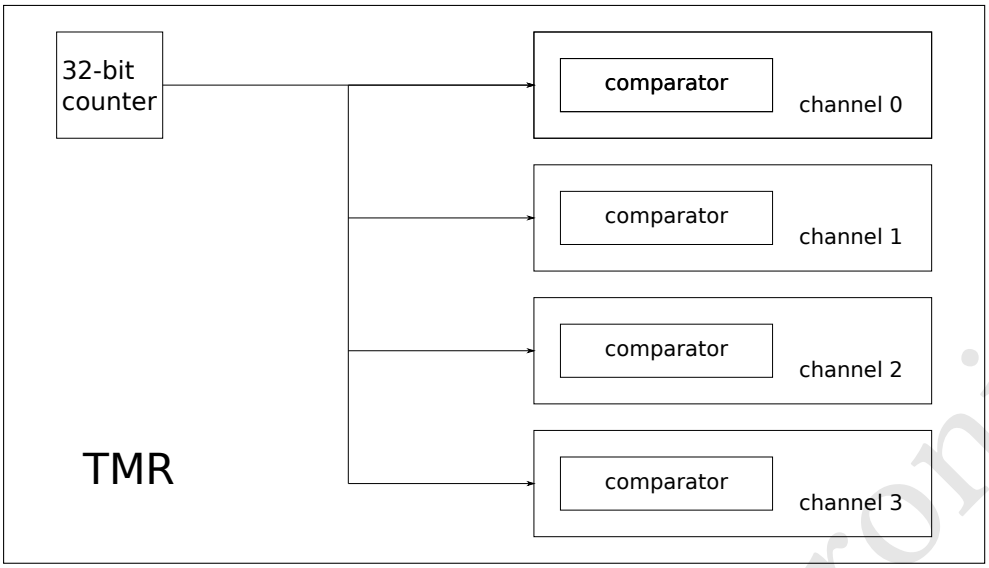


Figure 27.1: TMR Block Diagram

27.3 Register Definition

This section includes the TMR module memory map and detailed descriptions of all registers.

The TMR allows 32/16/8-bit accesses. An attempted reference to a reserved address generates a bus error termination.

27.3.1 TMR Memory Map

TMR0 base address 40150000h
TMR1 base address 40154000h

Table 27.1: TMR Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Prescaler Register (PRS)	32	RW	00000000h
8h	Count Register (CNT)	32	RW	00000000h
10h	Channel Control Register (CH0_CTRL)	32	RW	00000000h
14h	Channel Interrupt Register (CH0_INT)	32	RW	00000000h
18h	Channel Compare Register (CH0_CMP)	32	RW	00000000h
20h	Channel Control Register (CH1_CTRL)	32	RW	00000000h
24h	Channel Interrupt Register (CH1_INT)	32	RW	00000000h
28h	Channel Compare Register (CH1_CMP)	32	RW	00000000h
30h	Channel Control Register (CH2_CTRL)	32	RW	00000000h
34h	Channel Interrupt Register (CH2_INT)	32	RW	00000000h
38h	Channel Compare Register (CH2_CMP)	32	RW	00000000h
40h	Channel Control Register (CH3_CTRL)	32	RW	00000000h
44h	Channel Interrupt Register (CH3_INT)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
48h	Channel Compare Register (CH3_CMP)	32	RW	00000000h

27.3.1.1 TMR CTRL Register

Offset: 0h

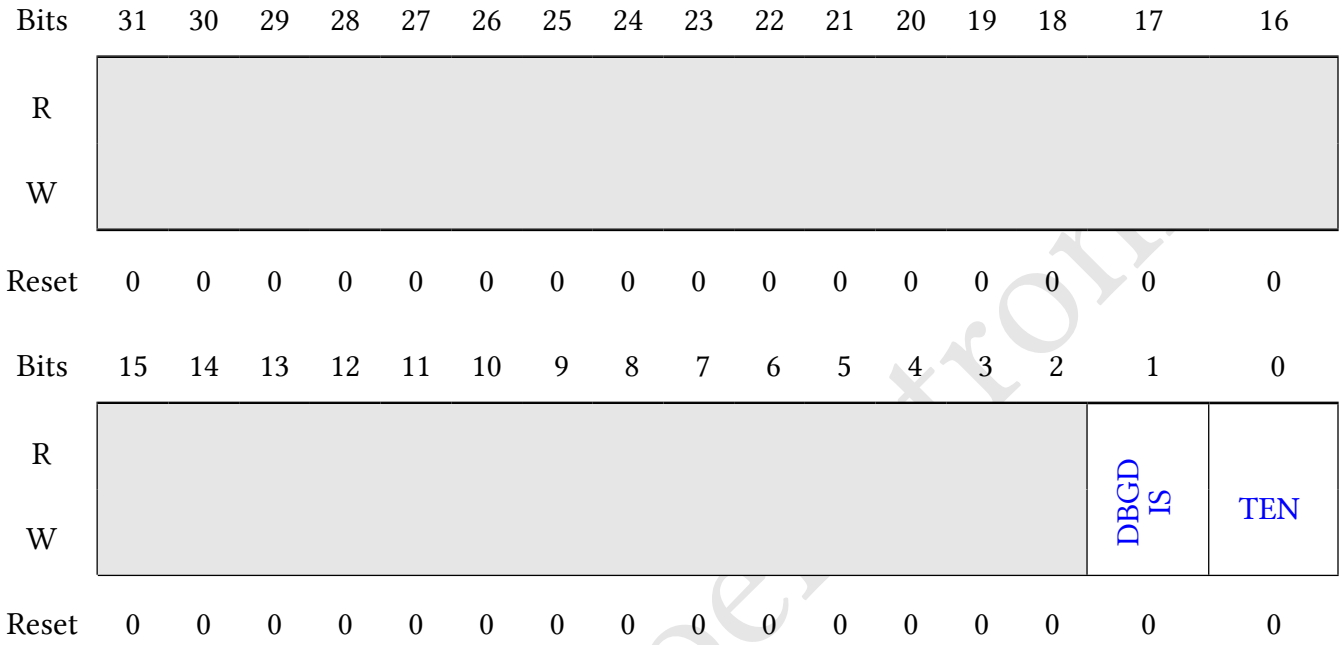


Table 27.2: TMR CTRL Register Description

Field	Function
1 DBGDIS	**Debug Disable** Stops the timer when the chip enters debug mode. 0b - The timer runs in debug mode. 1b - The timer stops in debug mode.
0 TEN	**Timer Enable** Enable the timer. 0b - The timer is disabled. 1b - The timer is enabled.

27.3.1.2 TMR PRS Register

Offset: 4h

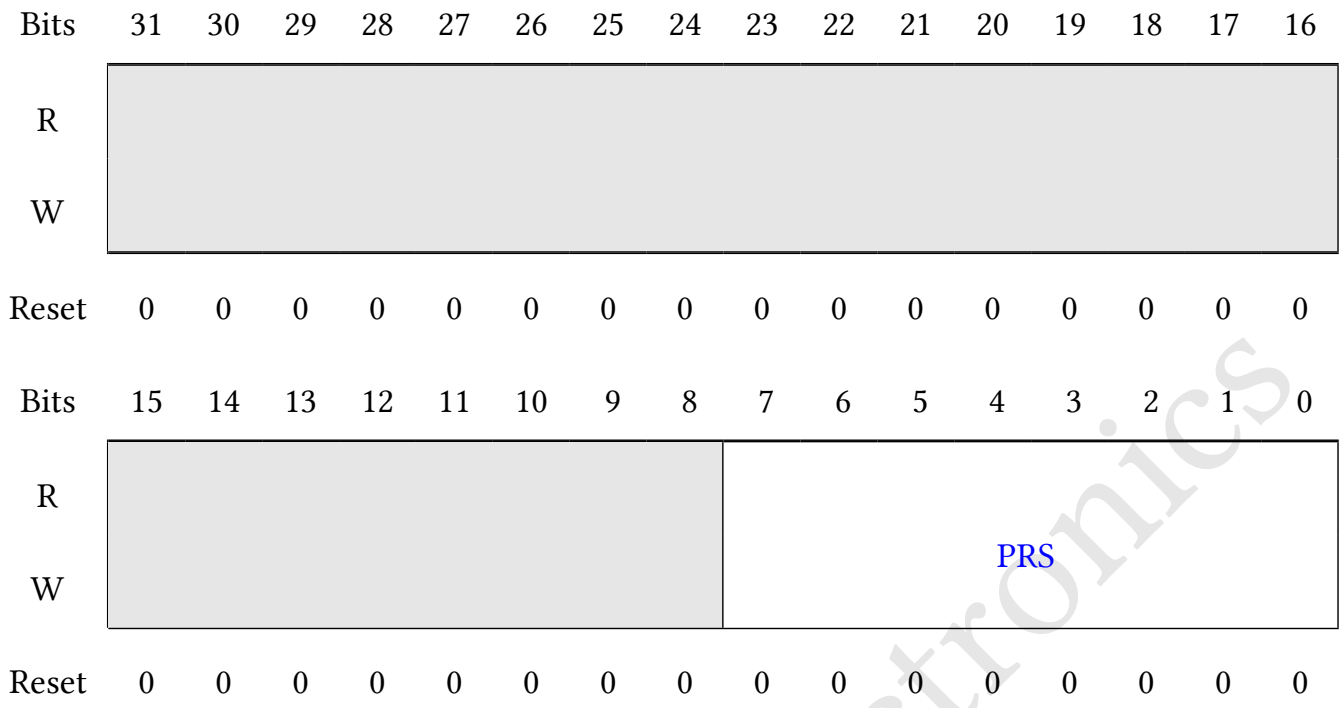


Table 27.3: TMR PRS Register Description

Field	Function
7 - 0 PRS	**Timer Prescaler** Selects the module clock divide value for the prescaler (1–256). The module clock is divided by (PRS + 1)

Yuntu Microelectronics

27.3.1.3 TMR CNT Register

Holds the timer count value.

Offset: 8h

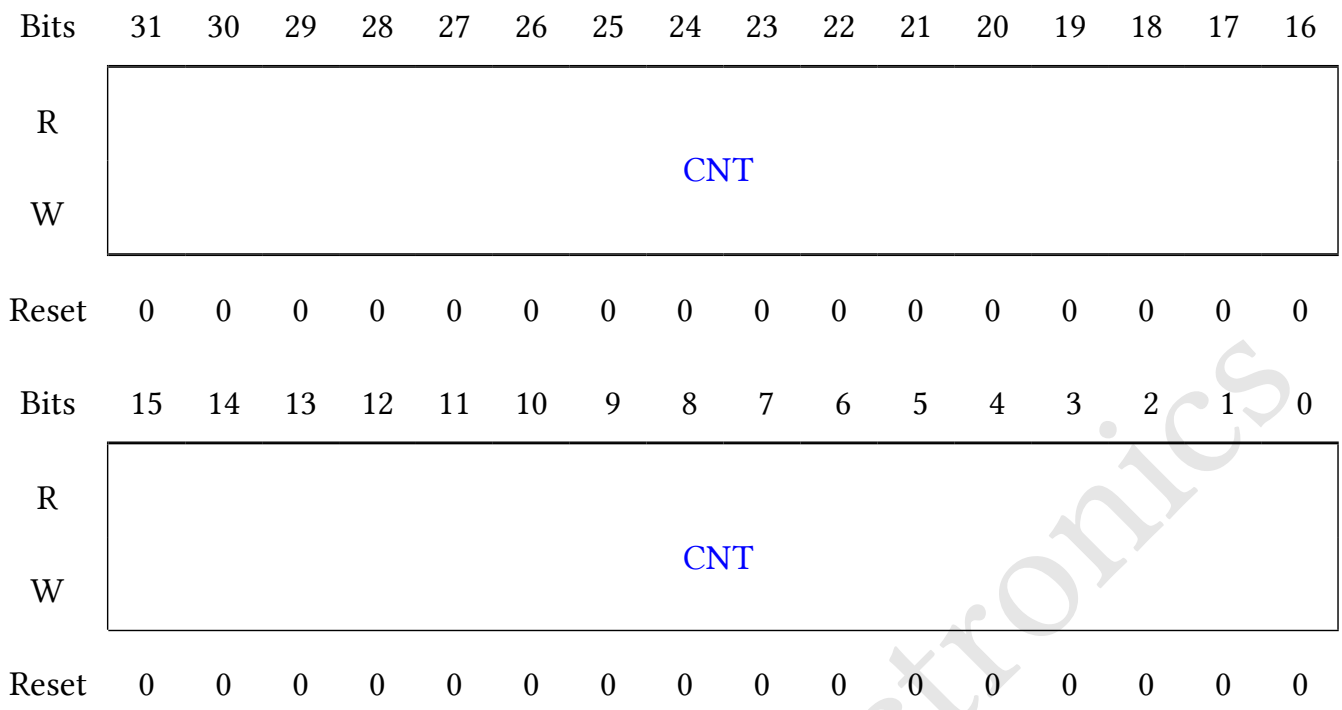


Table 27.4: TMR CNT Register Description

Field	Function
31 - 0 CNT	**Timer Count** The time base for all compare channels. When enabled, the timer count increments at the rate of the module clock divided by the prescale value. Writing operation to this register will load writing value to it immediately.

Yuntu Microelectronics

27.3.1.4 TMR CH_CTRL Register

Offset: 10h + n * 10h

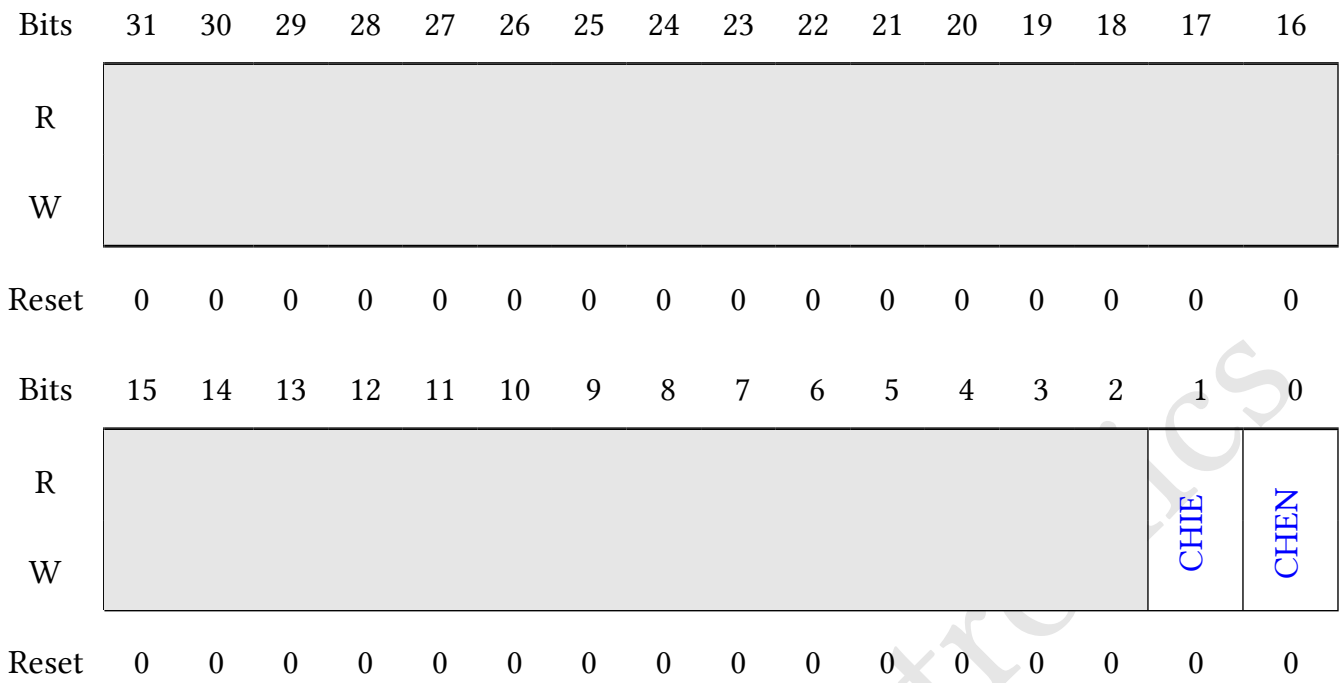


Table 27.5: TMR CH_CTRL Register Description

Field	Function
1 CHIE	**Channel Interrupt Enable** 0b - Channel interrupt is disabled. 1b - Channel interrupt is enabled.
0 CHEN	**Channel Enable** 0b - Channel is disabled. 1b - Channel is enabled.

Yuntu Microelectronics

27.3.1.5 TMR CH_INT Register

Offset: 14h + n * 10h

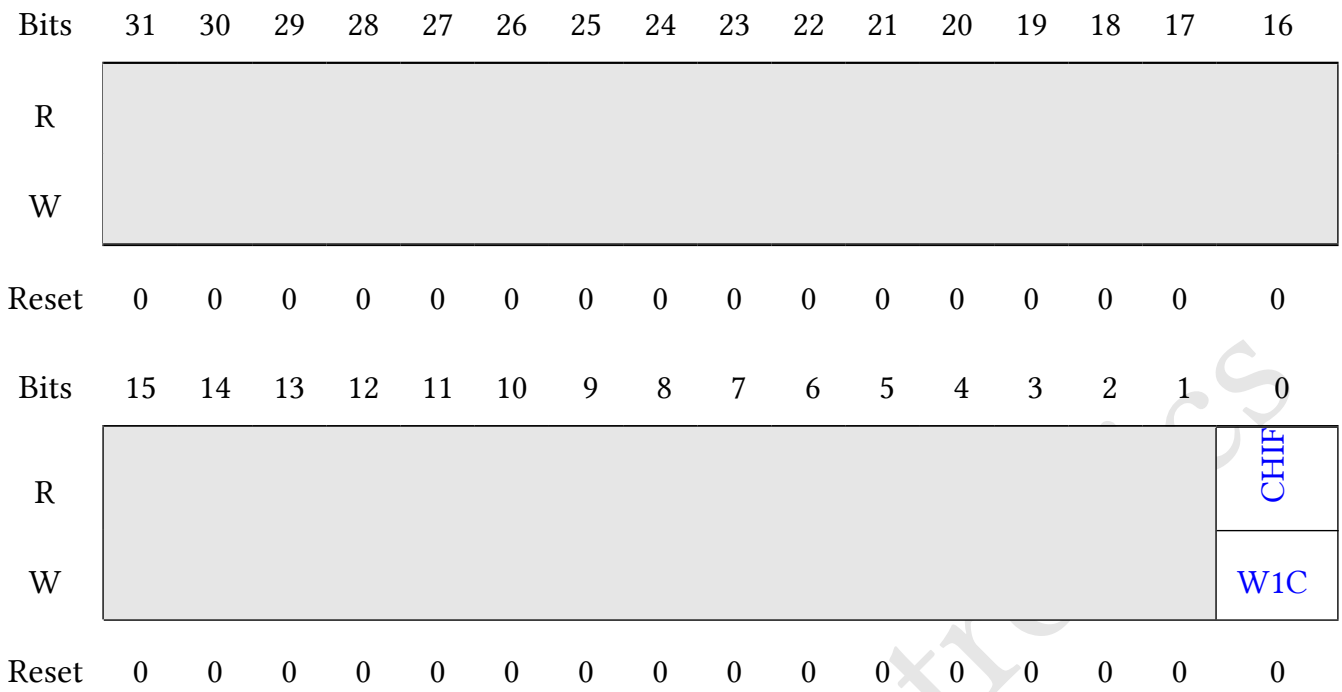


Table 27.6: TMR CH_INT Register Description

Field	Function
0 CHIF	**Channel Interrupt Flag** Indicates the channel interrupt is asserted due to a match on the channel. Write 1 to clear it. 0b - Channel interrupt is not asserted. 1b - Channel interrupt is asserted.

27.3.1.6 TMR CH_CMP Register

Offset: 18h + n * 10h

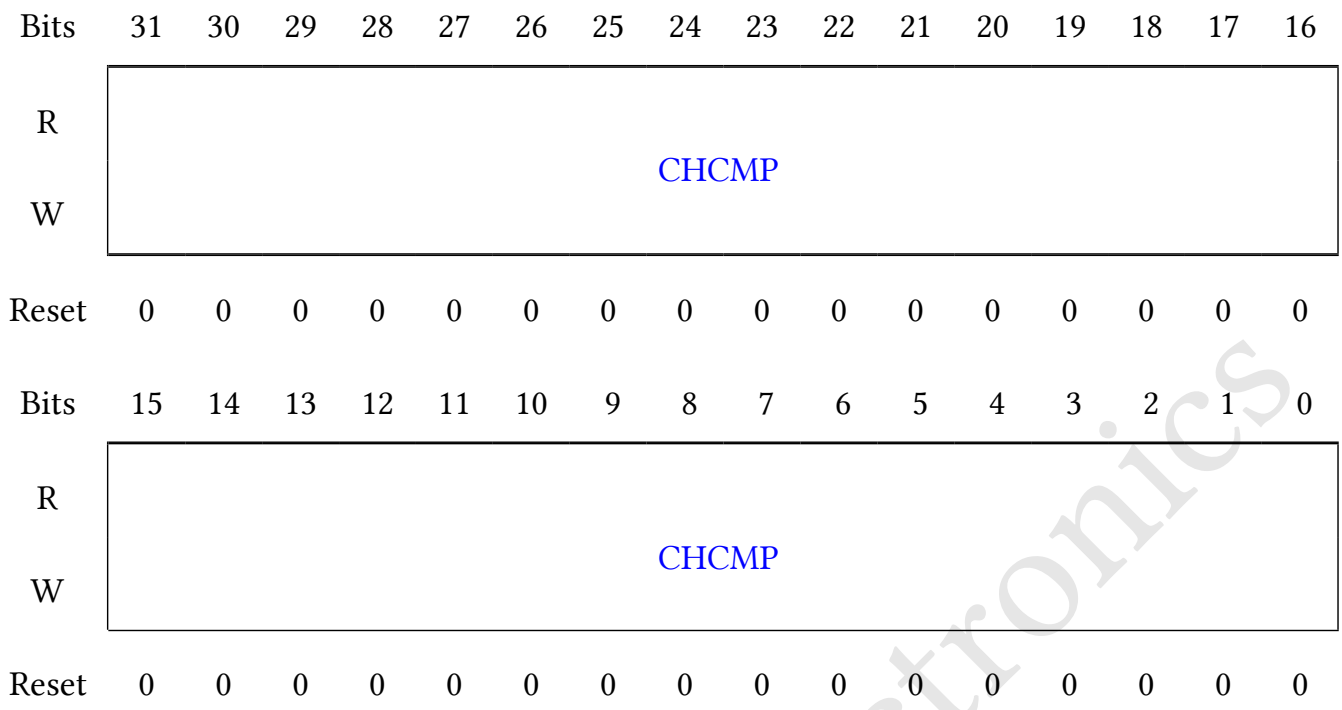


Table 27.7: TMR CH_CMP Register Description

Field	Function
31 - 0 CHCMP	**Channel Compare** If the channel is enabled (CH_CTRL[CHEN]), when the timer count (CNT) matches this value, TMR sets the channel interrupt flag (CH_INT[CHIF]). If the channel interrupt is enabled (CH_CTRL[CHIE]), TMR asserts the channel interrupt.

27.4 Functional Description

27.4.1 Count-up Timer

TMR has one 32-bit count-up timer which is used for four channels. When enabled, the counter increments at the module clock divided by a prescaler value which is from 1 to 256. When enabled in Normal mode, the timer increments continuously. The counter rolls over at FFFF_FFFFh to 0000_0000h with no restrictions at this boundary.

27.4.2 Compare Channels

TMR has four identical compare channels. Each channel contains a channel control register(CH_CTRL), an interrupt flag register(CH_INT) and a channel compare register (CH_CMP). When the channel and channel interrupt are enabled and its channel compare value matches the timer count, TMR sets the channel interrupt flag and generates an interrupt request on that channel.

27.4.3 Operation Modes

TMR supports the chip operation modes of operation as follows:

Yuntu Microelectronics

- Normal: When the timer is enabled (CTRL[TEN] = 1), the timer counts up continuously.
- Debug: If CTRL[DBGDIS] = 1, counter stops in debug mode. Counter counts as normal mode by default.

27.4.4 Configure the Timer

1. Set the initial counter value(CNT[CNT]).
2. Specify TMR behavior in chip Debug mode (CTRL[DBGDIS]).
3. Set the counter prescaler (PRS[PRS]).
4. Start the timer (CTRL[TEN]).

27.4.5 Configure the Compare Channels

For each compare channel:

1. Set the channel compare value (CH_CMP[CHCMP]).
2. Enable the compare channel (CH_CTRL[CHEN]).
3. Optional enable the channel interrupt (CH_CTRL[CHIE]).

27.4.6 Respond to Compare Channel Events

For each compare channel:

1. Check the channel interrupt flag (CH_INT[CHIF]).
2. If the channel interrupt flag is set, interrupt request is responded.
3. When the channel interrupt has been handled, clear the channel interrupt flag (CH_INT[CHIF]).

Yuntu Microelectronics

Chapter 28

Periodic Timer (pTMR)

28.1 Chip-specific pTMR Information

28.1.1 Instantiation Information

YTM32B1HA0x contains 3 pTMR modules with four channels of 32bit timers that can be used to count and raise an interrupt.

28.2 Introduction

28.2.1 Block Diagram

The following figure shows the pTMR block diagram.

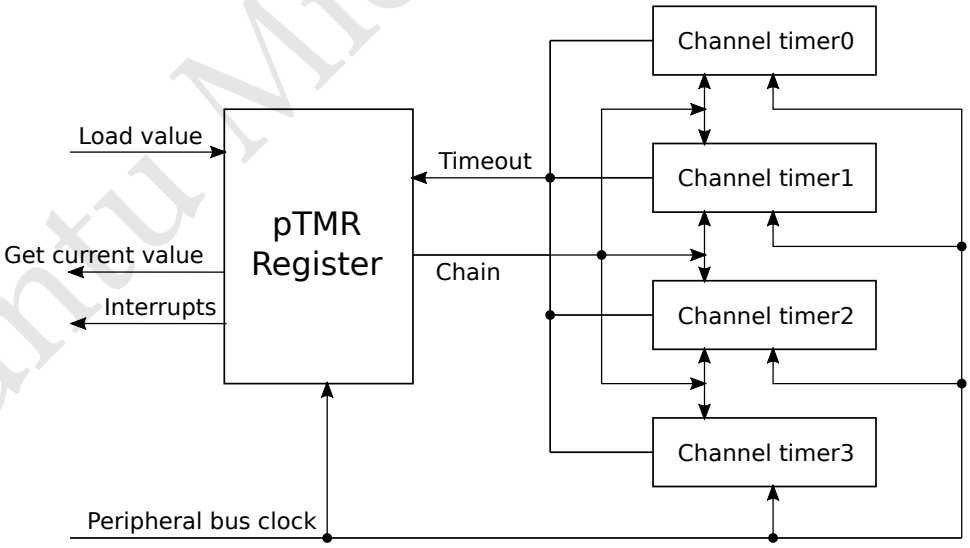


Figure 28.1: Block Diagram of pTMR

28.2.2 Features

- Four channels of 32bit timers, each timer has independent timeout periods
- Timers can generate interrupts, and each channel can generate independent interrupt request

Yuntu Microelectronics

- Support chain mode to connect multiple timer to a longer timer
- Ability to stop in debug mode

28.3 Register Definition

The pTMR includes registers to control pTMR module enable or disable and control each channel timer enable or disable, load start value, get current count value, and enable or disable timer interrupt and flag. And it also can link 2/3/4 channels as a whole.

Accessing a reserved address will result in a bus error.

Register_address = Base_address + Address_offset
For example: as CHn_TSV
Address_offset = 0x100 + n * 0x10 (n=0/1/2/3)

28.3.1 pTMR Memory Map

pTMR0 base address 40168000h
pTMR1 base address 4016c000h
pTMR2 base address 40170000h

Table 28.1: pTMR Memory Map

Offset	Register	Width	Access	Reset Value
0h	Module Control Register (MCR)	32	RW	00000000h
100h	Timer Start Value Register (CH0_TSV)	32	RW	00000000h
104h	Timer Current Value Register (CH0_TCV)	32	RW	00000000h
108h	Timer Control Register (CH0_TCR)	32	RW	00000000h
10ch	Timer Flag Register (CH0_TFR)	32	RW	00000000h
110h	Timer Start Value Register (CH1_TSV)	32	RW	00000000h
114h	Timer Current Value Register (CH1_TCV)	32	RW	00000000h
118h	Timer Control Register (CH1_TCR)	32	RW	00000000h
11ch	Timer Flag Register (CH1_TFR)	32	RW	00000000h
120h	Timer Start Value Register (CH2_TSV)	32	RW	00000000h
124h	Timer Current Value Register (CH2_TCV)	32	RW	00000000h
128h	Timer Control Register (CH2_TCR)	32	RW	00000000h
12ch	Timer Flag Register (CH2_TFR)	32	RW	00000000h
130h	Timer Start Value Register (CH3_TSV)	32	RW	00000000h
134h	Timer Current Value Register (CH3_TCV)	32	RW	00000000h
138h	Timer Control Register (CH3_TCR)	32	RW	00000000h
13ch	Timer Flag Register (CH3_TFR)	32	RW	00000000h

28.3.1.1 pTMR MCR Register

Offset: 0h

Yuntu Microelectronics

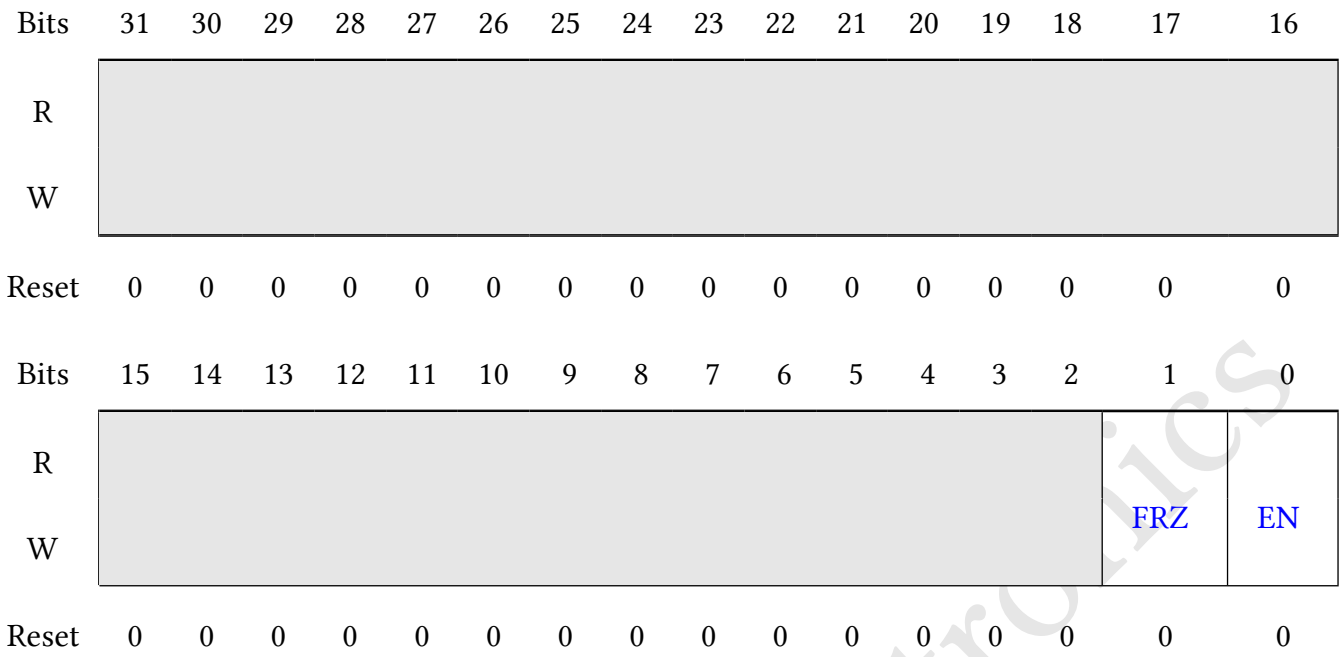


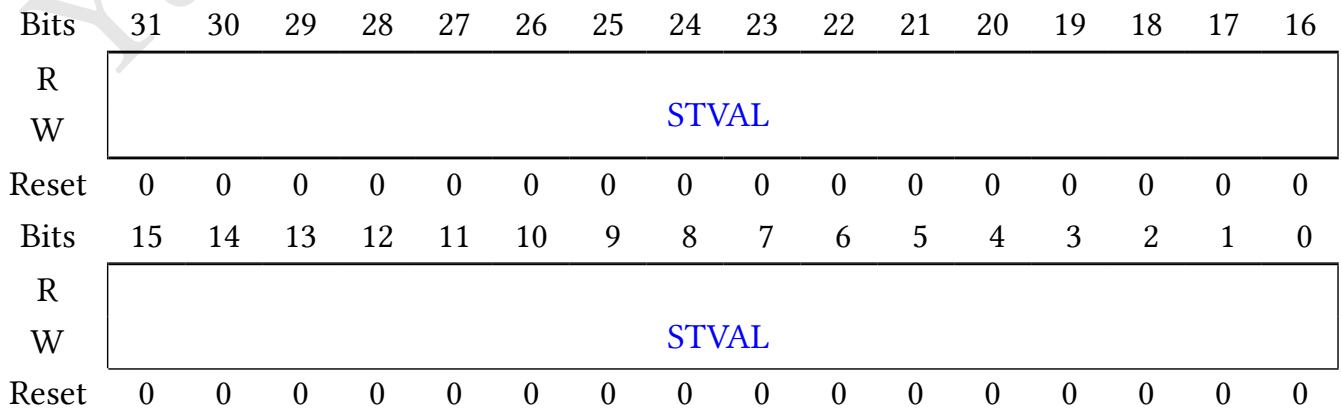
Table 28.2: pTMR MCR Register Description

Field	Function
1 FRZ	**Freeze** Allows the timers to be stopped when the device enters debug mode. 0b - pTMR continues to run in debug mode 1b - pTMR stops in debug mode
0 EN	**pTMR Enable** Enable pTMR module 0b - pTMR is disabled 1b - pTMR is enabled

28.3.1.2 pTMR CH_TSV Register

Timer Start Value Register

Offset: 100h + n * 10h



Yuntu Microelectronics

Table 28.3: pTMR CH_TSV Register Description

Field	Function
31 - 0 STVAL	**Start Value** This register set the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and loads the value again. Writing a new value to this register will not restart the timer, instead the value will be loaded once the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

28.3.1.3 pTMR CH_TCV Register

Timer Current Value Register

Offset: 104h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28.4: pTMR CH_TCV Register Description

Field	Function
31 - 0 CVAL	**Current Value** This register stands for current timer value when user read this register. The value of this register will be frozen in debug mode if MCR[FRZ] is set.

28.3.1.4 pTMR CH_TCR Register

Timer Control Register

Offset: 108h + n * 10h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W														CHAIN	TIE	TEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 28.5: pTMR CH_TCR Register Description

Field	Function
2 CHAIN	**Chain Mode** When setting this bit, it will link multi-channel in sequence to combine a new bits counter, it just links the front channel. 0b - Chaining is disabled, channel timer runs independently 1b - Chaining is enabled
1 TIE	**Timer Interrupt Enable** Enable interrupt when setting this bit 0b - Disable timer interrupt of corresponding channel 1b - Enable timer interrupt of corresponding channel
0 TEN	**Timer Enable** Enable the timer of corresponding channel 0b - Disable the timer of corresponding channel 1b - Enable the timer of corresponding channel

28.3.1.5 pTMR CH_TFR Register

Timer Flag Register

Offset: 10ch + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																TIF
W																W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 28.6: pTMR CH_TFR Register Description

Field	Function
0 TIF	**Timer Interrupt Flag** Hardware will set this bit as 1 when timer interrupt is enabled, and writing logic 1 to this bit will clear this bit as 0. 0b - Timeout did not occur 1b - Timeout occurred

28.4 Functional Description

28.4.1 Timers

These timers load start values in the CH_TSV register, then count down until it reaches 0, then they load their respective start value. Once one timer reaches 0, it will generate an interrupt and set the interrupt flag.

Interrupt can be enabled or masked by setting TIE bit in the CH_TCR register of each channel. A new interrupt can be generated after the former one is cleared. When user wants to clear the interrupt flag, TIF bit in the CH_TFR register should be set. When it is set to 1, it will be cleared.

The current counter value of each channel timer could be read from CH_TCV register of its own.

The counter period can be restarted by firstly disabling, then enabling the timer with the TEN bit in the CH_TCR register.

The counter period of a running timer can be modified, by firstly disabling the timer, setting a new value to load and then enabling the timer again.

28.4.2 Chain Mode

These timers can be linked as a longer timer, setting CHAIN bit in the CH_TCR register to 1 will enable the chain mode. In this mode, timers only link with the front one, and timeout interrupt of each timer can work independently. For example, if set CHAIN in CH1_TCR, then timer0 and timer1 are linked to a 64-bit timer, timer1 timeout will be longer.

28.4.3 Debug Mode

These timers will be frozen in the debug mode when MCR[FRZ] bit is set to enable. Its purpose is to aid software engineer to halt the processor, investigate the current state of the system and then continue the operation.

28.4.4 Interrupt

pTMR module supports four interrupts generated by four channels, each channel can generate interrupt individually. Please refer to the table of interrupt vectors. Timer interrupt can be disabled by setting the TIE bit to 0. The timer interrupt flag(TIF bit) is set to 1 while timeout occurs, and interrupt flag can be cleared by setting TIF bit from CH_TFR register to 1.

Yuntu Microelectronics

Chapter 29

Low Power Timer (lpTMR)

29.1 Introduction

The low-power timer (lpTMR) can be configured to operate as a time counter with optional prescaler, or as a pulse counter with optional glitch filter, across all power modes.

29.1.1 Features

The features of the lpTMR module include:

- 16-bit time counter or pulse counter with compare
- Optional interrupt can generate asynchronous wakeup from any low-power mode
- Hardware trigger output
- Counter supports free-running mode or reset on compare
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter
  - Rising-edge or falling-edge

29.2 Register Definition

This subsection provides a description of the memory map and registers.

29.2.1 lpTMR Memory Map

lpTMR0 base address 40174000h

Table 29.1: lpTMR Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Prescaler Register (PRS)	32	RW	00000000h
8h	DMA and Interrupt Enable Register (DIE)	32	RW	00000000h
ch	Status Register (STS)	32	RW	00000000h
10h	Compare Value Register (CMP)	32	RW	00000000h
14h	Latch Timer Counter Register (LCNT)	32	WO	00000000h
18h	Timer Current Counter Register (CNT)	32	RW	00000000h

Yuntu Microelectronics

29.2.1.1 lpTMR CTRL Register

Offset: 0h

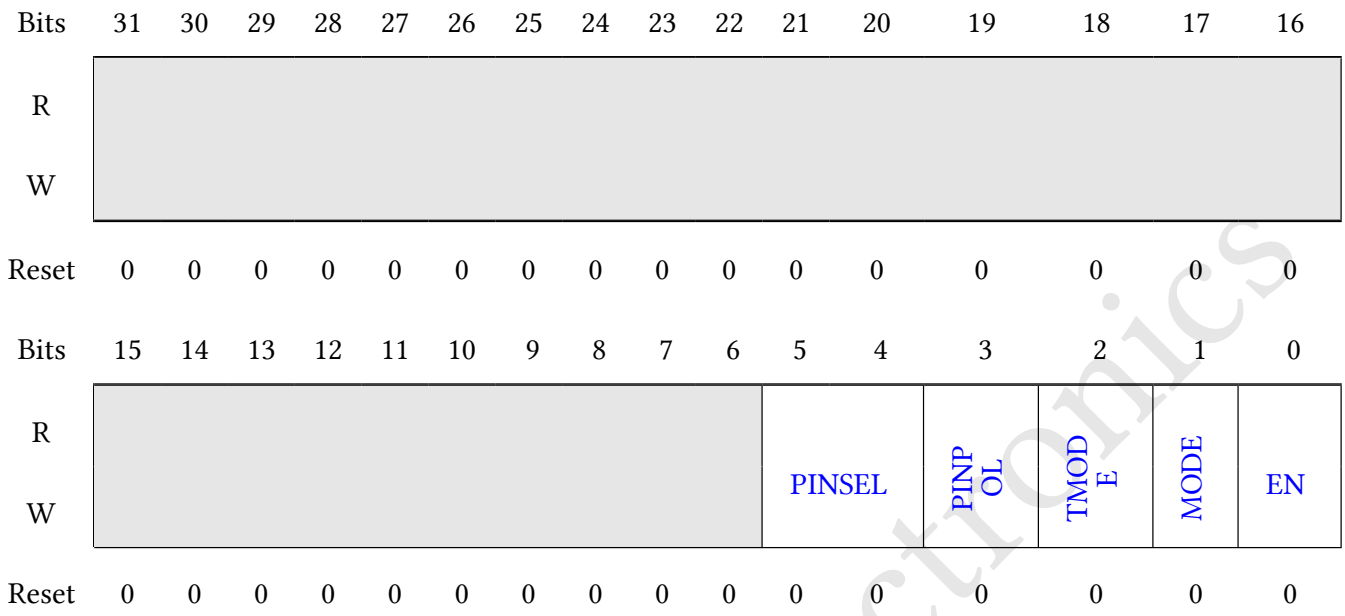


Table 29.2: lpTMR CTRL Register Description

Field	Function
5 - 4 PINSEL	**Timer Pin Select** Configures the input source to be used in Pulse Counter mode. PINSEL must be altered only when the lpTMR is disabled. 00b - Pulse counter input 0 is selected. (It's from TMU) 01b - Pulse counter input 1 is selected. (It's from lpTMR_ALT1 pad) 10b - Pulse counter input 2 is selected. (It's from lpTMR_ALT2 pad) 11b - Pulse counter input 3 is selected. (It's from lpTMR_ALT3 pad)
3 PINPOL	**Timer Pin Polarity** Configures the polarity of the input source in Pulse Counter mode. PINPOL must be changed only when the lpTMR is disabled. 0b - Pulse Counter input source is active-high, and the CNT will increment on the rising-edge. 1b - Pulse Counter input source is active-low, and the CNT will increment on the falling-edge.
2 TMODE	**Timer Free-Running Mode** When clear, TMODE configures the CNT to reset whenever CCF is set. When set, TMODE configures the CNT to reset on overflow. TMODE must be altered only when the lpTMR is disabled. 0b - CNT is reset whenever CCF is set. 1b - CNT is reset on overflow.
1	**Timer Mode Select**

Yuntu Microelectronics

Field	Function
MODE	Configures the mode of the lpTMR. MODE must be altered only when the lpTMR is disabled. 0b - Time Counter mode. 1b - Pulse Counter mode.
0 EN	**Timer Enable** When EN is clear, it resets the lpTMR internal logic, including the CNT and CCF. When EN is set, the lpTMR is enabled. While writing 1 to this field, CTRL[5:1] must not be altered. 0b - lpTMR is disabled and internal logic is reset. 1b - lpTMR is enabled.

29.2.1.2 lpTMR PRS Register

Offset: 4h

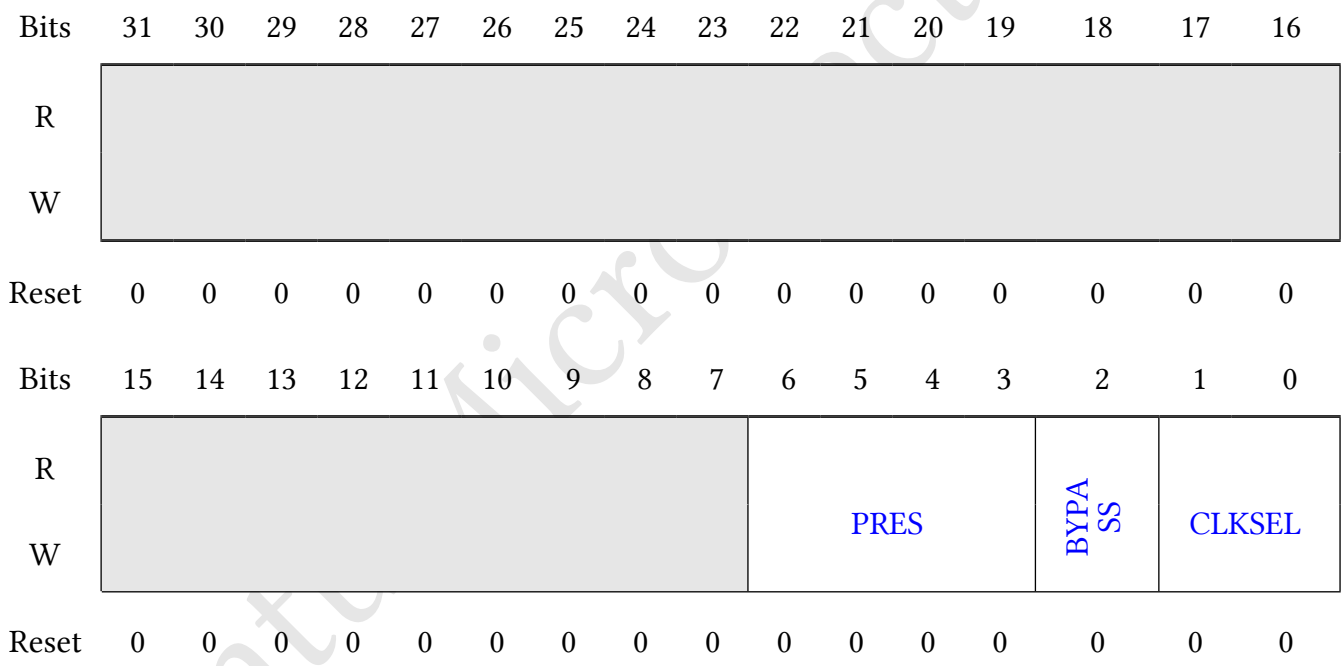


Table 29.3: lpTMR PRS Register Description

Field	Function
6 - 3	**Prescaler Value**

Yuntu Microelectronics

Field	Function
PRES	Configures the size of the Prescaler in Time Counter mode or width of the glitch filter in Pulse Counter mode. The width of the glitch filter can vary by 1 cycle due to synchronization of the pulse counter input. PRESCALE must be altered only when the lpTMR is disabled. 0000b - Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001b - Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010b - Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011b - Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100b - Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101b - Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110b - Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111b - Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges. 1000b - Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001b - Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010b - Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011b - Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100b - Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101b - Prescaler divides the prescaler clock by 16384; glitch filter recognizes change on input pin after 8192 rising clock edges. 1110b - Prescaler divides the prescaler clock by 32768; glitch filter recognizes change on input pin after 16384 rising clock edges. 1111b - Prescaler divides the prescaler clock by 65536; glitch filter recognizes change on input pin after 32768 rising clock edges.
2 BYPASS	**Prescaler Bypass** When BYPASS is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNT. When BYPASS is clear, the CNT is clocked by the output of the prescaler/glitch filter. BYPASS must be altered only when the lpTMR is disabled.
1 - 0 CLKSEL	**Clock Selection** 00b - func_clk (cannot be used in Powerdown mode) 01b - SIRC_DIV4_CLK 10b - SXOSC 11b - Reserved

Yuntu Microelectronics

29.2.1.3 lpTMR DIE Register

Offset: 8h

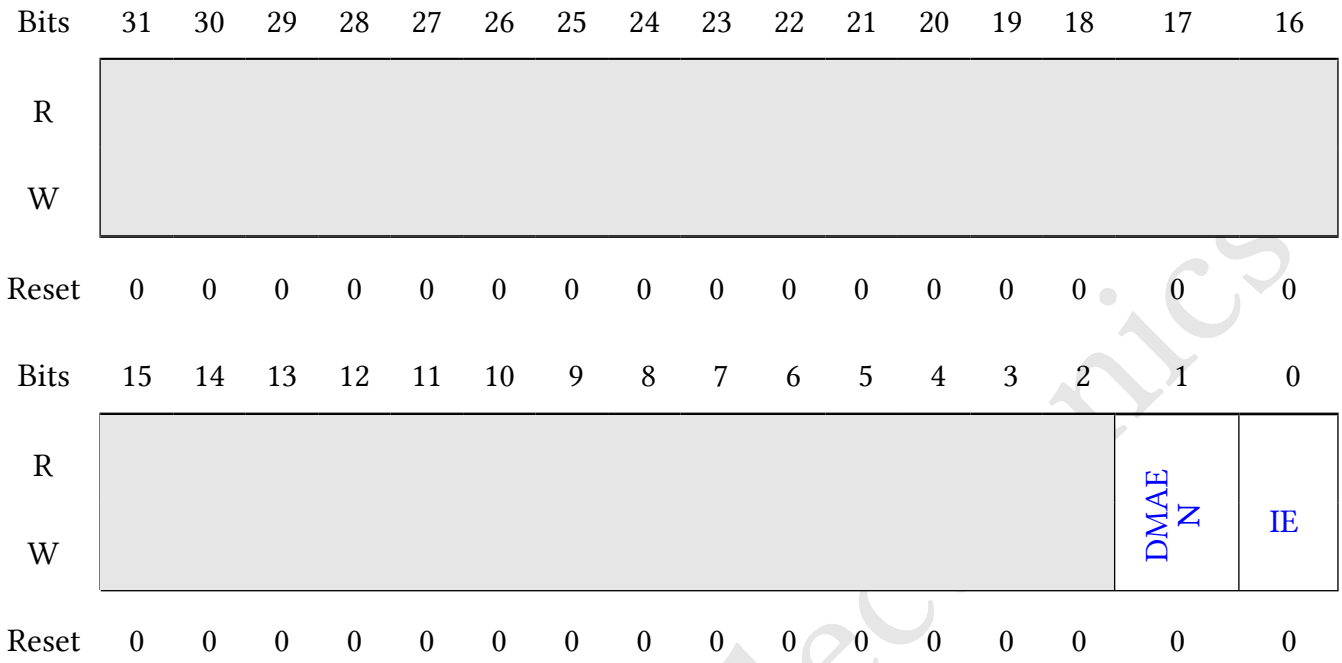


Table 29.4: lpTMR DIE Register Description

Field	Function
1 DMAEN	**Timer DMA Request Enable** When TDRE is set, the lpTMR DMA Request is generated whenever CCF is also set and the CCF is cleared when the DMA Controller is done. 0b - Timer DMA Request disabled. 1b - Timer DMA Request enabled.
0 IE	**Timer Interrupt Enable** When IE is set, the lpTMR Interrupt is generated whenever CCF is also set. 0b - Timer interrupt disabled. 1b - Timer interrupt enabled.

29.2.1.4 lpTMR STS Register

Offset: ch

Yuntu Microelectronics

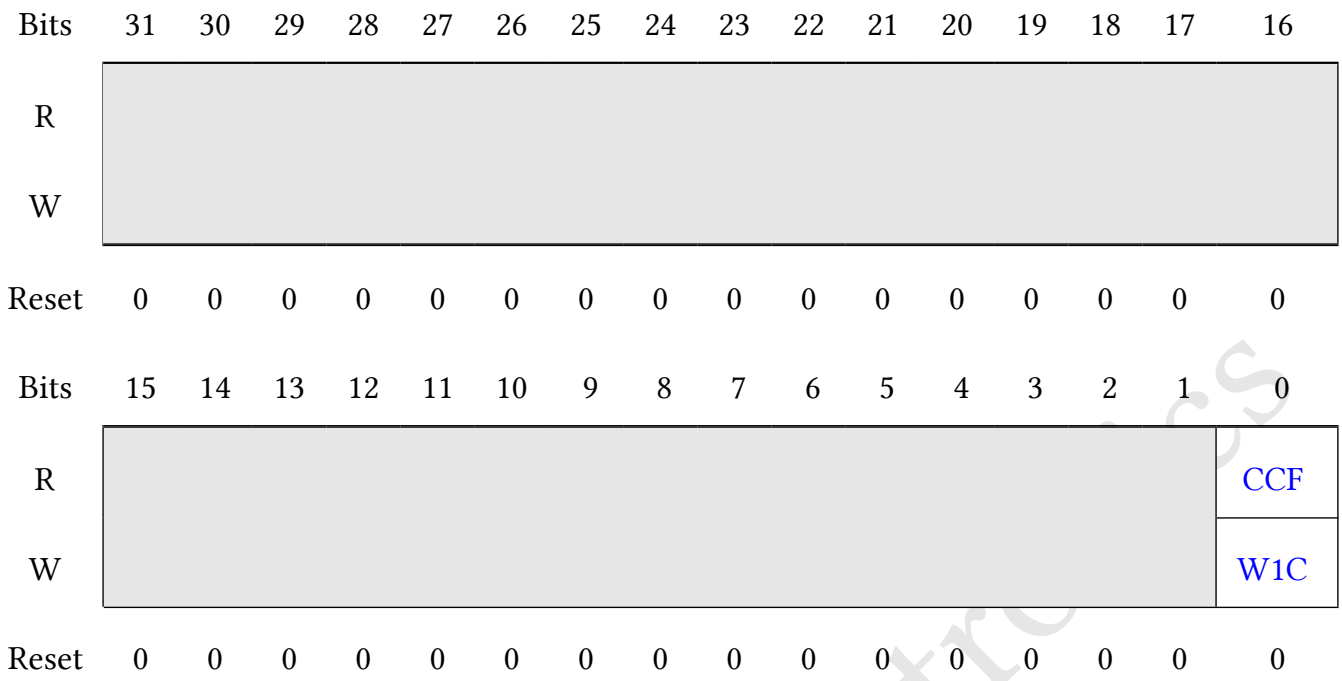


Table 29.5: lpTMR STS Register Description

Field	Function
0 CCF	**Counter Compare Flag** CCF is set when the lpTMR is enabled and the CNT equals the CMP and increments. CCF is cleared when the lpTMR is disabled or writing 1 to it. 0b - The value of CNT is not equal to CMP and increments. 1b - The value of CNT is equal to CMP and increments.

Yuntu Microelectronics

29.2.1.5 lpTMR CMP Register

Offset: 10h

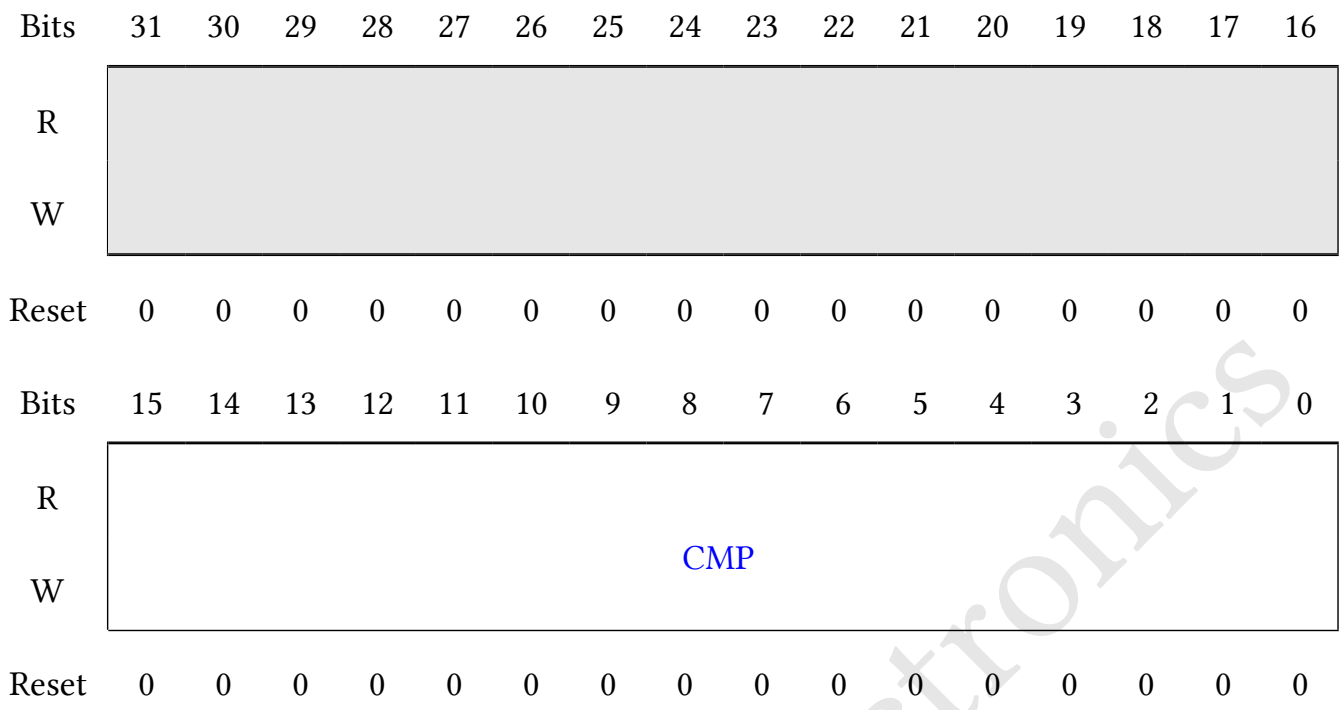


Table 29.6: lpTMR CMP Register Description

Field	Function
15 - 0 CMP	**Compare Value** When the lpTMR is enabled and the CNT equals the value in the CMP and increments, CCF is set and the hardware trigger asserts until the next time the CNT increments. If the CMP is 0, the hardware trigger will remain asserted until the lpTMR is disabled. If the lpTMR is enabled, the CMP must be altered only when CCF is set.

Yuntu Microelectronics

29.2.1.6 lpTMR LCNT Register

Offset: 14h

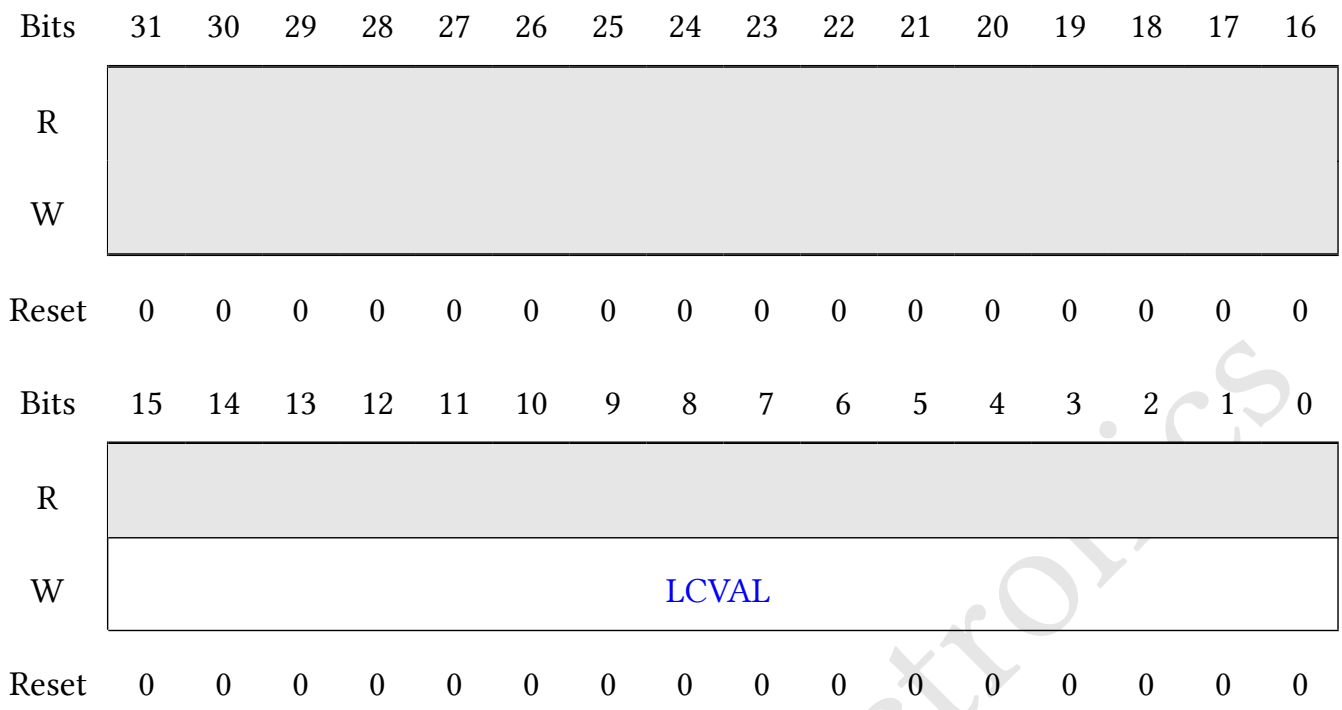


Table 29.7: lpTMR LCNT Register Description

Field	Function
15 - 0 LCVAL	**Latch Timer Counter Value** Write this register to latch timer counter value to CNT register, which is to avoid metastability.

Yuntu Microelectronics

29.2.1.7 lpTMR CNT Register

Offset: 18h

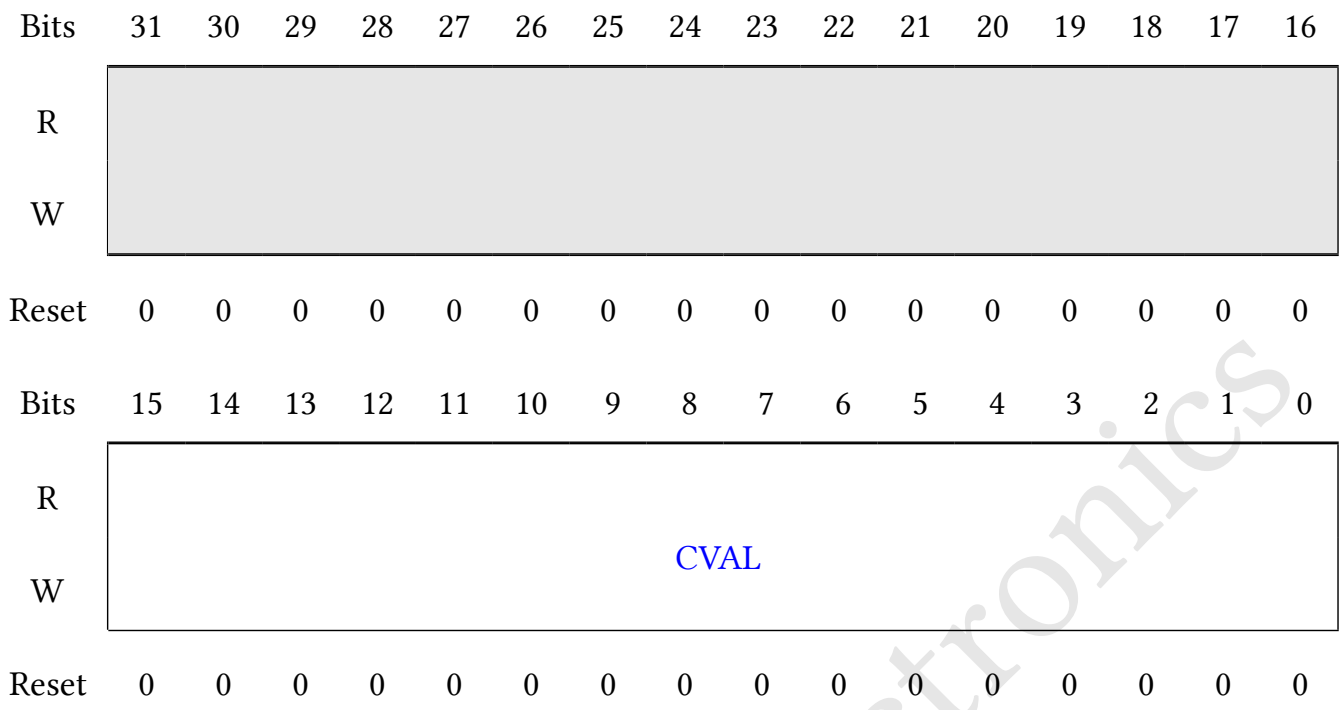


Table 29.8: lpTMR CNT Register Description

Field	Function
15 - 0 CVAL	**Timer Current Counter value** The CNT returns the current value of the lpTMR counter at the time this LCNT register was last written.

29.3 Functional Description

29.3.1 lpTMR Clocking

The lpTMR prescaler/glitch filter can be clocked by one of the clock sources from IPC module. The clock source must be enabled before the lpTMR is enabled.

The clock source selected by IPC may need to be configured to selected the clock source which remains enabled in low-power modes, otherwise the lpTMR will not operate during low-power modes.

In Pulse Counter mode with the prescaler/glitch filter bypassed, the selected input source directly clocks the CNT and no other clock source is required.

29.3.2 lpTMR Prescaler/Glitch Filter

The lpTMR prescaler and glitch filter share the same logic, it can be configured as either prescaler in Time Counter mode or as a glitch filter in Pulse Counter mode.

NOTE. The prescaler/glitch filter configuration must not be changed when the lpTMR is enabled.

Yuntu Microelectronics

29.3.2.1 Time Counter

In Time Counter mode, when the prescaler and lpTMR is enabled, the CNT will increment with every output of prescaler. When prescaler is bypassed and lpTMR is enabled, the selected prescaler clock increments the CNT on every clock cycle.

29.3.2.2 Pulse Counter

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks the CNT. When the lpTMR is first enabled, the output of the glitch filter is asserted, based on the value of CTRL[PINPOL] bit, that is, logic 1 for active-high and logic 0 for active-low.

The lpTMR CNT increments at each time that the glitch filter output asserts in pulse counter mode. The maximum rate for lpTMR CNT increments in pulse mode is 2^2 to 2^{16} prescaler clock edges based on the value of PRS[PRES] register.

In Pulse Counter mode, if the glitch filter is bypassed, the CNT increments on every assertion of the selected input source.

29.3.3 lpTMR Compare

When the CNT equals the value of the CMP and increments, the following events occur:

- lpTMR status bit STS[CCF] is set.
- lpTMR interrupt is generated if DIE[IE] is set.
- lpTMR hardware trigger is generated.
- CNT is reset if CTRL[TMODE] is clear.

When the lpTMR is enabled, the CMP can be changed only when STS[CCF] is set.

29.3.4 lpTMR Counter

The CNT increments by one on every:

- Prescaler clock in Time Counter mode with prescaler bypassed
- Prescaler output in Time Counter mode with prescaler enabled
- Input source assertion in Pulse Counter mode with glitch filter bypassed
- Glitch filter output in Pulse Counter mode with glitch filter enabled

29.3.5 lpTMR Hardware Trigger

The lpTMR hardware trigger asserts at the same time the STS[CCF] is set and can be used to trigger hardware events in other peripherals without software intervention.

- When the CMP is set to 0 with CTRL[TMODE] clear, then the lpTMR hardware trigger will assert on the first compare and does not deassert.
- When the CMP is set to a nonzero value, or, if CTRL[TMODE] is set, then the lpTMR hardware trigger will assert on each compare and deassert on the following increment of the CNT.

29.3.6 lpTMR Interrupt

The lpTMR interrupt is generated whenever DIE[IE] and STS[CCF] are set. STS[CCF] is cleared by disabling the lpTMR or by writing a logic 1 to it. DIE[IE] can be changed and STS[CCF] can be cleared while the lpTMR is enabled.

Yuntu Microelectronics

The lpTMR interrupt is generated asynchronously to the system clock and can be used to generate a wake-up signal from any low power mode, provided the lpTMR is enabled as a wake-up source.

29.3.7 Operation Modes

lpTMR supports the chip operation modes of operation as follows:

- Active: The lpTMR operates normally.
- Sleep: The lpTMR continues to operate normally and can be configured to exit the low power mode by generating an interrupt request.
- Deepsleep: The lpTMR continues to operate normally and can be configured to exit the low power mode by generating an interrupt request.
- Standby: The lpTMR continues to operate normally and can be configured to exit the low power mode by generating an interrupt request.
- Powerdown: The lpTMR continues to operate normally and can be configured to exit the low power mode as a wake-up source configured by WKU module.
- Debug: The lpTMR operates normally in Pulse Counter mode, but the counter does not increment in Time Counter mode.

29.4 Glossary

CMP	Comparator
CNT	Counter

Yuntu Microelectronics

Chapter 30

Enhanced Timer (eTMR)

30.1 Chip-specific eTMR Information

30.1.1 Instantiation Information

YTM32B1HA0x contains 6 eTMR instances. The following table lists the eTMR instances and features.

Table 30.1: eTMR Instances and features

Instances	Channels	Fault inputs	counter width	eTMR modulation	Quad Decoder	Hall sensor support	ACMP input support
eTMR0	8	4	16	Yes	No	Yes	No
eTMR1	8	4	16	No	Yes	Yes	Yes
eTMR2	8	4	16	No	Yes	Yes	Yes
eTMR3	8	4	32	Yes	No	Yes	No
eTMR4	8	4	16	No	No	Yes	No
eTMR5	8	4	16	No	No	Yes	No

30.1.2 Signal Description

Table 30.2: eTMR Signal Table

Signal Name	Description	Direction
TCLK_IN0	Optional external clock source 0 for eTMR counter	I
TCLK_IN1	Optional external clock source 1 for eTMR counter	I
TCLK_IN2	Optional external clock source 2 for eTMR counter	I
eTMR_CH0	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_CH1	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_CH2	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO

Yuntu Microelectronics

Table 30.2 continued from previous page

Signal Name	Description	Direction
eTMR_CH3	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_CH4	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_CH5	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_CH6	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_CH7	Input signal used for input capture mode. Output signal used for PWM mode and output compare mode.	IO
eTMR_FLT0	Used for fault event detection.	I
eTMR_FLT1	Used for fault event detection.	I
eTMR_FLT2	Used for fault event detection.	I
eTMR_FLT3	Used for fault event detection.	I
eTRM_QD_PHA	Phase A used for quadrature decoder.	I
eTMR_QD_PHB	Phase B used for quadrature decoder.	I

30.1.3 Clock Description

Table 30.3: eTMR Clock Table

Clock Name	Description
Bus clock	Bus clock used for register access. Configurable for eTMR counter.
External clock	Configurable for eTMR counter sync to bus clock, from PIN or function clock.

30.1.4 DMA Description

Table 30.4: eTMR DMA Table

Instances	No.	Request Source	Description
eTMR0	36	Compare event or capture event	'OR' of channels compare event or capture event generates request source
eTMR1	20-27	Compare event or capture event	channel 0-7 compare event or capture event generates request source
eTMR2	28-35	Compare event or capture event	channel 0-7 compare event or capture event generates request source
eTMR3	37	Compare event or capture event	'OR' of channels compare event or capture event generates request source
eTMR4	38	Compare event or capture event	'OR' of channels compare event or capture event generates request source

Yuntu Microelectronics

Table 30.4 continued from previous page

Instances	No.	Request Source	Description
eTMR5	39	Compare event or capture event	'OR' of channels compare event or capture event generates request source

30.1.5 Interrupt Description

Table 30.5: eTMR Interrupt Table

Instances	Vector	IRQn	Request Source	Description
eTMR0	129	113	Channel 0/1 interrupt	Channel 0/1 compare event, capture event supports interrupt generation
eTMR0	130	114	Channel 2/3 interrupt	Channel 2/3 compare event, capture event supports interrupt generation
eTMR0	131	115	Channel 4/5 interrupt	Channel 4/5 compare event, capture event supports interrupt generation
eTMR0	132	116	Channel 6/7 interrupt	Channel 6/7 compare event, capture event supports interrupt generation
eTMR0	133	117	Fault interrupt	Fault event supports interrupt generation
eTMR0	134	118	Counter overflow interrupt	eTMR counter overflow event supports interrupt generation
eTMR1	135	119	Channel 0/1 interrupt	Channel 0/1 compare event, capture event supports interrupt generation
eTMR1	136	120	Channel 2/3 interrupt	Channel 2/3 compare event, capture event supports interrupt generation
eTMR1	137	121	Channel 4/5 interrupt	Channel 4/5 compare event, capture event supports interrupt generation
eTMR1	138	122	Channel 6/7 interrupt	Channel 6/7 compare event, capture event supports interrupt generation
eTMR1	139	123	Fault interrupt	Fault event supports interrupt generation
eTMR1	140	124	Counter overflow interrupt	eTMR counter or QD counter overflow event supports interrupt generation
eTMR2	141	125	Channel 0/1 interrupt	Channel 0/1 compare event, capture event supports interrupt generation
eTMR2	142	126	Channel 2/3 interrupt	Channel 2/3 compare event, capture event supports interrupt generation
eTMR2	143	127	Channel 4/5 interrupt	Channel 4/5 compare event, capture event supports interrupt generation
eTMR2	144	128	Channel 6/7 interrupt	Channel 6/7 compare event, capture event supports interrupt generation
eTMR2	145	129	Fault interrupt	Fault event supports interrupt generation
eTMR2	146	130	Counter overflow interrupt	eTMR counter or QD counter overflow event supports interrupt generation
eTMR3	147	131	Channel 0/1 interrupt	Channel 0/1 compare event, capture event supports interrupt generation
eTMR3	148	132	Channel 2/3 interrupt	Channel 2/3 compare event, capture event supports interrupt generation
eTMR3	149	133	Channel 4/5 interrupt	Channel 4/5 compare event, capture event supports interrupt generation

Yuntu Microelectronics

Table 30.5 continued from previous page

Instances	Vector	IRQn	Request Source	Description
eTMR3	150	134	Channel 6/7 interrupt	Channel 6/7 compare event, capture event supports interrupt generation
eTMR3	151	135	Fault interrupt	Fault event supports interrupt generation
eTMR3	152	136	Counter overflow interrupt	eTMR counter overflow event supports interrupt generation
eTMR4	153	137	Channel 0/1 interrupt	Channel 0/1 compare event, capture event supports interrupt generation
eTMR4	154	138	Channel 2/3 interrupt	Channel 2/3 compare event, capture event supports interrupt generation
eTMR4	155	139	Channel 4/5 interrupt	Channel 4/5 compare event, capture event supports interrupt generation
eTMR4	156	140	Channel 6/7 interrupt	Channel 6/7 compare event, capture event supports interrupt generation
eTMR4	157	141	Fault interrupt	Fault event supports interrupt generation
eTMR4	158	142	Counter overflow interrupt	eTMR counter overflow event supports interrupt generation
eTMR5	159	143	Channel 0/1 interrupt	Channel 0/1 compare event, capture event supports interrupt generation
eTMR5	160	144	Channel 2/3 interrupt	Channel 2/3 compare event, capture event supports interrupt generation
eTMR5	161	145	Channel 4/5 interrupt	Channel 4/5 compare event, capture event supports interrupt generation
eTMR5	162	146	Channel 6/7 interrupt	Channel 6/7 compare event, capture event supports interrupt generation
eTMR5	163	147	Fault interrupt	Fault event supports interrupt generation
eTMR5	164	148	Counter overflow interrupt	eTMR counter overflow event supports interrupt generation

30.1.6 Power mode Description

Table 30.6: eTMR Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Keeps working
Deepsleep	Disable
Standby	Disable
Powerdown	Disable

30.2 Introduction

30.2.1 Overview

The eTMR contains a variety of functions, such as PWM output, input capture, output compare, and so on. There is little difference between the functions of each eTMR except for some special functions. The eTMR module works under the bus clock, it's disabled when the bus clock disappears except for the fault

Yuntu Microelectronics

function.

30.2.2 Features

This chip contains 6 eTMRs (eTMR0, eTMR1, eTMR2, eTMR3, eTMR4, eTMR5), their features are divided into common features and individual features.

30.2.2.1 Common features

- Configurable initial and final counter values
- Contain 8 channels
- Support two clock sources
  - Bus clock
  - External clock
- Support 7-bits clock prescaler
- Support four channel modes
  - Common timer
  - PWM mode
    - Independent mode for each channel
    - Complementary mode for each pair of channels
      - All channels support independent deadtime insertion
    - Support dithering
    - Channel output control (initialization, software control, mask control, double switch control, fault control)
      - Support 4 fault input sources
      - Support fault input from TMU or pad
      - Support fault input polarity control
      - Support fault input filter
      - Support fault input stretch
      - Support fault event generated by combinational logic
    - Relevant registers have buffer registers and support loading mechanism
  - Output Compare mode
    - The output can be configured to set, clear or toggle on match point
  - Input Capture mode
    - Support rising edges, falling edges or dual edges capture
    - Support input filter with a prescaler
    - Support capture test mode
    - Support pulse width measure
- Support generating triggers
  - Output triggers with adjustable pulse width on match point
  - Output pulse with adjustable width by PWM
- Polarity control is available for each channel
- Support GTB (Global Time Base)
- Support several interrupts
  - Channel interrupt (capture interrupt and compare interrupt)
  - Counter overflow interrupt
  - Fault event interrupt
- Support DMA
- Support counter running under debug mode

Yuntu Microelectronics

- support input from ACMP
- support hall sensor input

30.2.2.2 Individual features

- Counter
  - eTMR3 has a 32-bit counter
  - Other eTMRs have a 16-bit counter
- eTMR1 and eTMR2 support quadrature decoder mode
  - Contain a independent 16-bit counter with a clock prescaler
  - Support 4 up-down counting modes
  - Support phase A and phase B input filter
  - Support quadrature decoder counter overflow interrupt
- eTMR0 and eTMR3 support modulated output

Yuntu Microelectronics

30.2.3 Block diagram

The following figure shows the eTMR block diagram.

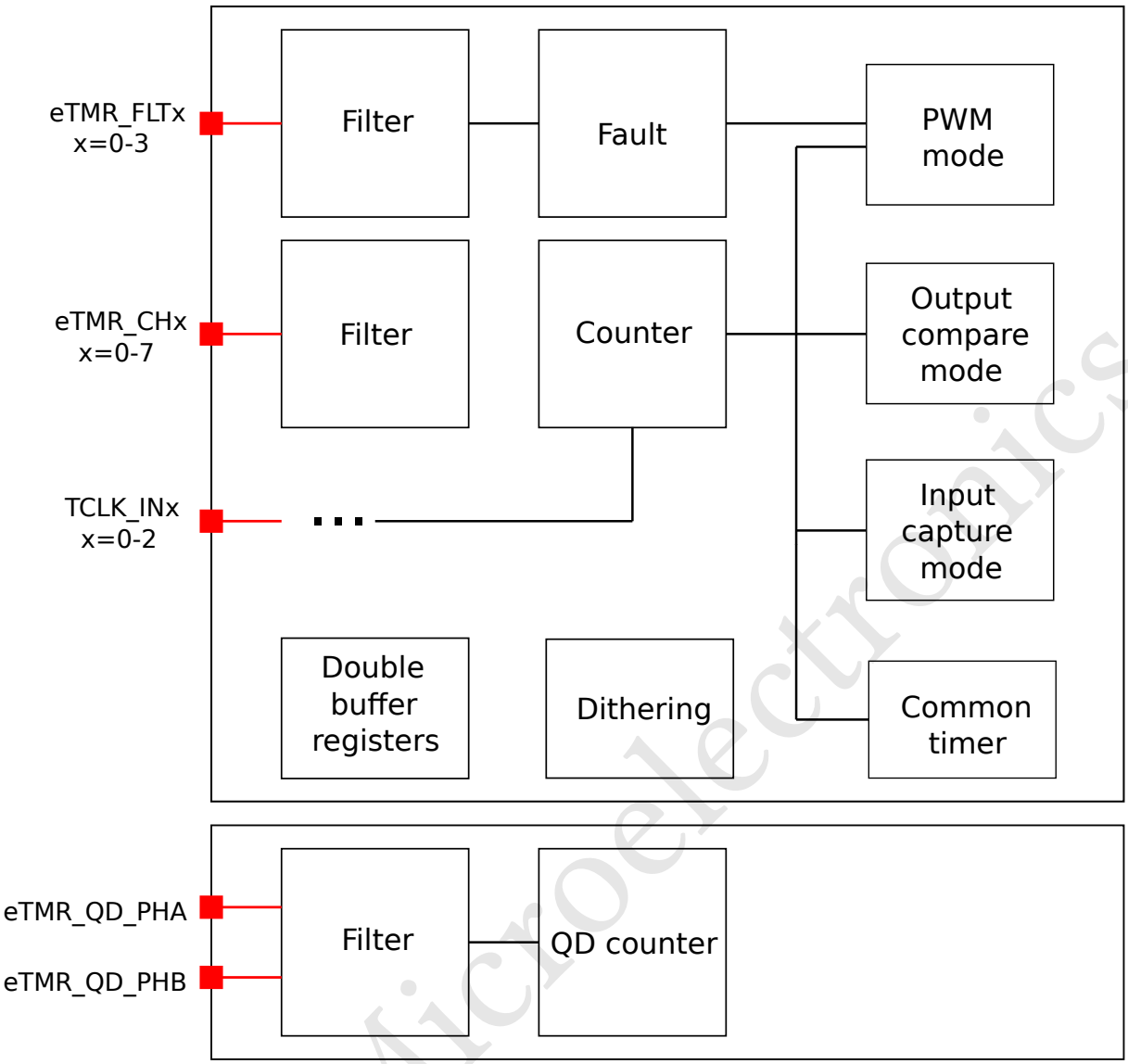


Figure 30.1: eTMR Block Diagram

30.3 Register Definition

NOTE. Quadrature Decoder mode associated register bits are only valid in eTMR1 & eTMR2

NOTE. eTMR3 MOD is 32-bit register.

Reading reserved registers is always 0, the write operation is not valid.

Register_address = Base_address + Address_offset.

30.3.1 eTMR Memory Map

eTMR0 base address 40124000h
eTMR1 base address 40128000h
eTMR2 base address 4012c000h
eTMR3 base address 40130000h
eTMR4 base address 40134000h
eTMR5 base address 40138000h

Yuntu Microelectronics

Table 30.7: eTMR Memory Map

Offset	Register	Width	Access	Reset Value
0h	Status Register (STS)	32	RW	00000000h
4h	Input/Output Status Register (IOSTS)	32	RO	00000000h
8h	Counter Value Register (CNT)	32	RW	00000000h
ch	Quadrature Decoder Counter Value Register (QDCNT)	32	RO	00000000h
10h	Control Register (CTRL)	32	RW	00000000h
14h	Synchronous Register (SYNC)	32	RW	00000000h
18h	Output Trigger Register (OTRIG)	32	RW	00000000h
1ch	Interrupt Enable Register (INTE)	32	RW	00000000h
20h	Channel Output Initial Value Register (CHINIT)	32	RW	00000000h
24h	Channel Software Value Register (CHSWV)	32	RW	00000000h
28h	Channel Output Mask Value Register (CHMASK)	32	RW	00000000h
2ch	Channel Fault Value Register (CHFV)	32	RW	00000000h
30h	Counter Initial Value Register (INIT)	32	RW	00000000h
34h	Counter Modulus Value Register (MOD)	32	RW	see description
38h	Dithering MOD Register (DITHMOD)	32	RW	00000000h
3ch	Counter Middle Value Register (MID)	32	RW	see description
40h	Quadrature Decoder Control Register (QDCTRL)	32	RW	00000000h
44h	Quadrature Decoder Counter Initial Value Register (QDINIT)	32	RW	00000000h
48h	Quadrature Decoder Modulus Register (QDMOD)	32	RW	see description
4ch	Fault Control Register (FAULT)	32	RW	00000000h
80h	Channel Control Register (CH0_CTRL)	32	RW	00000000h
84h	Value 0 Register (CH0_VAL0)	32	RW	00000000h
88h	Value 1 Register (CH0_VAL1)	32	RW	00000000h
8ch	Dithering Value 1 Register (CH0_DITHVAL1)	32	RW	00000000h
90h	Deadtime Register (CH0_DT)	32	RW	00000000h
94h	Capture Value Register (CH0_CVAL)	32	RO	00000000h
98h	Positive Pulse Timeout Flag Number Register (CH0_PPCNTN)	32	RO	00000000h
9ch	Positive Pulse Width Capture Value Register (CH0_PPCNTV)	32	RO	00000000h
a0h	Negative Pulse Timeout Flag Number Register (CH0_NPCNTN)	32	RO	00000000h
a4h	Negative Pulse Width Capture Value Register (CH0_NPCNTV)	32	RO	00000000h
c0h	Channel Control Register (CH1_CTRL)	32	RW	00000000h
c4h	Value 0 Register (CH1_VAL0)	32	RW	00000000h
c8h	Value 1 Register (CH1_VAL1)	32	RW	00000000h
cch	Dithering Value 1 Register (CH1_DITHVAL1)	32	RW	00000000h
d0h	Deadtime Register (CH1_DT)	32	RW	00000000h
d4h	Capture Value Register (CH1_CVAL)	32	RO	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
d8h	Positive Pulse Timeout Flag Number Register (CH1_PPCNTN)	32	RO	00000000h
dch	Positive Pulse Width Capture Value Register (CH1_PPCNTV)	32	RO	00000000h
e0h	Negative Pulse Timeout Flag Number Register (CH1_NPCNTN)	32	RO	00000000h
e4h	Negative Pulse Width Capture Value Register (CH1_NPCNTV)	32	RO	00000000h
100h	Channel Control Register (CH2_CTRL)	32	RW	00000000h
104h	Value 0 Register (CH2_VAL0)	32	RW	00000000h
108h	Value 1 Register (CH2_VAL1)	32	RW	00000000h
10ch	Dithering Value 1 Register (CH2_DITHVAL1)	32	RW	00000000h
110h	Deadtime Register (CH2_DT)	32	RW	00000000h
114h	Capture Value Register (CH2_CVAL)	32	RO	00000000h
118h	Positive Pulse Timeout Flag Number Register (CH2_PPCNTN)	32	RO	00000000h
11ch	Positive Pulse Width Capture Value Register (CH2_PPCNTV)	32	RO	00000000h
120h	Negative Pulse Timeout Flag Number Register (CH2_NPCNTN)	32	RO	00000000h
124h	Negative Pulse Width Capture Value Register (CH2_NPCNTV)	32	RO	00000000h
140h	Channel Control Register (CH3_CTRL)	32	RW	00000000h
144h	Value 0 Register (CH3_VAL0)	32	RW	00000000h
148h	Value 1 Register (CH3_VAL1)	32	RW	00000000h
14ch	Dithering Value 1 Register (CH3_DITHVAL1)	32	RW	00000000h
150h	Deadtime Register (CH3_DT)	32	RW	00000000h
154h	Capture Value Register (CH3_CVAL)	32	RO	00000000h
158h	Positive Pulse Timeout Flag Number Register (CH3_PPCNTN)	32	RO	00000000h
15ch	Positive Pulse Width Capture Value Register (CH3_PPCNTV)	32	RO	00000000h
160h	Negative Pulse Timeout Flag Number Register (CH3_NPCNTN)	32	RO	00000000h
164h	Negative Pulse Width Capture Value Register (CH3_NPCNTV)	32	RO	00000000h
180h	Channel Control Register (CH4_CTRL)	32	RW	00000000h
184h	Value 0 Register (CH4_VAL0)	32	RW	00000000h
188h	Value 1 Register (CH4_VAL1)	32	RW	00000000h
18ch	Dithering Value 1 Register (CH4_DITHVAL1)	32	RW	00000000h
190h	Deadtime Register (CH4_DT)	32	RW	00000000h
194h	Capture Value Register (CH4_CVAL)	32	RO	00000000h
198h	Positive Pulse Timeout Flag Number Register (CH4_PPCNTN)	32	RO	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
19ch	Positive Pulse Width Capture Value Register (CH4_PPCNTV)	32	RO	00000000h
1a0h	Negative Pulse Timeout Flag Number Register (CH4_NPCNTN)	32	RO	00000000h
1a4h	Negative Pulse Width Capture Value Register (CH4_NPCNTV)	32	RO	00000000h
1c0h	Channel Control Register (CH5_CTRL)	32	RW	00000000h
1c4h	Value 0 Register (CH5_VAL0)	32	RW	00000000h
1c8h	Value 1 Register (CH5_VAL1)	32	RW	00000000h
1cch	Dithering Value 1 Register (CH5_DITHVAL1)	32	RW	00000000h
1d0h	Deadtime Register (CH5_DT)	32	RW	00000000h
1d4h	Capture Value Register (CH5_CVAL)	32	RO	00000000h
1d8h	Positive Pulse Timeout Flag Number Register (CH5_PPCNTN)	32	RO	00000000h
1dch	Positive Pulse Width Capture Value Register (CH5_PPCNTV)	32	RO	00000000h
1e0h	Negative Pulse Timeout Flag Number Register (CH5_NPCNTN)	32	RO	00000000h
1e4h	Negative Pulse Width Capture Value Register (CH5_NPCNTV)	32	RO	00000000h
200h	Channel Control Register (CH6_CTRL)	32	RW	00000000h
204h	Value 0 Register (CH6_VAL0)	32	RW	00000000h
208h	Value 1 Register (CH6_VAL1)	32	RW	00000000h
20ch	Dithering Value 1 Register (CH6_DITHVAL1)	32	RW	00000000h
210h	Deadtime Register (CH6_DT)	32	RW	00000000h
214h	Capture Value Register (CH6_CVAL)	32	RO	00000000h
218h	Positive Pulse Timeout Flag Number Register (CH6_PPCNTN)	32	RO	00000000h
21ch	Positive Pulse Width Capture Value Register (CH6_PPCNTV)	32	RO	00000000h
220h	Negative Pulse Timeout Flag Number Register (CH6_NPCNTN)	32	RO	00000000h
224h	Negative Pulse Width Capture Value Register (CH6_NPCNTV)	32	RO	00000000h
240h	Channel Control Register (CH7_CTRL)	32	RW	00000000h
244h	Value 0 Register (CH7_VAL0)	32	RW	00000000h
248h	Value 1 Register (CH7_VAL1)	32	RW	00000000h
24ch	Dithering Value 1 Register (CH7_DITHVAL1)	32	RW	00000000h
250h	Deadtime Register (CH7_DT)	32	RW	00000000h
254h	Capture Value Register (CH7_CVAL)	32	RO	00000000h
258h	Positive Pulse Timeout Flag Number Register (CH7_PPCNTN)	32	RO	00000000h
25ch	Positive Pulse Width Capture Value Register (CH7_PPCNTV)	32	RO	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
260h	Negative Pulse Timeout Flag Number Register (CH7_NPCNTN)	32	RO	00000000h
264h	Negative Pulse Width Capture Value Register (CH7_NPCNTV)	32	RO	00000000h

30.3.1.1 eTMR STS Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CH7NPF	CH6NPF	CH5NPF	CH4NPF	CH3NPF	CH2NPF	CH1NPF	CH0NPF	CH7PPF	CH6PPF	CH5PPF	CH4PPF	CH3PPF	CH2PPF	CH1PPF	CH0PPF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	QDDIR	QDTOF	TOF	RF	F3F	F2F	F1F	F0F	CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W		W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.8: eTMR STS Register Description

Field	Function
31 CH7NPF	**Channel 7 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
30 CH6NPF	**Channel 6 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
29 CH5NPF	**Channel 5 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
28 CH4NPF	**Channel 4 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
27 CH3NPF	**Channel 3 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
26 CH2NPF	**Channel 2 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
25 CH1NPF	**Channel 1 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.

Yuntu Microelectronics

Field	Function
24 CH0NPF	**Channel 0 Negative Pulse Ready Flag** Capture negative pulse ready flag, write 1 to clear this bit.
23 CH7PPF	**Channel 7 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
22 CH6PPF	**Channel 6 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
21 CH5PPF	**Channel 5 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
20 CH4PPF	**Channel 4 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
19 CH3PPF	**Channel 3 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
18 CH2PPF	**Channel 2 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
17 CH1PPF	**Channel 1 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
16 CH0PPF	**Channel 0 Positive Pulse Ready Flag** Capture positive pulse ready flag, write 1 to clear this bit.
15 QDDIR	**Quadrature Decoder Direction** Indicate quadrature decoder counter direction. 0b - Quadrature counter increase. 1b - Quadrature counter decrease.
14 QDTOF	**Quadrature Decoder Timer Overflow Flag** When STS[QDTOF]=1, STS[QDDIR]=0, means quadrature decoder counter increase to QDMOD[QDMOD], then overflow to QDINIT[QDINIT]. When STS[QDTOF]=1, STS[QDDIR]=1, means quadrature decoder counter decrease to QDINIT[QDINIT], then overflow to QDMOD[QDMOD]. Write 1 to clear the bit.
13 TOF	**Timer Overflow Flag** Timer overflow flag sets when counter increase from MOD[MOD] to INIT[INIT]. Write 1 to clear the bit.
12 RF	**Reload Flag** Reload flag sets when registers loading occurs. write 1 to clear the bit.

Yuntu Microelectronics

Field	Function
11 F3F	**Fault 3 Flag** Fault event 3 occurs flag. Write 1 to clear the bit.
10 F2F	**Fault 2 Flag** Fault event 2 occurs flag. Write 1 to clear the bit.
9 F1F	**Fault 1 Flag** Fault event 1 occurs flag. Write 1 to clear the bit.
8 F0F	**Fault 0 Flag** Fault event 0 occurs flag. Write 1 to clear the bit.
7 CH7F	**Channel 7 Flag** Channel 7 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
6 CH6F	**Channel 6 Flag** Channel 6 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
5 CH5F	**Channel 5 Flag** Channel 5 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
4 CH4F	**Channel 4 Flag** Channel 4 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
3 CH3F	**Channel 3 Flag** Channel 3 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
2 CH2F	**Channel 2 Flag** Channel 2 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
1	**Channel 1 Flag**

Yuntu Microelectronics

Field	Function
CH1F	Channel 1 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.
0 CH0F	**Channel 0 Flag** Channel 0 flag is used for capture mode and compare mode. In capture mode, this flag sets when capture event occurs. In compare mode, this flag sets when counter matches the values. Write 1 to clear the bit.

30.3.1.2 eTMR IOSTS Register

Offset: 4h

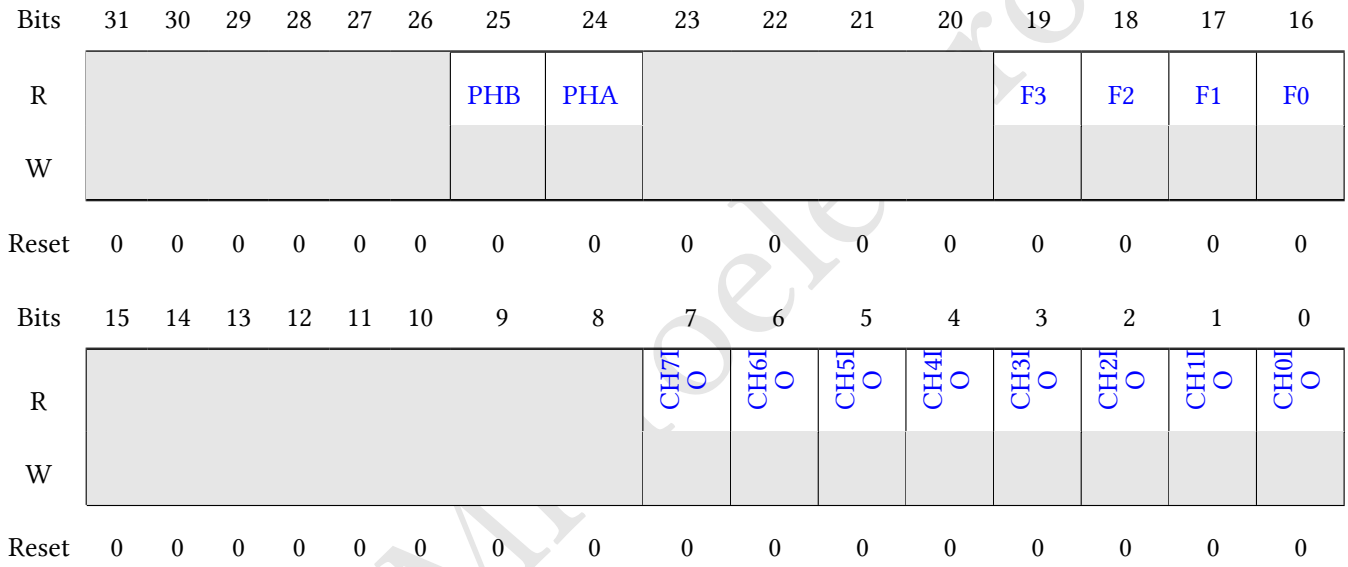


Table 30.9: eTMR IOSTS Register Description

Field	Function
25 PHB	**Phase B Status** Phase B input status is used for Quadrature Decoder. Read-only.
24 PHA	**Phase A Status** Phase A input status is used for Quadrature Decoder. Read-only.
19 F3	**Fault 3 Status** Fault 3 input status is used for fault logic. Read-only.
18 F2	**Fault 2 Status** Fault 2 input status is used for fault logic. Read-only.
17	**Fault 1 Status**

Yuntu Microelectronics

Field	Function
F1	Fault 1 input status is used for fault logic. Read-only.
16 F0	**Fault 0 Status** Fault 0 input status is used for fault logic. Read-only.
7 CH7IO	**Channel 7 I/O Status** In capture mode, indicate channel 7 input status. In compare mode or PWM mode, indicate channel 7 output status. Read-only.
6 CH6IO	**Channel 6 I/O Status** In capture mode, indicate channel 6 input status. In compare mode or PWM mode, indicate channel 6 output status. Read-only.
5 CH5IO	**Channel 5 I/O Status** In capture mode, indicate channel 5 input status. In compare mode or PWM mode, indicate channel 5 output status. Read-only.
4 CH4IO	**Channel 4 I/O Status** In capture mode, indicate channel 4 input status. In compare mode or PWM mode, indicate channel 4 output status. Read-only.
3 CH3IO	**Channel 3 I/O Status** In capture mode, indicate channel 3 input status. In compare mode or PWM mode, indicate channel 3 output status. Read-only.
2 CH2IO	**Channel 2 I/O Status** In capture mode, indicate channel 2 input status. In compare mode or PWM mode, indicate channel 2 output status. Read-only.
1 CH1IO	**Channel 1 I/O Status** In capture mode, indicate channel 1 input status. In compare mode or PWM mode, indicate channel 1 output status. Read-only.
0 CH0IO	**Channel 0 I/O Status** In capture mode, indicate channel 0 input status. In compare mode or PWM mode, indicate channel 0 output status. Read-only.

Yuntu Microelectronics

30.3.1.3 eTMR CNT Register

This register is 32-bit valid in eTMR3, other eTMRs are 16-bit valid.

Offset: 8h

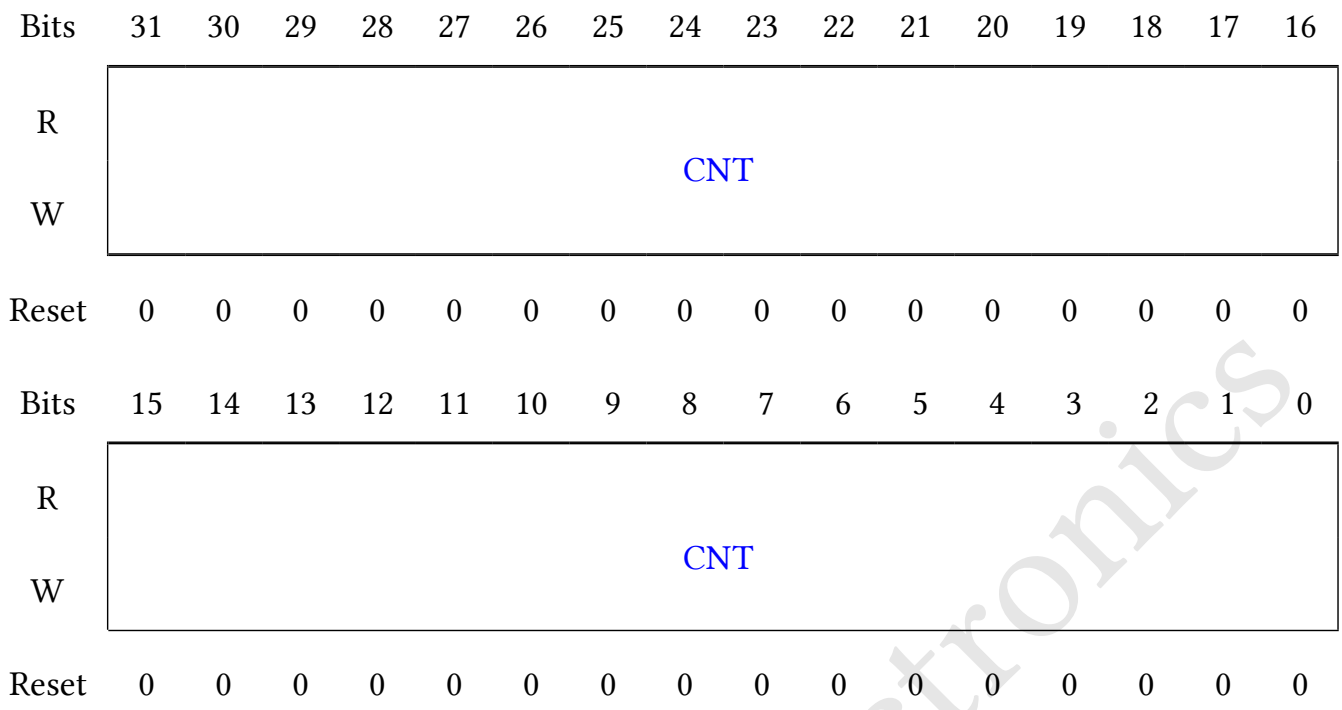


Table 30.10: eTMR CNT Register Description

Field	Function
31 - 0 CNT	**Counter Initial or Value Register** When CTRL[EN]=0, writing to this register initializes counter according to SYNC[CNTINIT] configuration. When CTRL[EN]=1, reading this register gets the counter value, and writing to this register reloads value from INIT.

Yuntu Microelectronics

30.3.1.4 eTMR QDCNT Register

Offset: ch

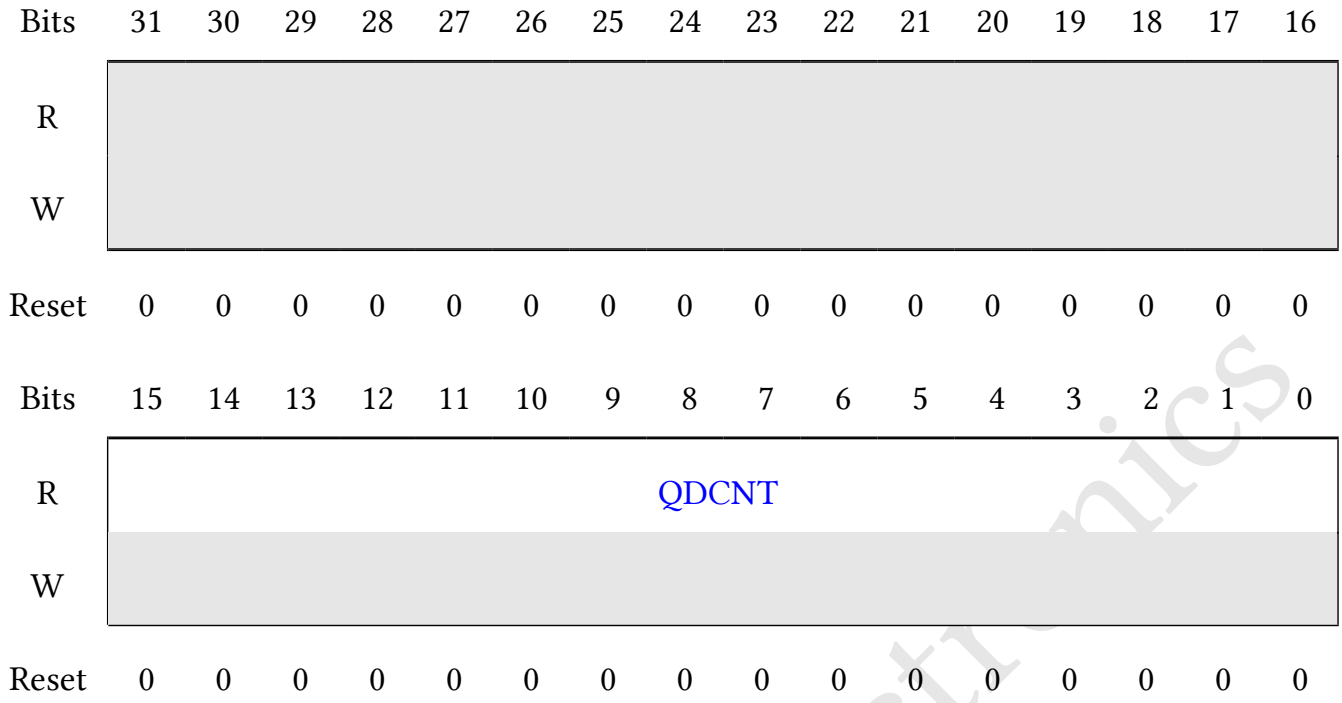
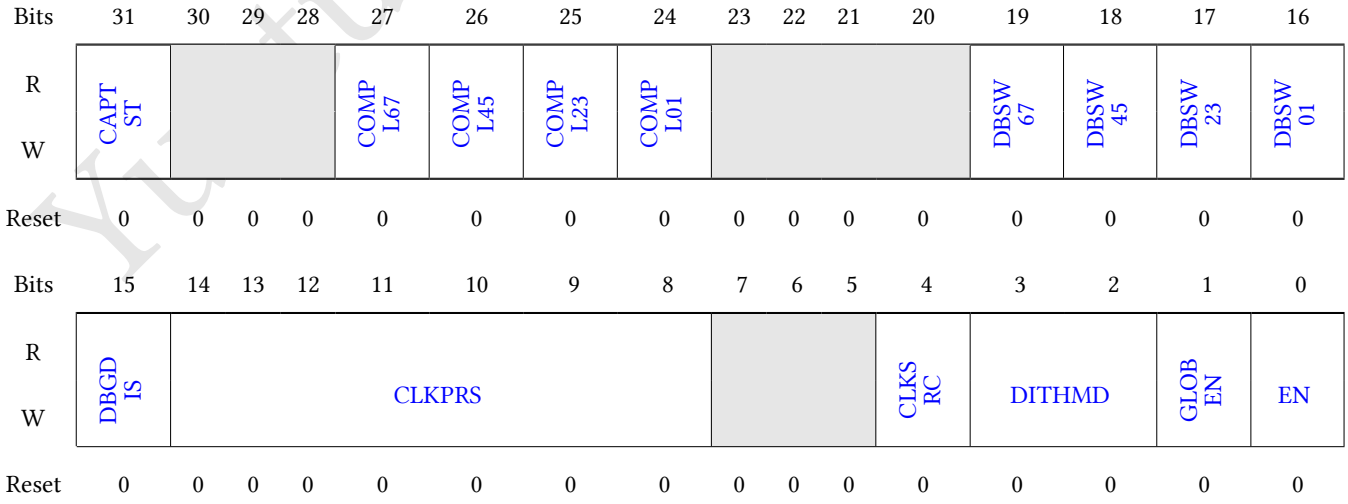


Table 30.11: eTMR QDCNT Register Description

Field	Function
15 - 0 QDCNT	**Quadrature Decoder Counter Value** Reading this register gets quadrature decoder counter value. The write operation is not valid.

30.3.1.5 eTMR CTRL Register

Offset: 10h



Yuntu Microelectronics

Table 30.12: eTMR CTRL Register Description

Field	Function
31 CAPTST	**Capture Test** This bit is used for the capture test. 0b - Capture test is disabled. 1b - Capture test is enabled.
27 COMPL67	**Complementary Mode enable for channel 6 and channel 7** Channel 6 and 7 output independently or complementary in PWM mode. 0b - Independent mode. 1b - Complementary mode.
26 COMPL45	**Complementary Mode enable for channel 4 and channel 5** Channel 4 and 5 output independently or complementary in PWM mode. 0b - Independent mode. 1b - Complementary mode.
25 COMPL23	**Complementary Mode enable for channel 2 and channel 3** Channel 2 and 3 output independently or complementary in PWM mode. 0b - Independent mode. 1b - Complementary mode.
24 COMPL01	**Complementary Mode enable for channel 0 and channel 1** Channel 0 and 1 output independently or complementary in PWM mode. 0b - Independent mode. 1b - Complementary mode.
19 DBSW67	**Double Switch** Channel 6 and 7 is used to generate the double switch signal. 0b - Double switch is disabled. 1b - Double switch is enabled.
18 DBSW45	**Double Switch** Channel 4 and 5 is used to generate the double switch signal. 0b - Double switch is disabled. 1b - Double switch is enabled.
17 DBSW23	**Double Switch** Channel 2 and 3 is used to generate the double switch signal. 0b - Double switch is disabled. 1b - Double switch is enabled.
16 DBSW01	**Double Switch** Channel 0 and 1 is used to generate the double switch signal. 0b - Double switch is disabled. 1b - Double switch is enabled.
15	**Debug Mode Disabled**

Yuntu Microelectronics

Field	Function
DBGDIS	Enable/disable the debug mode control. 0b - Counter stops under debug mode. 1b - Counter keeps running under debug mode.
14 - 8 CLKPRS	**Clock Prescaler** Clock prescaler for counter, from 1 to 128.
4 CLKSRC	**Clock Source** Clock source for counter, bus clock or external clock. 0b - Select bus clock. 1b - Select external clock.
3 - 2 DITHMD	**Dithering Mode** Select the dithering type in PWM mode. 00b - The dithering mode is disabled. 01b - Select period dithering. 10b - Select edge dithering. 11b - Select period and edge dithering.
1 GLOBEN	**Global Enable** eTMRs counter synchronization mechanisms. 0b - Global logic is disabled. 1b - Global logic is enabled. When SYNC[GLOB]=1 from any eTMR includes itself, the counter starts running. When all eTMRs includes itself SYNC[GLOB]=0, the counter stops.
0 EN	**eTMR Enable** eTMR enable/disable. 0b - The eTMR is disabled. 1b - The eTMR is enabled.

Yuntu Microelectronics

30.3.1.6 eTMR SYNC Register

Offset: 14h

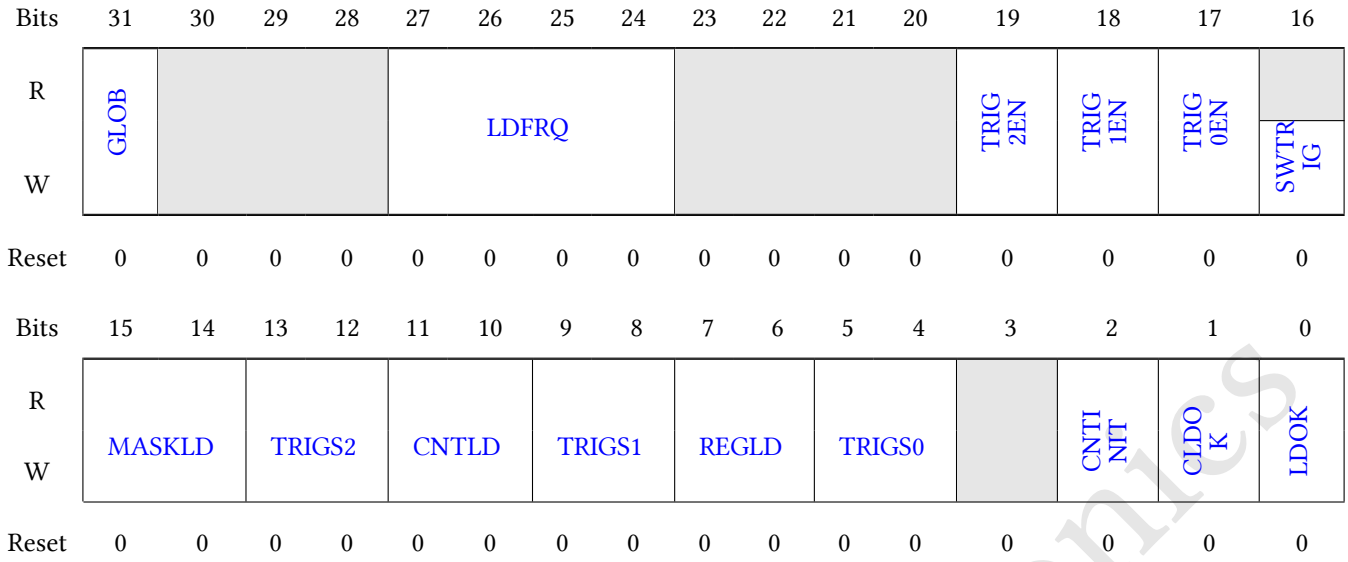


Table 30.13: eTMR SYNC Register Description

Field	Function
31 GLOB	**Global Base** Generate global base signal. This bit is used to synchronize with other eTMRs which CTRL[GLOBEN]=1. 0b - Not generate global base signal. 1b - Generate global base signal.
27 - 24 LDFRQ	**Registers Loading Frequency** Registers Loading occurs at the reloading opportunity that meets this specified frequency.
19 TRIG2EN	**Hardware Trigger 2 Enable** Enable hardware trigger 2. 0b - Hardware trigger 2 is disabled. 1b - Hardware trigger 2 is enabled.
18 TRIG1EN	**Hardware Trigger 1 Enable** Enable hardware trigger 1. 0b - Hardware trigger 1 is disabled. 1b - Hardware trigger 1 is enabled.
17 TRIG0EN	**Hardware Trigger 0 Enable** Enable hardware trigger 0. 0b - Hardware trigger 0 is disabled. 1b - Hardware trigger 0 is enabled.
16 SWTRIG	**Software Trigger** Software trigger is used for relevant registers loading. Writing 1 to this bit generates one bus clock cycle pulse trigger. Write-only.

Yuntu Microelectronics

Field	Function
15 - 14 MASKLD	**CHMASK Loading Opportunity** Select loading opportunity for CHMASK register. 00b - Loading together with registers (INIT, MID, MOD, CH_VAL0 and CH_VAL1). 01b - Loading occurs when counter matches with MOD[MOD]. 10b - Loading occurs when counter matches with MID[MID]. 11b - Loading occurs with trigger event which selected by SYNC[TRIGS2]. Any valid hardware trigger x will generate a trigger event (x=0,1,2).
13 - 12 TRIGS2	**Trigger Source 2** 01b - Select software trigger. 10b - Select hardware trigger. 11b - Software and hardware trigger are both valid.
11 - 10 CNTLD	**Counter Loading Opportunity** Select loading opportunity for counter, the counter loads with value from INIT +1. 00b - Loading together with registers (INIT, MID, MOD, CH_VAL0 and CH_VAL1). 01b - Loading occurs when counter matches with MID[MID]. 10b - Loading occurs with trigger event which selected by SYNC[TRIGS1]. Any valid hardware trigger x will generate a trigger event (x=0,1,2). 11b - Reserved.
9 - 8 TRIGS1	**Trigger Source 1.** Select the loading trigger source for counter. When selecting hardware trigger, any trigger enabled by SYNC[TRIGxEN] configuratgion is as valid trigger source. 00b - Disable trigger. 01b - Select software trigger. 10b - Select hardware trigger. 11b - Software and hardware trigger are both valid.
7 - 6 REGLD	**Registers Loading Opportunity** Select loading opportunity for registers (INIT, MID, MOD, CH_VAL0, CH_VAL1). 00b - Loading is disabled. 01b - Loading occurs when counter matches with MOD[MOD]. 10b - Loading occurs when counter matches with MID[MID]. 11b - Loading occurs with trigger event which selected by SYNC[TRIGS0]. Any valid hardware trigger x will generate a trigger event (x=0,1,2).
5 - 4	**Trigger Source 0.**

Yuntu Microelectronics

Field	Function
TRIGS0	Select the loading trigger source for registers (INIT, MOD, MID, CH_VAL0, CH_VAL1). When selecting hardware trigger, any trigger enabled by SYNC[TRIGxEN] configuratgion is as valid trigger source. 00b - Disable trigger. 01b - Select software trigger. 10b - Select hardware trigger. 11b - Software and hardware trigger are both valid.
2 CNTINIT	**Counter Initial Value Control** Before the eTMR enables, select initial value source for counter. 0b - Counter initial value is from register CNT by writing operation. 1b - Counter initial value is from register INIT.
1 CLDOK	**Clear LDOK** Write 1 to this bit to clear SYNC[LDOK], and it automatically clear when SYNC[LDOK]=0.
0 LDOK	**Registers Loading OK** Set LDOK=1 to indicate registers (INIT, MID, MOD, CH_VAL0, CH_VAL1) loading is ready, it automatically clear when registers loading successfully. 0b - Registers loading is not ready. 1b - Registers loading is ready and waits for loading opportunity.

30.3.1.7 eTMR OTRIG Register

Offset: 18h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W									TWD							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TFRQ					PULSESRC						TSRC		MODTEN	MIDTEN	INITTEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.14: eTMR OTRIG Register Description

Field	Function
23 - 16 TWD	**Output Trigger Width** Trigger output width configurable from 1 to 256 bus clock cycles. Default trigger width is 1 bus clock cycle.
15 - 12 TFRQ	**Trigger Frequency** Trigger output occurs at the opportunity that meets the specified frequency. When OTRIG[TFRQ]=0, each PWM cylce generates one trigger. When OTRIG[TFRQ]=1, each two PWM cylces generates one trigger. When OTRIG[TFRQ]=x, each (x+1) PWM cylces generates one trigger.
10 - 8 PULSESRC	**Pulse-out Source** Select PWM as the pulse-out signal. 000b - Select channel 0 PWM as the pulse-out signal. 001b - Select channel 1 PWM as the pulse-out signal. 010b - Select channel 2 PWM as the pulse-out signal. 011b - Select channel 3 PWM as the pulse-out signal. 100b - Select channel 4 PWM as the pulse-out signal. 101b - Select channel 5 PWM as the pulse-out signal. 110b - Select channel 6 PWM as the pulse-out signal. 111b - select channel 7 PWM as the pulse-out signal.
4 TSRC	**Trigger Source** Trigger output source: matching event occurs or PWM output. 0b - Select trigger that generated by matching event. 1b - Select PWM output as the trigger.
2 MODTEN	**MOD Matching Trigger Enable** Enable/disable the MOD[MOD] matching trigger. 0b - Not generate trigger when counter matches with MOD[MOD]. 1b - Generate trigger when counter matches with MOD[MOD].
1 MIDTEN	**MID Matching Trigger Enable** Enable/disable the MID[MID] matching trigger. 0b - Not generate trigger when counter matches with MID[MID]. 1b - Generate trigger when counter matches with MID[MID].
0 INITTEN	**INIT Matching Trigger Enable** Enable/disable the INIT[INIT] matching trigger. 0b - Not generate trigger when counter matches with INIT[INIT] + 1. 1b - Generate trigger when counter matches with INIT[INIT] + 1.

30.3.1.8 eTMR INTE Register

Offset: 1ch

Yuntu Microelectronics

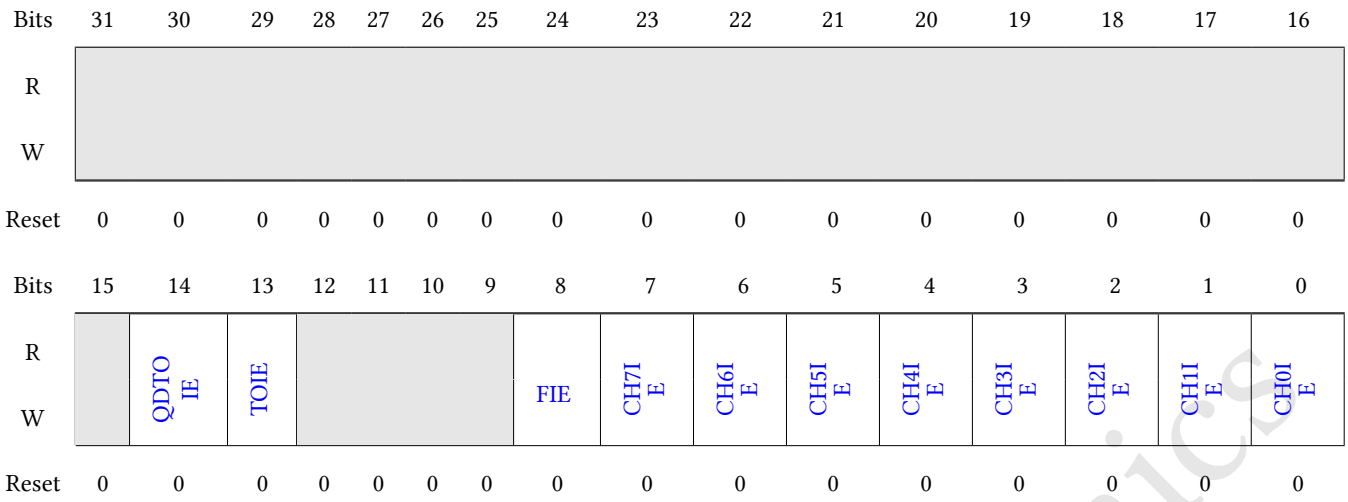


Table 30.15: eTMR INTE Register Description

Field	Function
14 QDTOIE	**Quadrature Decoder Timer Overflow Interrupt Enable** Quadrature decoder timer overflow interrupt enable/disable. 0b - Quadrature decoder timer overflow interrupt is disabled. 1b - Quadrature decoder timer overflow interrupt is enabled.
13 TOIE	**Timer Overflow Interrupt Enable** Timer overflow interrupt enable/disable. 0b - Timer overflow interrupt is disabled. 1b - Timer overflow interrupt is enabled.
8 FIE	**Fault Interrupt Enable** Fault interrupt enable/disable. 0b - Fault interrupt is disabled. 1b - Fault interrupt is enabled.
7 CH7IE	**Channel 7 Interrupt Enable** Channel 7 capture/compare interrupt enable/disable. 0b - Channel 7 interrupt is disabled. 1b - Channel 7 interrupt is enabled.
6 CH6IE	**Channel 6 Interrupt Enable** Channel 6 capture/compare interrupt enable/disable. 0b - Channel 6 interrupt is disabled. 1b - Channel 6 interrupt is enabled.
5 CH5IE	**Channel 5 Interrupt Enable** Channel 5 capture/compare interrupt enable/disable. 0b - Channel 5 interrupt is disabled. 1b - Channel 5 interrupt is enabled.
4	**Channel 4 Interrupt Enable**

Yuntu Microelectronics

Field	Function
CH4IE	Channel 4 capture/compare interrupt enable/disable. 0b - Channel 4 interrupt is disabled. 1b - Channel 4 interrupt is enabled.
3 CH3IE	**Channel 3 Interrupt Enable** Channel 3 capture/compare interrupt enable/disable. 0b - Channel 3 interrupt is disabled. 1b - Channel 3 interrupt is enabled.
2 CH2IE	**Channel 2 Interrupt Enable** Channel 2 capture/compare interrupt enable/disable. 0b - Channel 2 interrupt is disabled. 1b - Channel 2 interrupt is enabled.
1 CH1IE	**Channel 1 Interrupt Enable** Channel 1 capture/compare interrupt enable/disable. 0b - Channel 1 interrupt is disabled. 1b - Channel 1 interrupt is enabled.
0 CH0IE	**Channel 0 Interrupt Enable** Channel 0 capture/compare interrupt enable/disable. 0b - Channel 0 interrupt is disabled. 1b - Channel 0 interrupt is enabled.

30.3.1.9 eTMR CHINIT Register

Offset: 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R									CH7INITV	CH6INITV	CH5INITV	CH4INITV	CH3INITV	CH2INITV	CH1INITV	CH0INITV
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W									CH7INIT	CH6INIT	CH5INIT	CH4INIT	CH3INIT	CH2INIT	CH1INIT	CH0INIT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.16: eTMR CHINIT Register Description

Field	Function
23 CH7INITV	**Channel 7 Output Initial Value** Channel 7 output initial value. 0b - Channel 7 output initial value is 0. 1b - Channel 7 output initial value is 1.
22 CH6INITV	**Channel 6 Output Initial Value** Channel 6 output initial value. 0b - Channel 6 output initial value is 0. 1b - Channel 6 output initial value is 1.
21 CH5INITV	**Channel 5 Output Initial Value** Channel 5 output initial value. 0b - Channel 5 output initial value is 0. 1b - Channel 5 output initial value is 1.
20 CH4INITV	**Channel 4 Output Initial Value** Channel 4 output initial value. 0b - Channel 4 output initial value is 0. 1b - Channel 4 output initial value is 1.
19 CH3INITV	**Channel 3 Output Initial Value** Channel 3 output initial value. 0b - Channel 3 output initial value is 0. 1b - Channel 3 output initial value is 1.
18 CH2INITV	**Channel 2 Output Initial Value** Channel 2 output initial value. 0b - Channel 2 output initial value is 0. 1b - Channel 2 output initial value is 1.
17 CH1INITV	**Channel 1 Output Initial Value** Channel 1 output initial value. 0b - Channel 1 output initial value is 0. 1b - Channel 1 output initial value is 1.
16 CH0INITV	**Channel 0 Output Initial Value** Channel 0 output initial value. 0b - Channel 0 output initial value is 0. 1b - Channel 0 output initial value is 1.
7	**Channel 7 Output Initial**

Yuntu Microelectronics

Field	Function
CH7INIT	Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 7 output not initial. 1b - Channel 7 output initial.
6 CH6INIT	**Channel 6 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 6 output not initial. 1b - Channel 6 output initial.
5 CH5INIT	**Channel 5 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 5 output not initial. 1b - Channel 5 output initial.
4 CH4INIT	**Channel 4 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 4 output not initial. 1b - Channel 4 output initial.
3 CH3INIT	**Channel 3 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 3 output not initial. 1b - Channel 3 output initial.
2 CH2INIT	**Channel 2 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 2 output not initial. 1b - Channel 2 output initial.
1 CH1INIT	**Channel 1 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 1 output not initial. 1b - Channel 1 output initial.
0 CH0INIT	**Channel 0 Output Initial** Write 1 to this bit to initialize channel output with CHINIT[CHxINITV], channel output change with a counter matching event. Write-Only. 0b - Channel 0 output not initial. 1b - Channel 0 output initial.

Yuntu Microelectronics

30.3.1.10 eTMR CHSWV Register

Offset: 24h

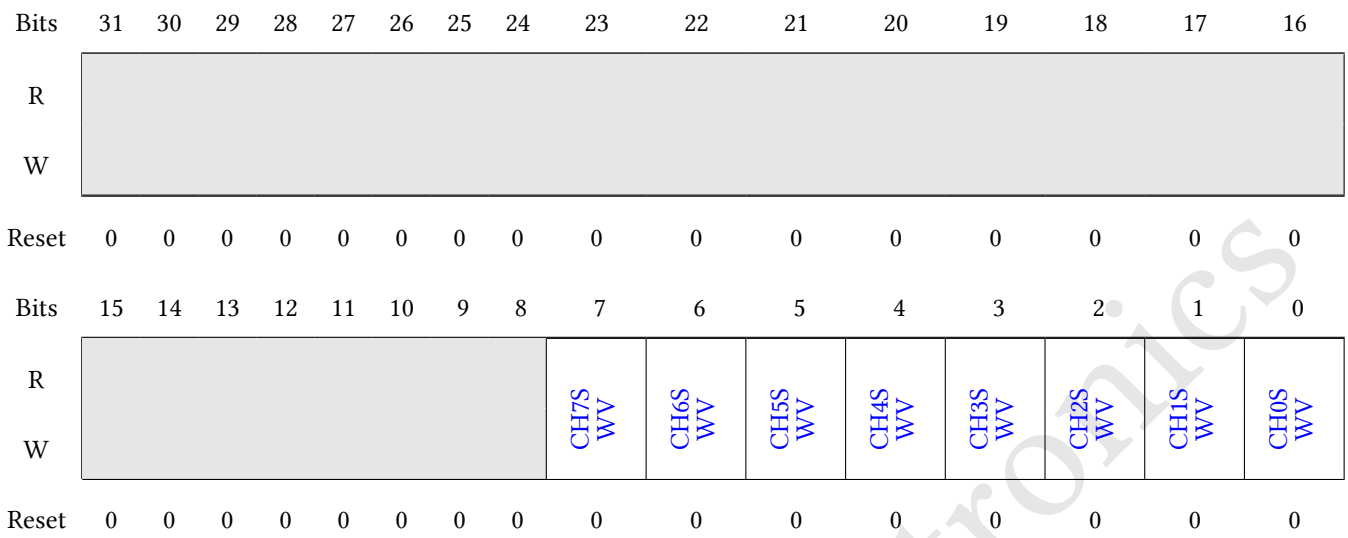


Table 30.17: eTMR CHSWV Register Description

Field	Function
7 CH7SWV	**Channel 7 Software Value** Channel 7 output value configured by software. 0b - Channel 7 output value forced to 0 by software. 1b - Channel 7 output value forced to 1 by software.
6 CH6SWV	**Channel 6 Software Value** Channel 6 output value configured by software. 0b - Channel 6 output value forced to 0 by software. 1b - Channel 6 output value forced to 1 by software.
5 CH5SWV	**Channel 5 Software Value** Channel 5 output value configured by software. 0b - Channel 5 output value forced to 0 by software. 1b - Channel 5 output value forced to 1 by software.
4 CH4SWV	**Channel 4 Software Value** Channel 4 output value configured by software. 0b - Channel 4 output value forced to 0 by software. 1b - Channel 4 output value forced to 1 by software.
3 CH3SWV	**Channel 3 Software Value** Channel 3 output value configured by software. 0b - Channel 3 output value forced to 0 by software. 1b - Channel 3 output value forced to 1 by software.
2	**Channel 2 Software Value**

Yuntu Microelectronics

Field	Function
CH2SWV	Channel 2 output value configured by software. 0b - Channel 2 output value forced to 0 by software. 1b - Channel 2 output value forced to 1 by software.
1 CH1SWV	**Channel 1 Software Value** Channel 1 output value configured by software. 0b - Channel 1 output value forced to 0 by software. 1b - Channel 1 output value forced to 1 by software.
0 CH0SWV	**Channel 0 Software Value** Channel 0 forced value by software. 0b - Channel 0 output value forced to 0 by software. 1b - Channel 0 output value forced to 1 by software.

30.3.1.11 eTMR CHMASK Register

Offset: 28h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH7MV		CH6MV		CH5MV		CH4MV		CH3MV		CH2MV		CH1MV		CH0MV	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W									CH7M	CH6M	CH5M	CH4M	CH3M	CH2M	CH1M	CH0M
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.18: eTMR CHMASK Register Description

Field	Function
31 - 30 CH7MV	**Channel 7 Output Mask Value** Define channel 7 output mask value, it's enabled when CHMASK[CH7M]=1. 00b - Channel 7 output mask value is 0. 01b - Channel 7 output mask value is 1. 10b - Channel 7 output mask value is tri-state. 11b - Channel 7 output mask value is tri-state.
29 - 28	**Channel 6 Output Mask Value**

Yuntu Microelectronics

Field	Function
CH6MV	Define channel 6 output mask value, it's enabled when CHMASK[CH6M]=1. 00b - Channel 6 output mask value is 0. 01b - Channel 6 output mask value is 1. 10b - Channel 6 output mask value is tri-state. 11b - Channel 6 output mask value is tri-state.
27 - 26 CH5MV	**Channel 5 Output Mask Value** Define channel 5 output mask value, it's enabled when CHMASK[CH5M]=1. 00b - Channel 5 output mask value is 0. 01b - Channel 5 output mask value is 1. 10b - Channel 5 output mask value is tri-state. 11b - Channel 5 output mask value is tri-state.
25 - 24 CH4MV	**Channel 4 Output Mask Value** Define channel 4 output mask value, it's enabled when CHMASK[CH4M]=1. 00b - Channel 4 output mask value is 0. 01b - Channel 4 output mask value is 1. 10b - Channel 4 output mask value is tri-state. 11b - Channel 4 output mask value is tri-state.
23 - 22 CH3MV	**Channel 3 Output Mask Value** Define channel 3 output mask value, it's enabled when CHMASK[CH3M]=1. 00b - Channel 3 output mask value is 0. 01b - Channel 3 output mask value is 1. 10b - Channel 3 output mask value is tri-state. 11b - Channel 3 output mask value is tri-state.
21 - 20 CH2MV	**Channel 2 Output Mask Value** Define channel 2 output mask value, it's enabled when CHMASK[CH2M]=1. 00b - Channel 2 output mask value is 0. 01b - Channel 2 output mask value is 1. 10b - Channel 2 output mask value is tri-state. 11b - Channel 2 output mask value is tri-state.
19 - 18 CH1MV	**Channel 1 Output Mask Value** Define channel 1 output mask value, it's enabled when CHMASK[CH1M]=1. 00b - Channel 1 output mask value is 0. 01b - Channel 1 output mask value is 1. 10b - Channel 1 output mask value is tri-state. 11b - Channel 1 output mask value is tri-state.
17 - 16 CH0MV	**Channel 0 Output Mask Value** Define channel 0 output mask value, it's enabled when CHMASK[CH0M]=1. 00b - Channel 0 output mask value is 0. 01b - Channel 0 output mask value is 1. 10b - Channel 0 output mask value is tri-state. 11b - Channel 0 output mask value is tri-state.

Yuntu Microelectronics

Field	Function
7 CH7M	**Channel 7 Output Mask** Channel 7 output mask control used for PWM mode. 0b - Channel 7 output PWM normally. 1b - Channel 7 output masked by CHMASK[CH7MV].
6 CH6M	**Channel 6 Output Mask** Channel 6 output mask control used for PWM mode. 0b - Channel 6 output PWM normally. 1b - Channel 6 output masked by CHMASK[CH6MV].
5 CH5M	**Channel 5 Output Mask** Channel 5 output mask control used for PWM mode. 0b - Channel 5 output PWM normally. 1b - Channel 5 output masked by CHMASK[CH5MV].
4 CH4M	**Channel 4 Output Mask** Channel 4 output mask control used for PWM mode. 0b - Channel 4 output PWM normally. 1b - Channel 4 output masked by CHMASK[CH4MV].
3 CH3M	**Channel 3 Output Mask** Channel 3 output mask control used for PWM mode. 0b - Channel 3 output PWM normally. 1b - Channel 3 output masked by CHMASK[CH3MV].
2 CH2M	**Channel 2 Output Mask** Channel 2 output mask control used for PWM mode. 0b - Channel 2 output PWM normally. 1b - Channel 2 output masked by CHMASK[CH2MV].
1 CH1M	**Channel 1 Output Mask** Channel 1 output mask control used for PWM mode. 0b - Channel 1 output PWM normally. 1b - Channel 1 output masked by CHMASK[CH1MV].
0 CH0M	**Channel 0 Output Mask** Channel 0 output mask control used for PWM mode. 0b - Channel 0 output PWM normally. 1b - Channel 0 output masked by CHMASK[CH0MV].

30.3.1.12 eTMR CHFV Register

Offset: 2ch

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7FV		CH6FV		CH5FV		CH4FV		CH3FV		CH2FV		CH1FV		CH0FV	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.19: eTMR CHFV Register Description

Field	Function
15 - 14 CH7FV	**Channel 7 Fault Value** Channel 7 forced value when fault event occurs. 00b - Channel 7 output value forced to 0. 01b - Channel 7 output value forced to 1. 10b - Channel 7 output value forced to tri-state. 11b - Channel 7 output value forced to tri-state.
13 - 12 CH6FV	**Channel 6 Fault Value** Channel 6 forced value when fault event occurs. 00b - Channel 6 output value forced to 0. 01b - Channel 6 output value forced to 1. 10b - Channel 6 output value forced to tri-state. 11b - Channel 6 output value forced to tri-state.
11 - 10 CH5FV	**Channel 5 Fault Value** Channel 5 forced value when fault event occurs. 00b - Channel 5 output value forced to 0. 01b - Channel 5 output value forced to 1. 10b - Channel 5 output value forced to tri-state. 11b - Channel 5 output value forced to tri-state.
9 - 8 CH4FV	**Channel 4 Fault Value** Channel 4 forced value when fault event occurs. 00b - Channel 4 output value forced to 0. 01b - Channel 4 output value forced to 1. 10b - Channel 4 output value forced to tri-state. 11b - Channel 4 output value forced to tri-state.

Yuntu Microelectronics

Field	Function
7 - 6 CH3FV	**Channel 3 Fault Value** Channel 3 forced value when fault event occurs. 00b - Channel 3 output value forced to 0. 01b - Channel 3 output value forced to 1. 10b - Channel 3 output value forced to tri-state. 11b - Channel 3 output value forced to tri-state.
5 - 4 CH2FV	**Channel 2 Fault Value** Channel 2 forced value when fault event occurs. 00b - Channel 2 output value forced to 0. 01b - Channel 2 output value forced to 1. 10b - Channel 2 output value forced to tri-state. 11b - Channel 2 output value forced to tri-state.
3 - 2 CH1FV	**Channel 1 Fault Value** Channel 1 forced value when fault event occurs. 00b - Channel 1 output value forced to 0. 01b - Channel 1 output value forced to 1. 10b - Channel 1 output value forced to tri-state. 11b - Channel 1 output value forced to tri-state.
1 - 0 CH0FV	**Channel 0 Fault Value** Channel 0 forced value when fault event occurs. 00b - Channel 0 output value forced to 0. 01b - Channel 0 output value forced to 1. 10b - Channel 0 output value forced to tri-state. 11b - Channel 0 output value forced to tri-state.

Yuntu Microelectronics

30.3.1.13 eTMR INIT Register

This register is 32-bit valid in eTMR3, and 16-bit in other instances.

Offset: 30h

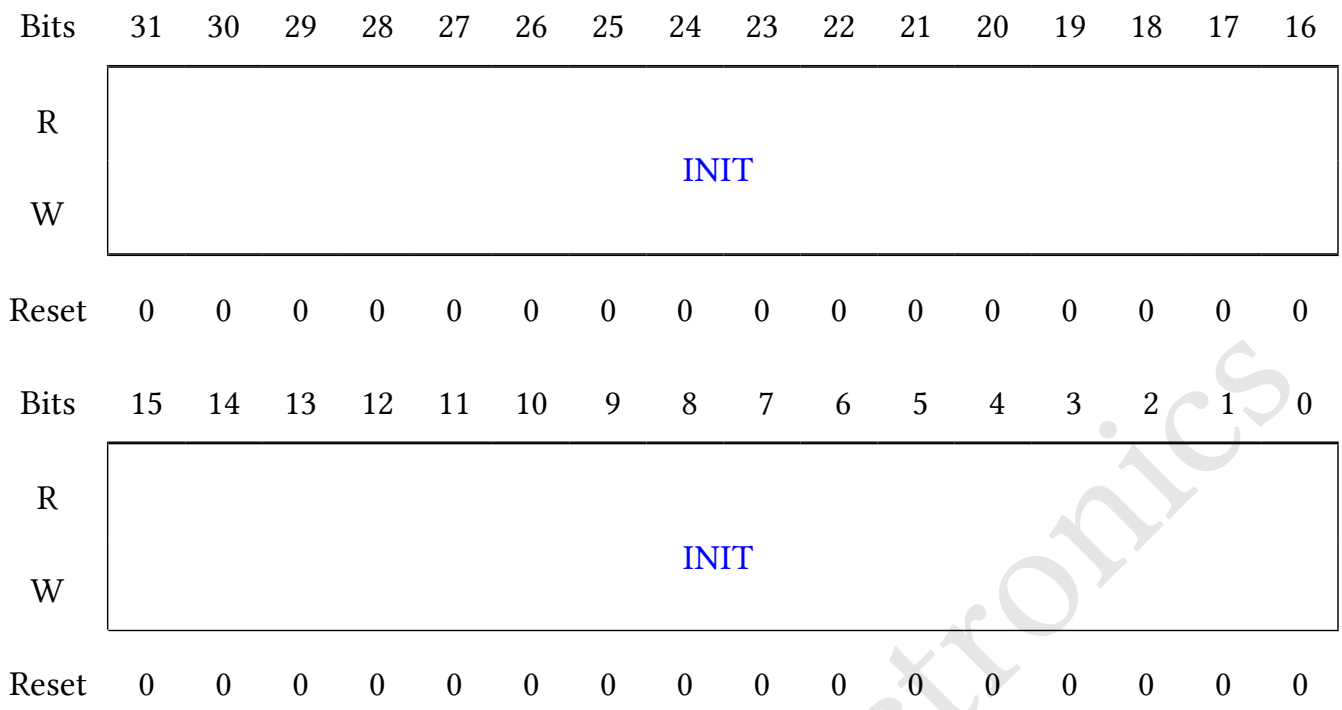


Table 30.20: eTMR INIT Register Description

Field	Function
31 - 0 INIT	**Counter Initial Value** Counter initial value

Yuntu Microelectronics

30.3.1.14 eTMR MOD Register

This register is 32-bit valid in eTMR3, and 16-bit in other instances. Since the counter is 32-bit in eTMR3. The default value is 0xFFFFFFFF in eTMR3, 0x0000FFFF in other eTMRs.

Offset: 34h

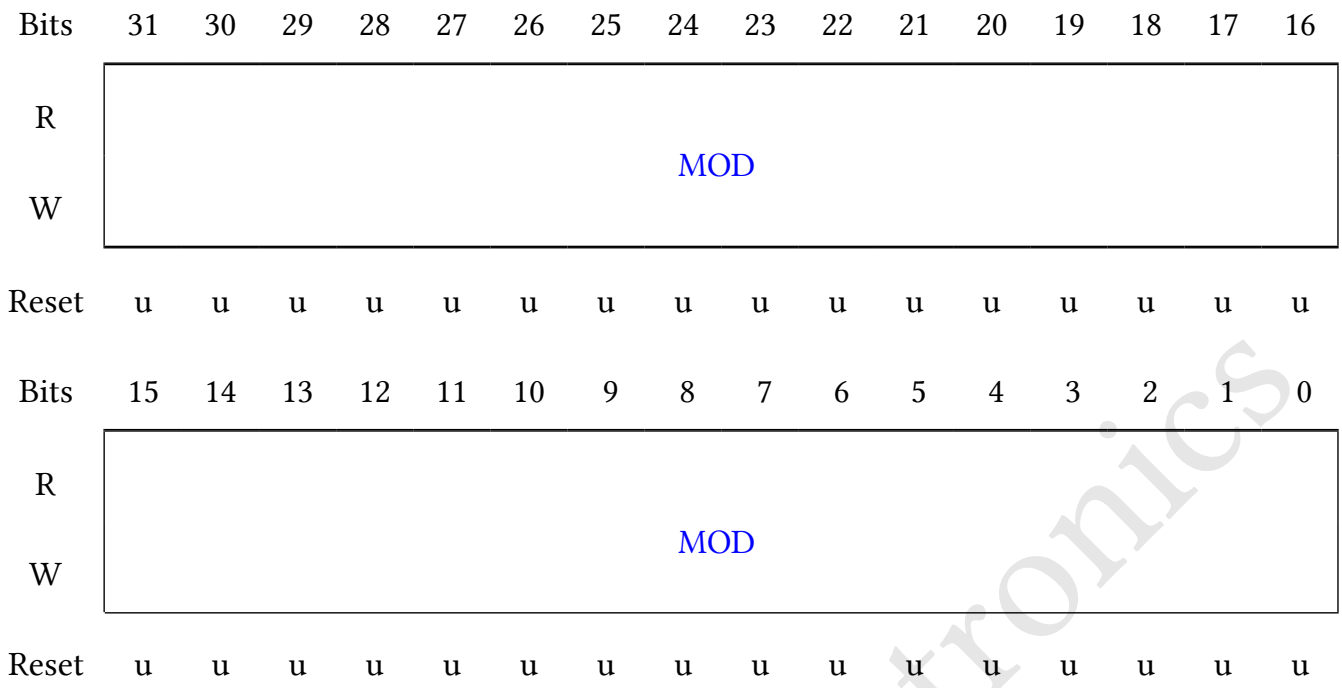


Table 30.21: eTMR MOD Register Description

Field	Function
31 - 0 MOD	**Counter Modulus value** Counter modulus value.

Yuntu Microelectronics

30.3.1.15 eTMR DITHMOD Register

Offset: 38h

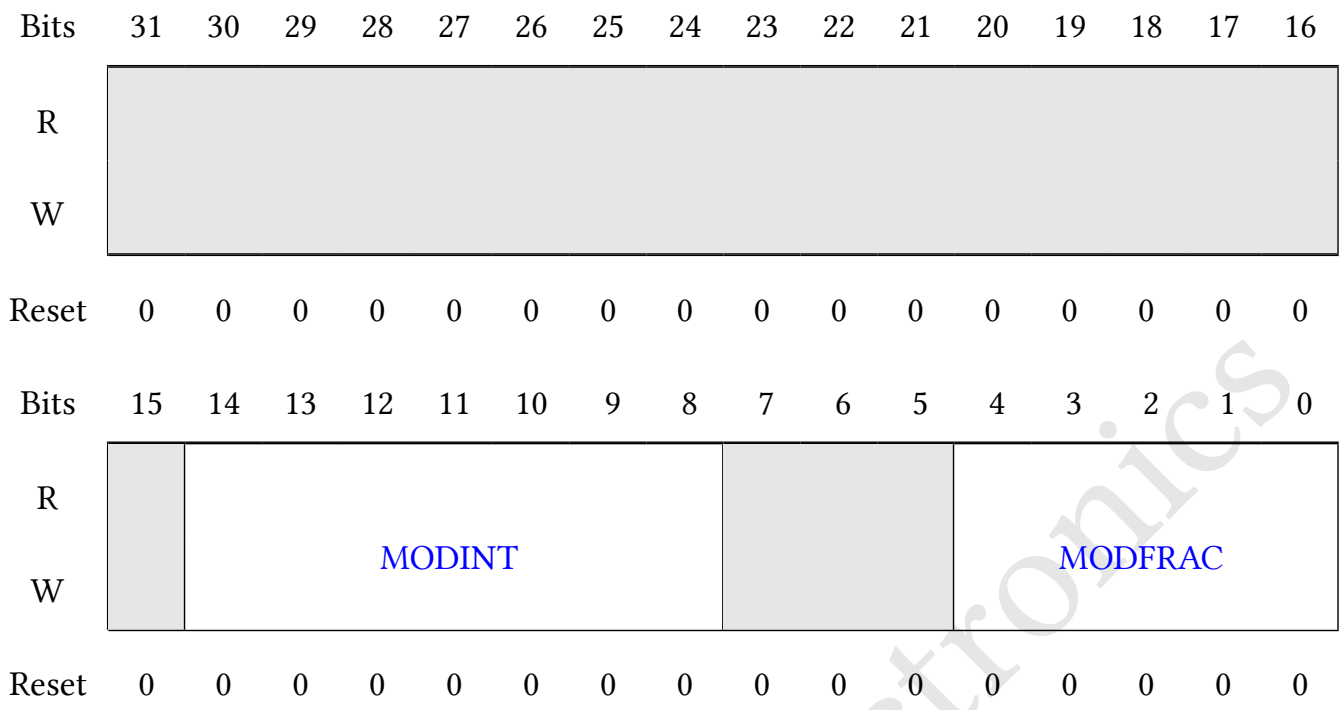


Table 30.22: eTMR DITHMOD Register Description

Field	Function
14 - 8 MODINT	**Modulus Integer Part** The integer part used for period dithering.
4 - 0 MODFRAC	**Modulus Fractional Part** The fractional part used for period dithering.

Yuntu Microelectronics

30.3.1.16 eTMR MID Register

This register is 32-bit valid in eTMR3, and 16-bit in other instances. The default value is 0x0000FFFF in eTMR3, others' are 0x000000FF.

Offset: 3ch

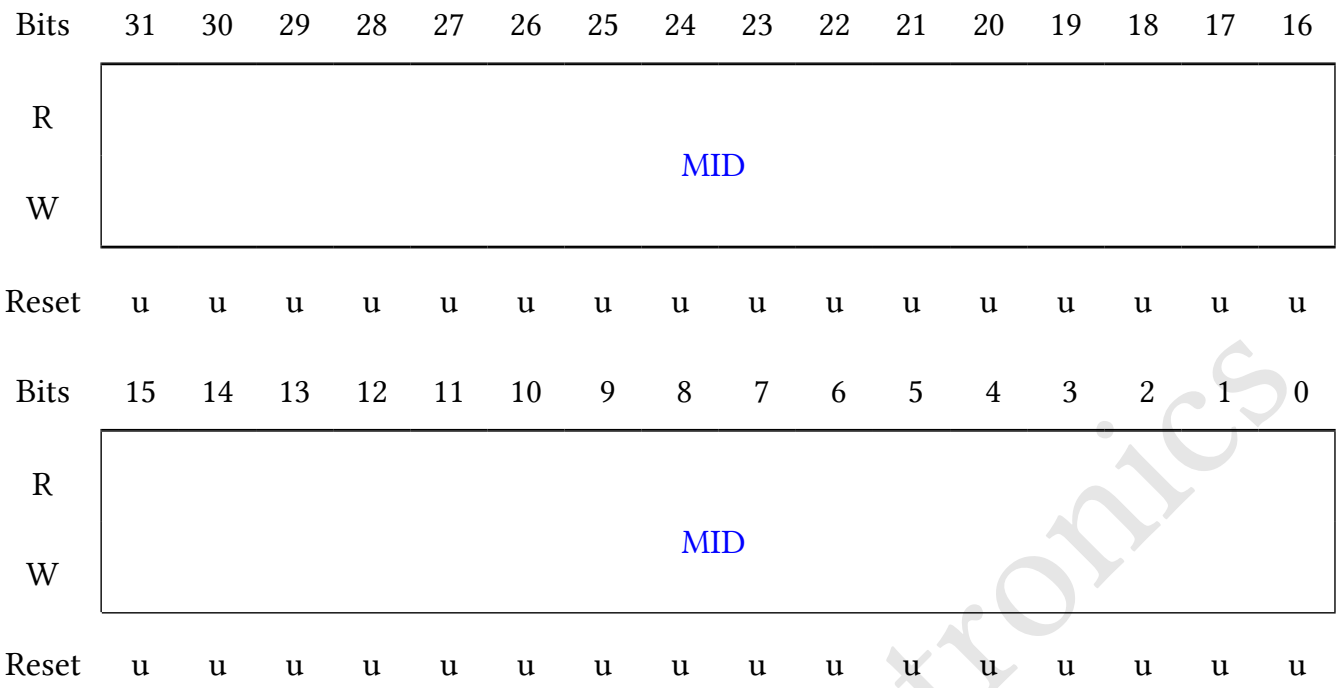
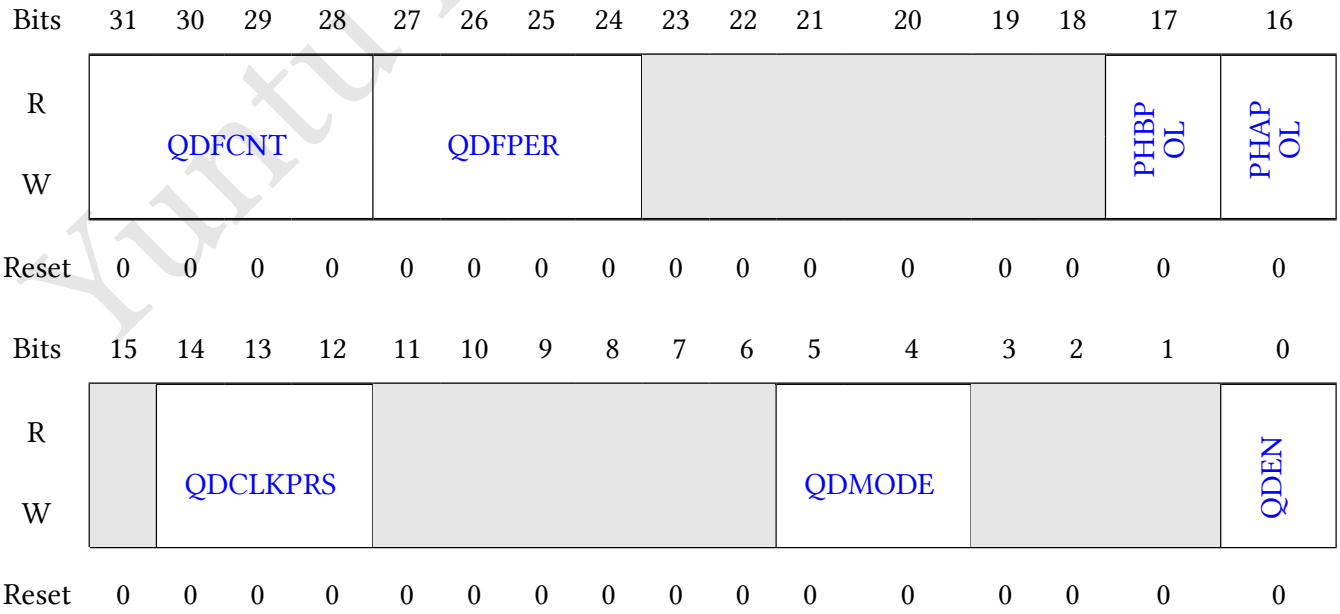


Table 30.23: eTMR MID Register Description

Field	Function
31 - 0 MID	**Counter Middle Value** Counter middle value.

30.3.1.17 eTMR QDCTRL Register

Offset: 40h



Yuntu Microelectronics

Table 30.24: eTMR QDCTRL Register Description

Field	Function
31 - 28 QDFCNT	**Quadrature Decoder Filter Sample Count** Quadrature decoder phase A and phase B input filter sample count, the input is ignored when its valid width was less than QDFCNT x QDFPER.
27 - 24 QDFPER	**Quadrature Decoder Filter Sample Period** Quadrature decoder phase A and phase B input filter sample period, the input is ignored when its valid width was less than QDFCNT x QDFPER.
17 PHBPOL	**Phase B Polarity** Phase B polarity selection. 0b - Not invert phase B input. 1b - Invert phase B input.
16 PHAPOL	**Phase A Polarity** Phase A polarity selection. 0b - Not invert phase A input. 1b - Invert phase A input.
14 - 12 QDCLKPRS	**Quadrature Decoder Clock Prescaler** Clock prescaler for quadrature decoder counter. 000b - Divided by 1. 001b - Divided by 2. 010b - Divided by 4. 011b - Divided by 8. 100b - Divided by 16. 101b - Divided by 32. 110b - Divided by 64. 111b - Divided by 128.
5 - 4 QDMODE	**Quadrature Decoder Mode** Select decoding mode in the quadrature decoder mode. 00b - Phase A and phase B decoding mode. When phase B changes follow phase A, the counter increases. When phase A changes follow phase B, the counter decreases. 01b - Phase A and phase B decoding mode. When phase A changes follow phase B, the counter increases. When phase B changes follow phase A, the counter decreases. 10b - Counting and direction decoding mode. Phase B input controls direction. When phase B input is high, the counter increases with phase A pulse. When phase B input is low, the counter decreases with phase A pulse. 11b - Counting and direction decoding mode. Phase A input controls direction. When phase A input is high, the counter increases with phase B pulse. When phase A input is low, the counter decreases with phase B pulse.
0	**Quadrature Decoder Enable**

Yuntu Microelectronics

Field	Function
QDEN	Quadrature decoder enable/disable. 0b - Quadrature decoder is disabled. 1b - Quadrature decoder is enabled.

30.3.1.18 eTMR QDINIT Register

Offset: 44h

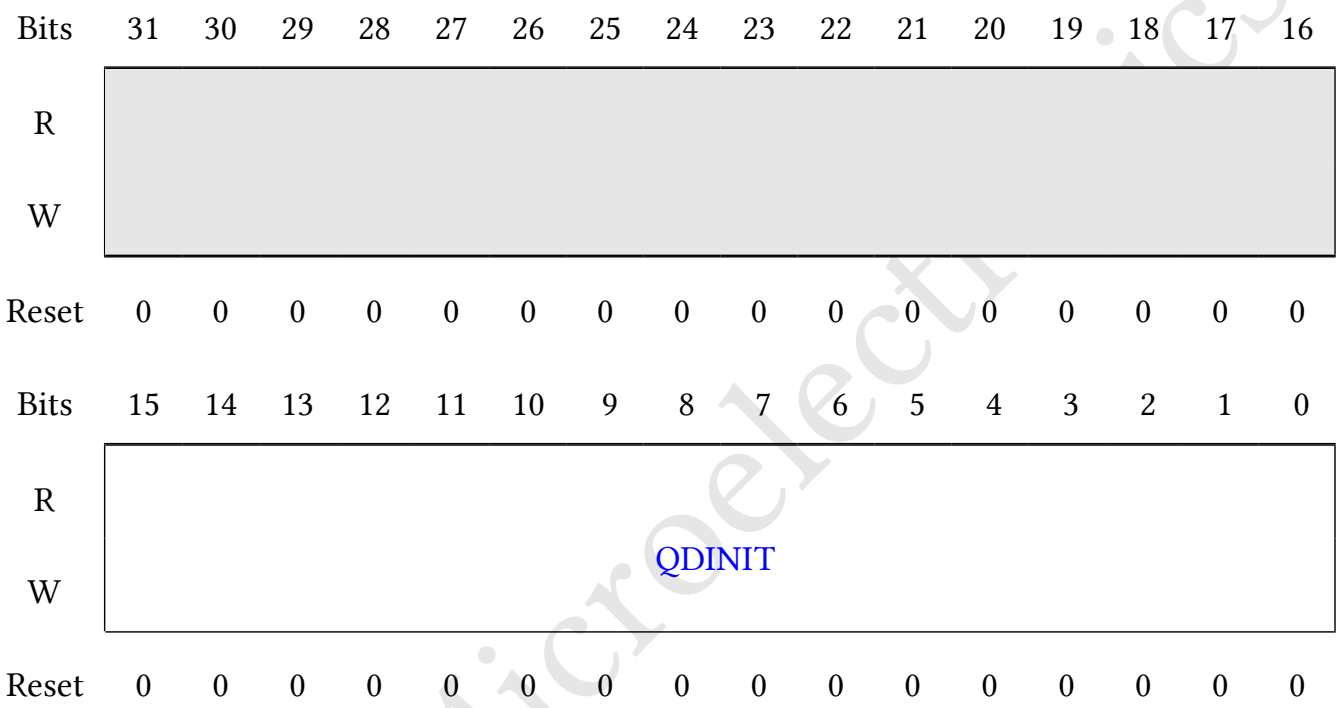


Table 30.25: eTMR QDINIT Register Description

Field	Function
15 - 0 QDINIT	**Quadrature Decoder Counter Initial value** Quadrature decoder counter initial value.

30.3.1.19 eTMR QDMOD Register

The default value is 0x0000FFFF in eTMR1 and eTMR2, others are 0x00000000.

Offset: 48h

Yuntu Microelectronics

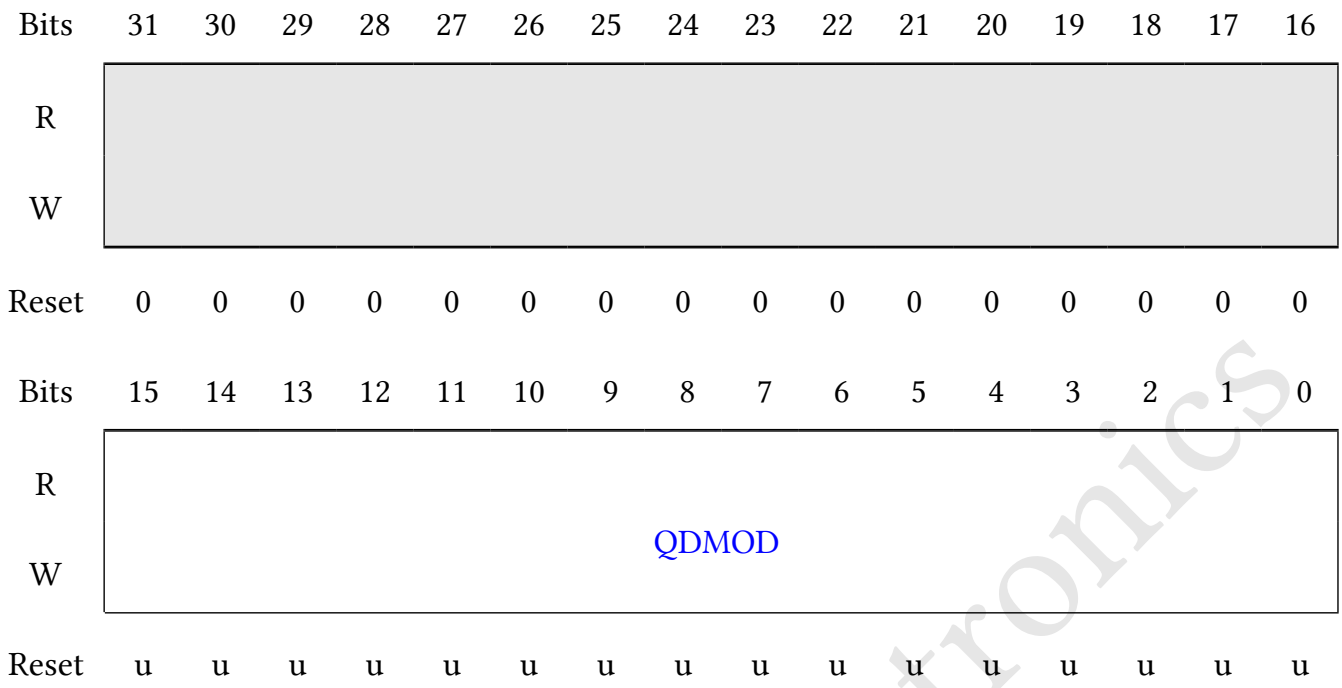
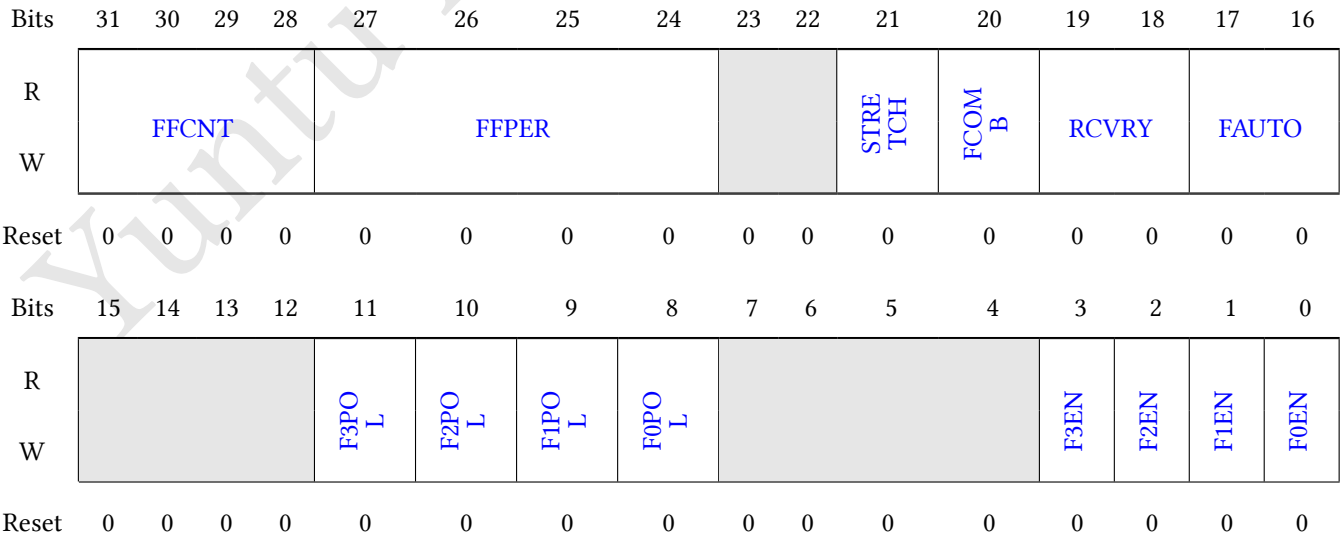


Table 30.26: eTMR QDMOD Register Description

Field	Function
15 - 0 QDMOD	**Quadrature Decoder Counter Modulus value** Quadrature decoder counter Modulus value.

30.3.1.20 eTMR FAULT Register

Offset: 4ch



Yuntu Microelectronics

Table 30.27: eTMR FAULT Register Description

Field	Function
31 - 28 FFCNT	**Fault Filter Sample Count** Fault input filter sample count, the input is ignored when its valid width was less than FFCNT x FFPER.
27 - 24 FFPER	**Fault Filter Sample Period** Fault input filter sample period, the input is ignored when its valid width was less than FFCNT x FFPER.
21 STRETCH	**Fault Input Stretch** Fault input stretch to 2 bus clock cycles width at least.
20 FCOMB	**Fault Combinational Logic** When there is no bus clock, the fault event is generated by the combinational logic circuit. 0b - Fault combinational logic disable. 1b - Fault combinational logic enable.
19 - 18 RCVRY	**PWM Recovery** Select PWM recovery opportunity after fault revoke. 00b - PWM recovery is disabled. 01b - PWM can recover when counter matches with MID[MID]. 10b - PWM can recover when counter matches with MOD[MOD]. 11b - PWM can recover immediately.
17 - 16 FAUTO	**Fault Automatic Recovery Mode** Select fault automatic recovery mode after the fault event end. 00b - After clearing STS[FxF] flag manually, the fault recovers automatically. 01b - The fault recovers immediately, not care STS[FxF] flag. 10b - The fault recovers immediately and STS[FxF] flag clear automatically. 11b - Reserved.
11 F3POL	**Fault 3 Input Polarity** Fault 3 input polarity selection. 0b - High level signal is regarded as fault signal. 1b - Invert fault 3 input. Low level signal is regarded as fault signal.
10 F2POL	**Fault 2 Input Polarity** Fault 2 input polarity selection. 0b - High level signal is regarded as fault signal. 1b - Invert fault 2 input. Low level signal is regarded as fault signal.
9	**Fault 1 Input Polarity**

Yuntu Microelectronics

Field	Function
F1POL	Fault 1 input polarity selection. 0b - High level signal is regarded as fault signal. 1b - Invert fault 1 input. Low level signal is regarded as fault signal.
8 F0POL	**Fault 0 Input Polarity** Fault 0 input polarity selection. 0b - High level signal is regarded as fault signal. 1b - Invert fault 0 input. Low level signal is regarded as fault signal.
3 F3EN	**Fault 3 Enable** Fault 3 enable. 0b - Fault 3 disable. 1b - Fault 3 enable.
2 F2EN	**Fault 2 Enable** Fault 2 enable. 0b - Fault 2 disable. 1b - Fault 2 enable.
1 F1EN	**Fault 1 Enable** Fault 1 enable. 0b - Fault 1 disable. 1b - Fault 1 enable.
0 F0EN	**Fault 0 Enable** Fault 0 enable. 0b - Fault 0 disable. 1b - Fault 0 enable.

30.3.1.21 eTMR CH_CTRL Register

Offset: 80h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W		DMAEN		LDEN			CAPEDGE		CFCNT				CFPER			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL1CMP		VAL0CMP				PWMSRC		VAL1TEN	VAL0TEN	CHPOL	CENTER			CHMODE	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.28: eTMR CH_CTRL Register Description

Field	Function
30	**DMA Enable**

Yuntu Microelectronics

Field	Function
DMAEN	DMA enable/disable for compare/capture mode. 0b - DMA is disabled. 1b - DMA is enabled.
28 LDEN	**Registers Loading Enable** Current channel registers (CH_VAL0/CH_VAL1) loading enable/disable. 0b - Registers Loading is disabled. 1b - Registers Loading is enabled.
25 - 24 CAPEDGE	**Capture Edge** Select the capture edge in the input capture mode. 00b - Capture is disabled. 01b - Capture rising edge. 10b - Capture falling edge. 11b - Capture both rising edge and falling edge.
23 - 20 CFCNT	**Capture Filter Sample Counter** Capture input filter sample counter, the input is ignored when its valid width was less than CFCNT x CFPER.
19 - 16 CFPER	**Capture Filter Sample Period** Capture input filter sample period, the input is ignored when its valid width was less than CFCNT x CFPER.
15 - 14 VAL1CMP	**VAL1 Compare Mode** Select the output value when counter matches with CH_VAL1[VAL1] in compare mode. 00b - Output value 0 when counter matches with CH_VAL1[VAL1]. 01b - Output value 1 when counter matches with CH_VAL1[VAL1]. 10b - Output toggle when counter matches with CH_VAL1[VAL1]. 11b - Reserved.
13 - 12 VAL0CMP	**VAL0 Compare Mode** Select the output value when counter matches with CH_VAL0[VAL0] in compare mode. 00b - Output value 0 when counter matches with CH_VAL0[VAL0]. 01b - Output value 1 when counter matches with CH_VAL0[VAL0]. 10b - Output toggle when counter matches with CH_VAL0[VAL0]. 11b - Reserved.
9 - 8 PWMSRC	**PWM Source** Select PWM generation source in PWM mode. 00b - Normal mode, PWM is generated by VAL0 and VAL1. 01b - Invert mode, invert the PWM in normal mode. 10b - Software mode, PWM is generated by CHSWV register. 11b - Reserved.
7	**VAL1 Trigger Enable**

Yuntu Microelectronics

Field	Function
VAL1TEN	Enable/disable generating trigger when counter matches with CH_VAL1[VAL1]. 0b - Not generate trigger when counter matches with CH_VAL1[VAL1]. 1b - Generate trigger when counter matches with CH_VAL1[VAL1].
6 VAL0TEN	**VAL0 Trigger Enable** Enable/disable generating trigger when counter matches with CH_VAL0[VAL0]. 0b - Not generate trigger when counter matches with CH_VAL0[VAL0]. 1b - Generate trigger when counter matches with CH_VAL0[VAL0].
5 CHPOL	**Channel Output Polarity** Channel output polarity control. 0b - Not invert the channel output. 1b - Invert the channel output.
4 CENTER	**Center Aligned Mode** Center aligned PWM generate when center aligned mode enabled. 0b - Center aligned mode disabled. 1b - Center aligned mode enabled.
1 - 0 CHMODE	**Channel Mode** Select channel mode. 00b - Common timer. 01b - PWM mode. 10b - Compare mode. 11b - Capture mode.

30.3.1.22 eTMR CH_VAL0 Register

This register is 32-bit valid in eTMR3. Other eTMRs are 16-bit valid.

Offset: 84h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.29: eTMR CH_VAL0 Register Description

Field	Function
31 - 0 VAL0	**Value 0** Define the value 0.

30.3.1.23 eTMR CH_VAL1 Register

This register is 32-bit valid in eTMR3. Other eTMRs are 16-bit valid.

Offset: 88h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VAL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VAL1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.30: eTMR CH_VAL1 Register Description

Field	Function
31 - 0 VAL1	**Value 1** Define the value 1.

30.3.1.24 eTMR CH_DITHVAL1 Register

Offset: 8ch + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W		VAL1INT										VAL1FRAC				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.31: eTMR CH_DITHVAL1 Register Description

Field	Function
14 - 8 VAL1INT	**Value 1 Integer Part** The integer part used for duty cycle dithering.
4 - 0 VAL1FRAC	**Value 1 Fractional Part** The fractional part used for duty cycle dithering.

30.3.1.25 eTMR CH_DT Register

Offset: 90h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DTVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.32: eTMR CH_DT Register Description

Field	Function
15 - 0 DTVAL	**Deadtime Value** Deatime value used in complementary mode.

30.3.1.26 eTMR CH_CVAL Register

This register is 32-bit valid in eTMR3. Other eTMRs are 16-bit valid.

Offset: 94h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.33: eTMR CH_CVAL Register Description

Field	Function
31 - 0 CVAL	**Capture Value** In capture mode, store counter value when a capture event occurs.

30.3.1.27 eTMR CH_PPCNTN Register

This register is 32-bit valid in eTMR3, other eTMRs are 16-bit valid.

Offset: 98h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W									PPCNTN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.34: eTMR CH_PPCNTN Register Description

Field	Function
7 - 0 PPCNTN	**Positive Pulse Counter Periods Number** Positive pulse width measurement counter period number.

30.3.1.28 eTMR CH_PPCNTV Register

This register is 32-bit valid in eTMR3, other eTMRs are 16-bit valid.

Offset: 9ch + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PPCNTV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PPCNTV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.35: eTMR CH_PPCNTV Register Description

Field	Function
31 - 0 PPCNTV	**Positive Pulse Width Capture Value** Positive pulse width measurement capture counter value.

30.3.1.29 eTMR CH_NPCNTN Register

This register is 32-bit valid in eTMR3, other eTMRs are 16-bit valid.

Offset: a0h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W									NPCNTN							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 30.36: eTMR CH_NPCNTN Register Description

Field	Function
7 - 0 NPCNTN	**Negative Pulse Counter Periods Number** Negative pulse width measurement counter period number.

30.3.1.30 eTMR CH_NPCNTV Register

This register is 32-bit valid in eTMR3, other eTMRs are 16-bit valid.

Offset: a4h + n * 40h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NPCNTV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NPCNTV															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 30.37: eTMR CH_NPCNTV Register Description

Field	Function
31 - 0 NPCNTV	**Negative Pulse Width Capture Value** Negative pulse width measurement capture counter value.

30.4 Functional Description

30.4.1 Clock

The eTMR module works in bus clock domain. Two clock sources can be selected via CTRL[CLKSRC] to drive the 16-bit counter: bus clock and external clock. The external clock signal is from the pad and synchronized by bus clock. So, the bus clock frequency should be at least two times the external clock frequency. CIM_ETMROPT0[ETMRxECS] defines three external clock sources that can be selected, see ETMR Option 0 Register for details. In addition, the eTMR module supports a 7-bit clock prescaler used for dividing the counter clock. The prescaler supports frequency division from 1 to 128.

Counter clock CLK_{CNT} configuration:

- Select clock source CLK_{SRC} by CTRL[CLKSRC]
- Divide the clock source CLK_{SRC} by CTRL[CLKPRS]

$$CLK_{CNT} = \frac{CLK_{SRC}}{CTRL[CLKPRS] + 1}$$

30.4.2 Counter

Each eTMR module has a 16-bit counter. The counter supports up-counting mode.

When the module is disabled, The initial value of the counter is configurable by SYNC[CNTINIT]. When the module is enabled and loading event occurs, counter value initial from INIT[INIT] register.

Anyway, when the counter reaches MOD[MOD], the next value is from INIT[INIT] register. As shown in Figure eTMR Counter Overflow. If the counter increases from MOD[MOD] to INIT[INIT], the timeout flag (STS[TOF]) will be set.

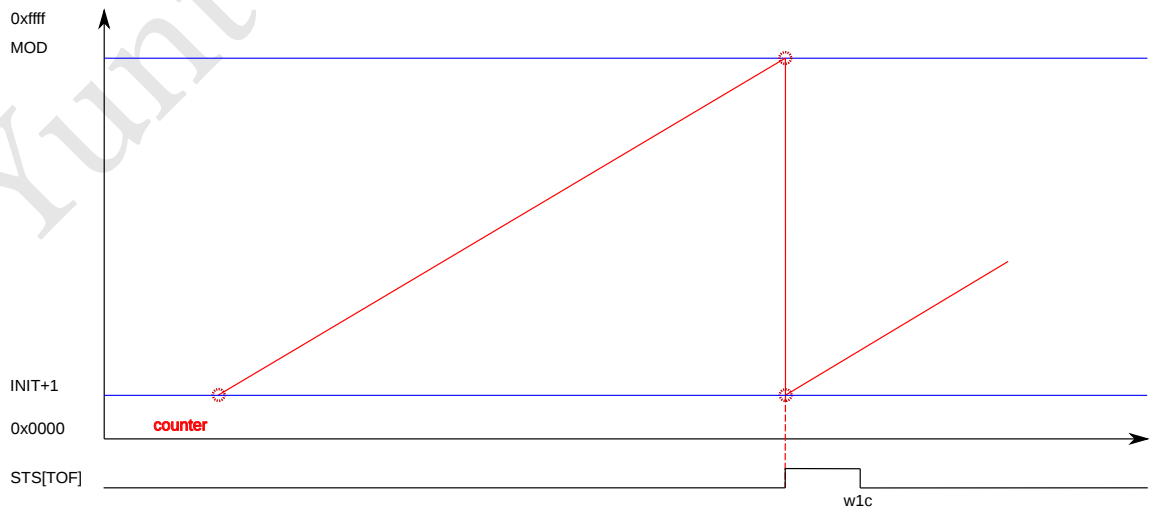


Figure 30.2: eTMR Counter Overflow

Yuntu Microelectronics

30.4.2.1 Global Time Base

Different eTMRs' counter can counts synchronously on the same time base. One eTMR acts as the master, other eTMRs are as slaves. CTRL[GLOBEN] = 1 means current eTMR works in master-slave mode. SYNC[GLOB] = 1 means current eTMR acts as master and other eTMRs act as slaves.

30.4.3 Common timer

When all CHx_CTRL[CHMODE]=0x0 and eTMR enabled. eTMR is as a common timer which can generate a timer overflow interrupt.

30.4.4 PWM mode

When CHx_CTRL[CHMODE]=0x1, the channel x is in PWM mode. Each channel is capable of generating center aligned PWM, edge aligned PWM and asymmetrical PWM. All channels support independent mode and specific channel pairs support complementary mode. In addition, the eTMR supports output fault control and mask control in PWM mode. As shown in Figure eTMR PWM Generation Block Diagram.

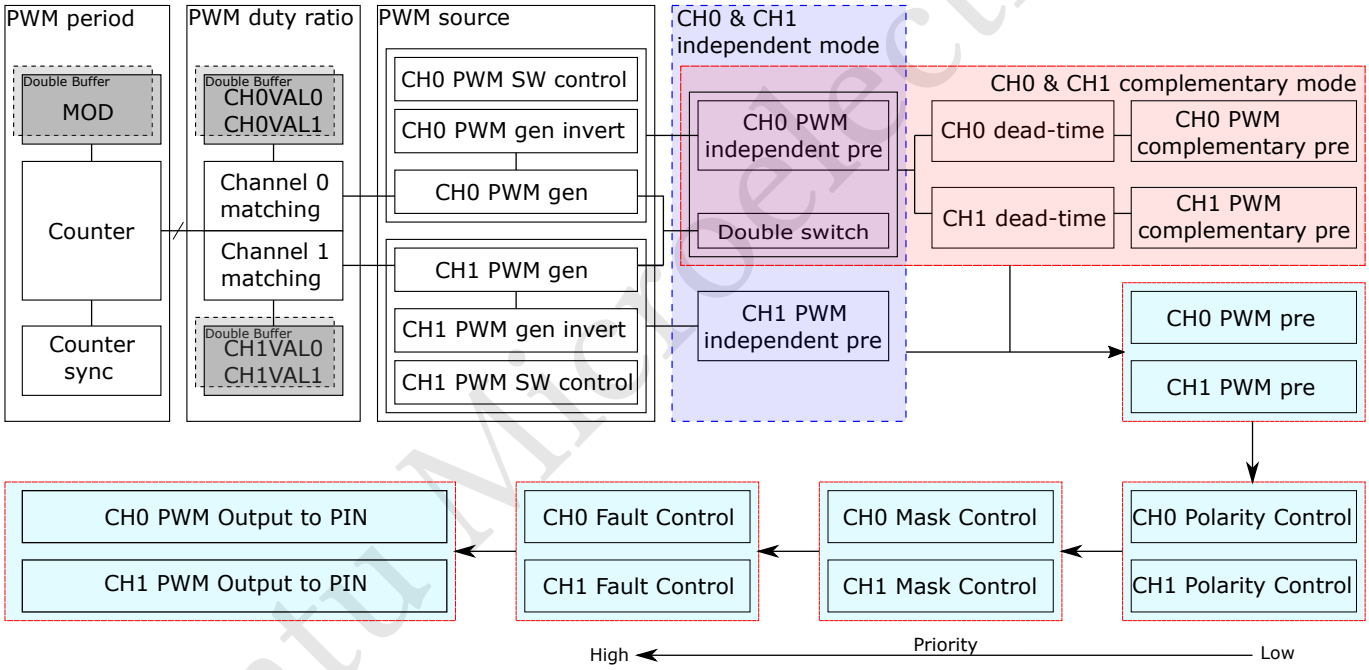


Figure 30.3: eTMR PWM Generation Block Diagram

The period of output PWM is up to counter clock and MOD[MOD]:

$$PER_{PWM} = \frac{MOD[MOD] - INIT[INIT]}{CLK_{CNT}}$$

The duty cycle of output PWM is up to CH_VAL0[VAL0] and CH_VAL1[VAL1], PWM set when the counter matches CH_VAL0[VAL0], PWM clear when the counter matches CH_VAL1[VAL1]:

- CH_VAL0[VAL0] < CH_VAL1[VAL1]:

$$DUTY_{PWM} = \frac{CH_VAL1[VAL1] - CH_VAL0[VAL0]}{PER_{PWM}} \times 100\%$$

Yuntu Microelectronics

- CH_VAL0[VAL0] > CH_VAL1[VAL1]:

$$DUTY_{PWM} = 1 - \frac{CH_VAL0[VAL0] - CH_VAL1[VAL1]}{PER_{PWM}} \times 100\%$$

30.4.4.1 Center Aligned PWM

When CHx_CTRL[CENTER]=1'b0, center aligned PWM configured by INIT, CHx_VAL0, CHx_VAL1 and MOD shown in Figure eTMR PWM Generation Block Diagram. The configuration steps are as follows:

- Determine period and $DUTY_{PWM}$ as needed
- Calculate CHx_VAL0[VAL0] and CHx_VAL1[VAL1]

$$CH_VAL0[VAL0] = \frac{(1 - DUTY_{PWM}) \times (MOD[MOD] - INIT[INIT])}{2}$$

$$CH_VAL1[VAL1] = \frac{(1 + DUTY_{PWM}) \times (MOD[MOD] - INIT[INIT])}{2}$$

When CHx_CTRL[CENTER]=1'b1, center aligned PWM configured by INIT, CHx_VAL0 and MOD. CHx_VAL0 and CHx_VAL1 is updated from $DUTY_{PWM}$(initial CHx_VAL0) when center aligned PWM generates shown in Figure eTMR PWM Generation Block Diagram. The configuration steps are as follows: - Determine period and $DUTY_{PWM}$ as needed - Configue INIT and MOD from period - Configue CHx_VAL0 from $DUTY_{PWM}$

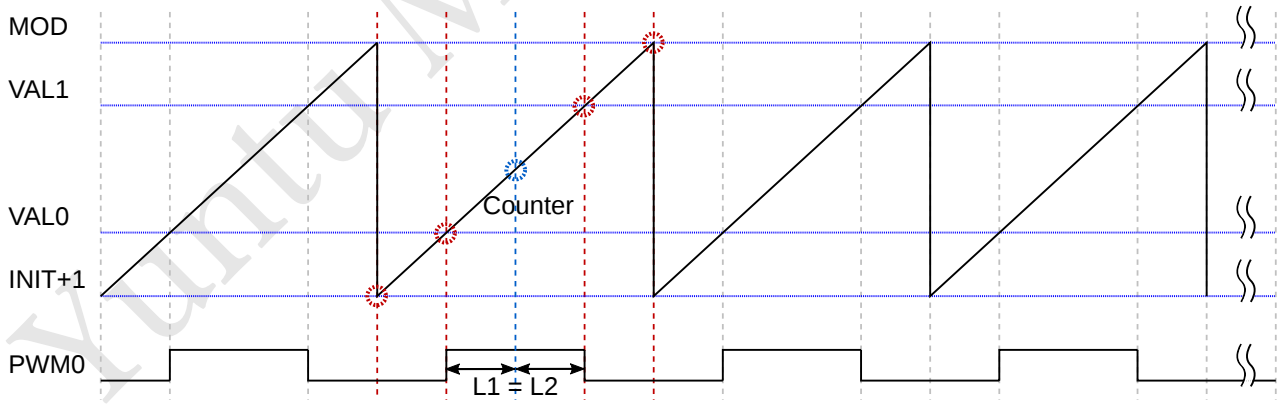


Figure 30.4: eTMR Center Aligned PWM

30.4.4.2 Edge Aligned PWM

Edge aligned PWM contains left edge aligned PWM and right edge aligned PWM. As shown in Figures eTMR Left Edge Aligned PWM and eTMR Right Edge Aligned PWM

Yuntu Microelectronics

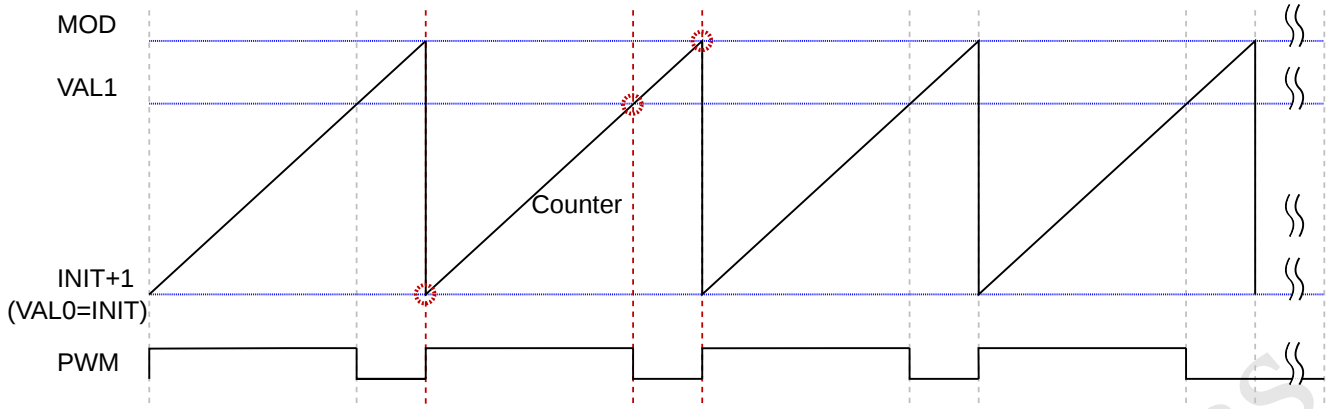


Figure 30.5: eTMR Left Edge Aligned PWM

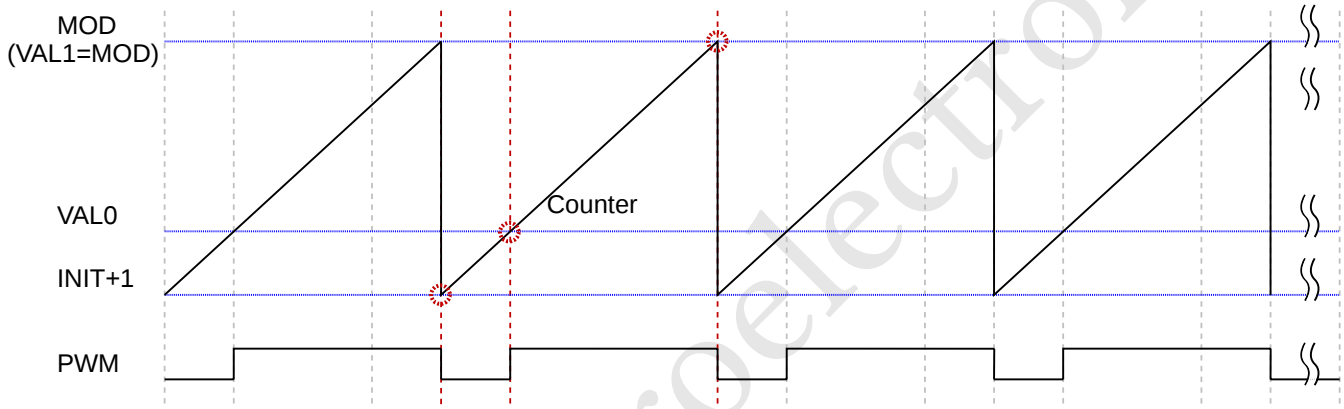


Figure 30.6: eTMR Right Edge Aligned PWM

Left edge aligned PWM configuration:

- Determine MOD[MOD] and $DUTY_{PWM}$ as needed
- Configure CH_VAL0[VAL0]=INIT[INIT]
- Calculate CH_VAL1[VAL1]

$$CH_VAL1[VAL1] = DUTY_{PWM} \times (MOD[MOD] - INIT[INIT]) + INIT[INIT]$$

Right edge aligned PWM configuration:

- Determine MOD[MOD] and $DUTY_{PWM}$ as needed
- Configure CH_VAL1[VAL1]=MOD[MOD]
- Calculate CH_VAL0[VAL0]

$$CH_VAL0[VAL0] = (1 - DUTY_{PWM}) \times (MOD[MOD] - INIT[INIT]) + INIT[INIT]$$

30.4.4.3 Asymmetrical PWM

Because the rising edge and falling edge of PWM is both configurable, it's easy to generate asymmetrical PWM by configuring CH_VAL0[VAL0] and CH_VAL1[VAL1]. As shown in Figure eTMR Asymmetrical PWM.

Asymmetrical PWM configuration:

Yuntu Microelectronics

- Determine MOD[MOD], and $DUTY_{PWM}$ as needed
- Configure CH_VAL0[VAL0] and CH_VAL1[VAL1] as needed

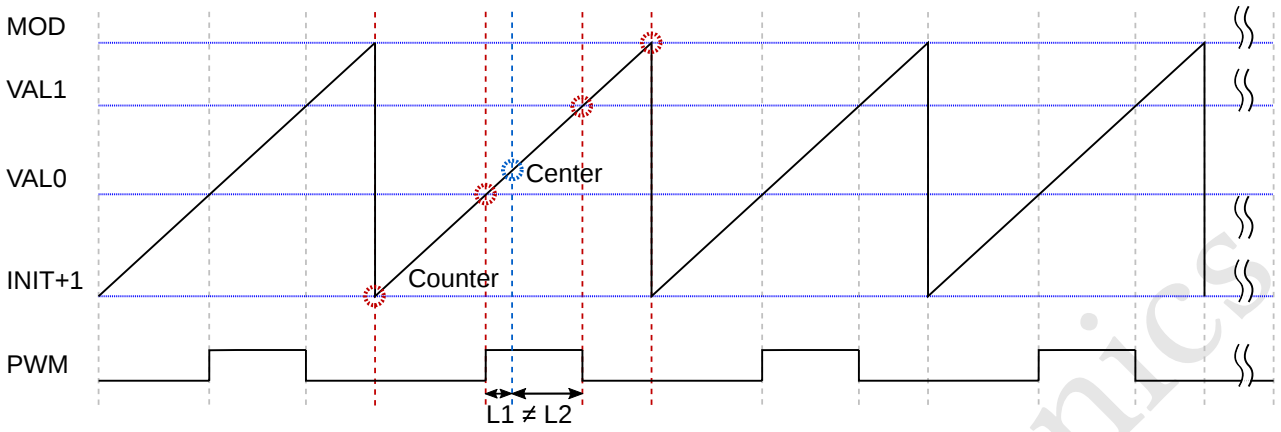


Figure 30.7: eTMR Asymmetrical PWM

30.4.4.4 Software-generated PWM

Software-generated PWM is generated by register CHSWV[CHxSWV] writing operations. Software-generated PWM sets when writing 1 to CHSWV[CHxSWV] and clears when writing 0 to CHSWV[CHxSWV].

30.4.4.5 Double-switch PWM

Each eMTR contains different pairs: channel 0-1, channel 2-3, channel 4-5 and channel 6-7. Any channel pair can generate two different PWMs(PWMa and PWMb). Double-switch PWM is PWMa XOR PWMb. Generally double-switch PWM used as a complementary output source for motor application. This function is enabled when CTRL[DBSWxx]=1.

30.4.4.6 PWM Source

The source of PWM is selected by CH_CTRL[PWMSRC]:

- CH_CTRL[PWMSRC]=0: select the raw PWM.
- CH_CTRL[PWMSRC]=1: select the inverted PWM, the raw PWM is inverted.
- CH_CTRL[PWMSRC]=2: select the software controlled PWM, the raw PWM is covered by CHSWV[CHxSWV].

30.4.4.7 Independent Mode

Each channel of the eTMR can output PWM independently, which is the independent mode. Channel 0/1 output PWM0/1 independently as shown in Figure eTMR Independent PWM.

Every channel has it's own control register, so it can use specific configurations to generate the independent PWM.

Independent PWM is selected when:

- CHx_CTRL[CHMODE]=0x1
- CTRL[COMPLxx]=0

Yuntu Microelectronics

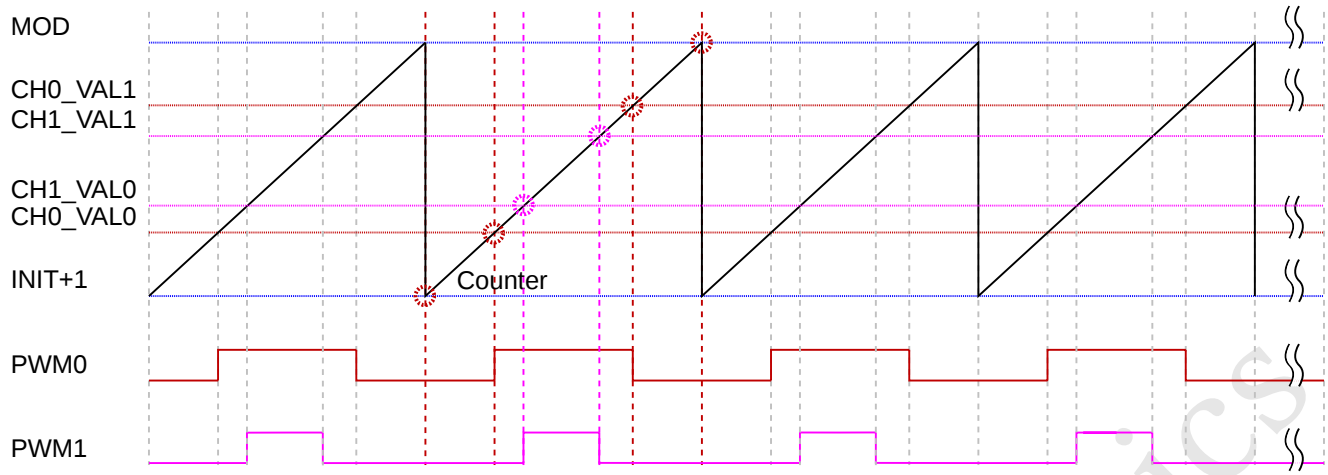


Figure 30.8: eTMR Independent PWM

30.4.4.8 Complementary Mode

Each channel pair of the eTMR can output complementary PWM, which is the complementary mode. Channel 0/1 output complementary PWM0/1 as shown in Figure eTMR Complementary PWM.

When enabling complementary mode, the channel n+1 output PWM is complementary with the channel n output. The CH_VAL0[VAL0] and CH_VAL1[VAL1] of channel n+1 don't work in complementary mode.

Complementary PWM is selected when:

- CHx_CTRL[CHMODE]=0x1
- CTRL[COMPLxx]=1

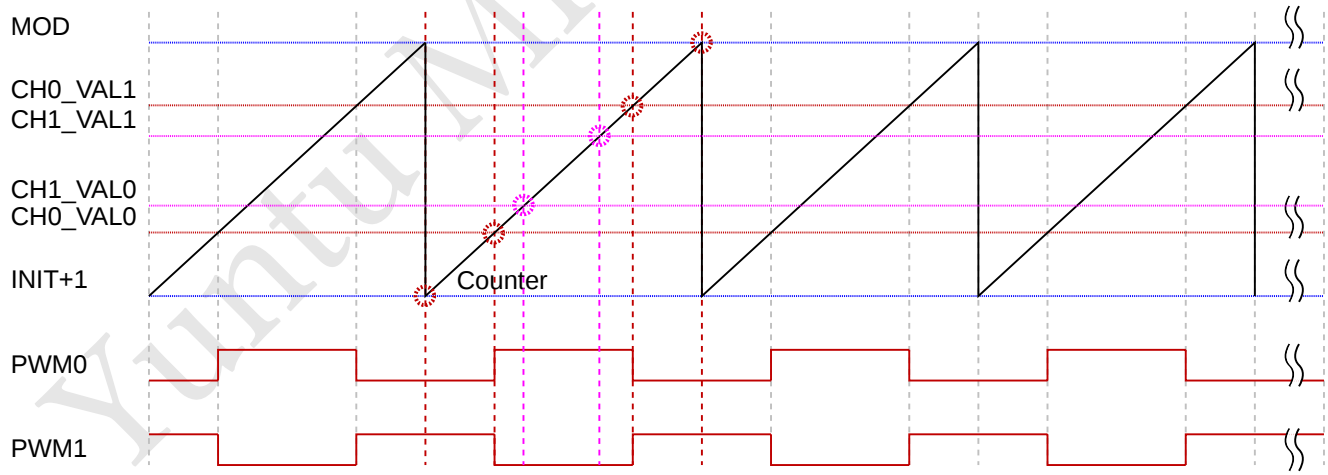


Figure 30.9: eTMR Complementary PWM

30.4.4.9 Complementary Mode with Dead-time Insertion

All channel pairs support dead-time insertion to stagger the edges of PWM in complementary mode.

Each channel supports dead-time insertion independently by CH_DT[DTVAL]. Dead-time value is CH_DT[DTVAL] * (bus clock period). Writing value 0 to CH_DT[DTVAL] to disable the dead-time

Yuntu Microelectronics

insertion. Channel 0/1 output complementary PWM0/1 as shown in Figure eTMR Complementary PWM Dead-time Insertion.

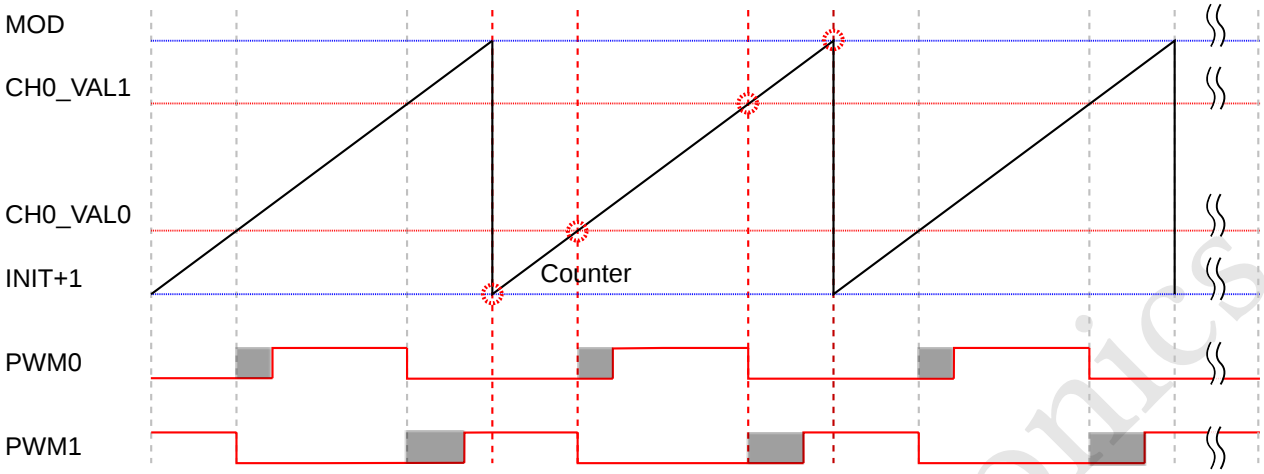


Figure 30.10: eTMR Complementary PWM Deadtime Insertion

30.4.4.10 Fault Detection and Protection

Each eTMR has four independent fault sources from TMU or pad. If a fault event is detected, the PWM output will force to the safe value. As shown in Figure eTMR Fault Block Diagram.

There are four available fault sources, which is fault 0, fault 1, fault 2, and fault 3. Fault 0, fault 1 and fault 2 are optional from TMU or pad, fault 3 is only from pad. Fault function is disabled when FAULT[FxEN]=0, please configure the specific function mode first before enabling it.

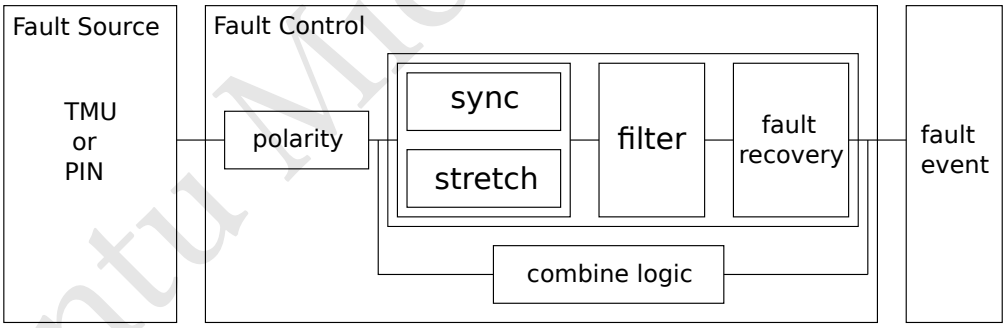


Figure 30.11: eTMR Fault Block Diagram

30.4.4.10.1 Safe Value

The fault function is available for all channels, and the fault event can force all channels to output the safe value. The safe value of each channel can be configured independently by the CHFV register. The safe value includes value 0, value 1, and tri-state.

Yuntu Microelectronics

30.4.4.10.2 Fault Source Selection

The fault source is defined by the CIM register, CIM_ETMROPT0[ETMR0_FAULT_SEL] defines the source of eTMR0, CIM_ETMROPT0[ETMR1_FAULT_SEL] defines the source of eTMR1, CIM_ETMROPT0[ETMR2_FAULT_SEL] defines the source of eTMR2, CIM_ETMROPT0[ETMR3_FAULT_SEL] defines the source of eTMR3. See eTMRx fault selection for details.

30.4.4.10.3 Fault Input Polarity

The polarity of the fault input is defined by FAULT[FxPOL]. When FAULT[FxPOL]=0, fault input high is valid. When FAULT[FxPOL]=1, fault input low is valid.

30.4.4.10.4 Fault Filter

The fault inputs are filtered by the digital filter. The valid input pulse width is defined by FAULT[FFCNT] and FAULT[FFPER], the input will be ignored when its valid width is less than FAULT[FFCNT] × FAULT[FFPER].

30.4.4.10.5 Input Stretch

To generate a fault event successfully, the fault input pulse should be bigger than one bus clock cycle. If the input signal is less than one bus clock cycle from other modules, the stretch function can stretch the input signal and generate a fault event. Set FAULT[STRETCH]=1 to enable the stretch function. The input signal after being stretched is at least two bus clock cycles. As shown in Figure eTMR Fault Stretch.

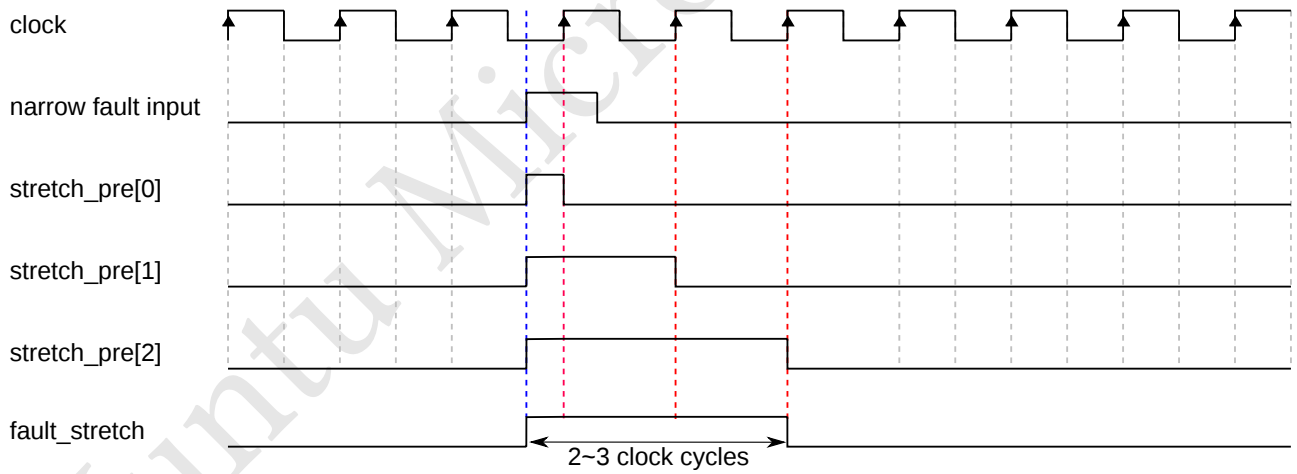


Figure 30.12: eTMR Fault Stretch

30.4.4.10.6 Combinational fault

When there is no bus clock, the fault event can only be generated by the combinational logic. Set FAULT[FCOMB]=1 to enable the combinational fault input. Any valid input signal results in a fault event when the combinational fault is enabled. The fault event generated by the combinational logic will keep until the bus clock recovery.

NOTE:

The fault flag won't be set while a fault event is generated by the combinational logic without bus clock.

Yuntu Microelectronics

30.4.4.10.7 Fault Recovery

The fault event clear opportunity is selectable after removing the fault input. The fault event clear opportunity can be immediate or when the counter reach specific values (MID or MOD), and it can also keep until disabling the fault. Configure FAULT[RCVRY] to define the fault event clear opportunity.

30.4.4.10.8 Fault Flag Clear

The fault flag (STS[FxF]) will be set when a fault event occurs. The fault flag and the fault recovery work together after the fault input is removed. When the fault event clear opportunity comes, it can be selected that the fault event is cleared immediately or cleared after clearing the fault flag. Configure FAULT[FAUTO] to choose the fault flag clearing mode.

30.4.4.11 Mask Control

The mask function is available for all channels in PWM mode. It's able to mask the PWM output by value 0 or value 1, and it can also disable the PWM output by masking it to tri-state. This function is configured by the CHMASK register. See Channel Output Mask Value Register for details.

30.4.5 Output Compare Mode

When CHx_CTRL[CHMODE]=0x2, the channel x is in output compare mode. In this mode, each channel can set, clear, or toggle the output when the counter reaches the specific value defined by CH_VAL0[VAL0] and CH_VAL1[VAL1]. Configure CH_CTRL[VAL0CMP] and CH_CTRL[VAL1CMP] to select the action of the corresponding channel when the counter reaches the CH_VAL0[VAL0] and CH_VAL1[VAL1]. As shown in Figure eTMR Output Compare. The channel flag (STS[CHxF]) will be set when a compare event occurs.

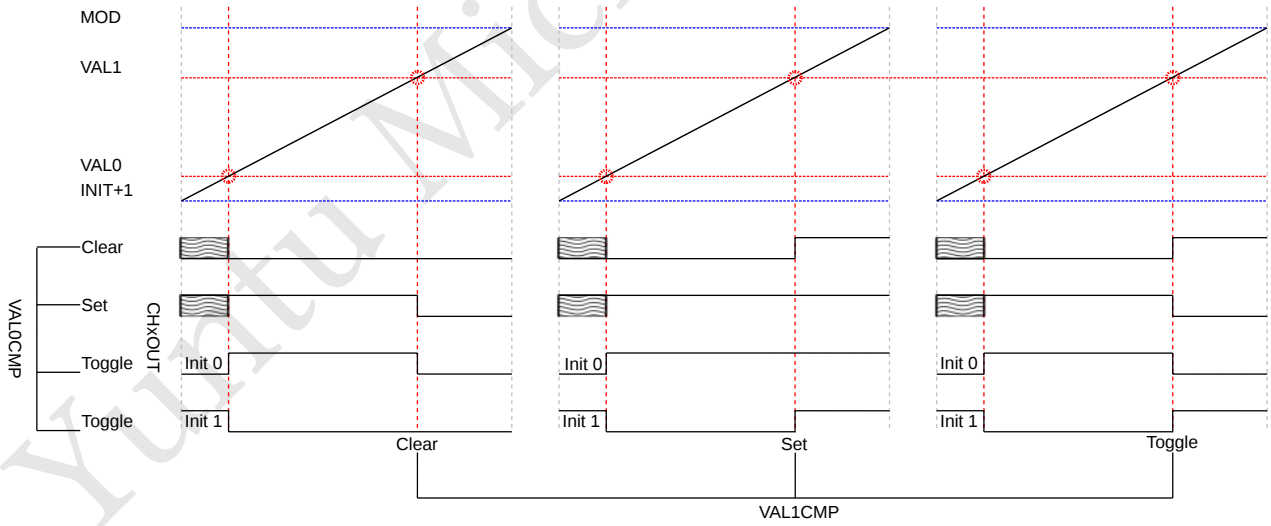


Figure 30.13: eTMR Output Compare

30.4.6 Input Capture Mode

When CHx_CTRL[CHMODE]=0x3, the channel x is in input capture mode. In this mode, each channel can detect the input signal change defined by CH_CTRL[CAPEDGE]. When the capture event occurs, the current counter value is saved to the CH_CVAL register. As shown in Figure eTMR Input Capture. The channel flag (STS[CHxF]) will be set when a capture event occurs.

Yuntu Microelectronics

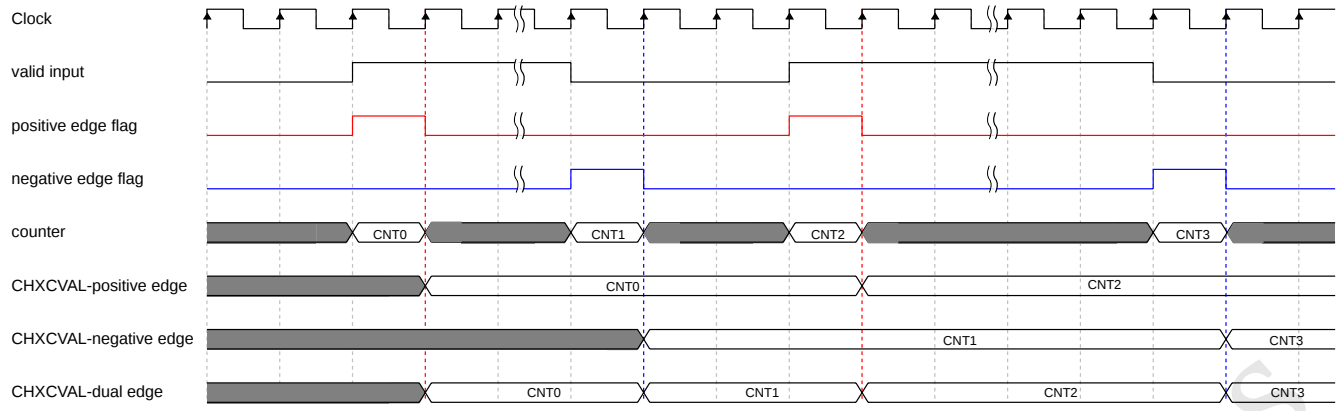


Figure 30.14: eTMR Input Capture

30.4.6.1 Channel Filter

The channel inputs are filtered by the digital filter. The valid input pulse width is defined by CH_CTRL[CFCNT] and CH_CTRL[CFPER], the input will be ignored when its valid width is less than CH_CTRL[CFCNT] × CH_CTRL[CFPER].

30.4.6.2 Capture Pulse Width

In input capture mode, eTMR supports positive and negative pulse width measure. The counter needs to start counting from 0(INIT[INIT]==0). When input capture event occurs, get capture status by reading STS[CHxNPF] and STS[CHxPPF] and read width data. Registers CH_PPCNTN[PPCNTN] and CH_PPCNTV[PPCNTV] contain positive pulse width data. Registers CH_NPCNTN[NPCNTN] and CH_NPCNTV[NPCNTV] contain negative pulse width data. As shown in Figure eTMR Pulse Measure.

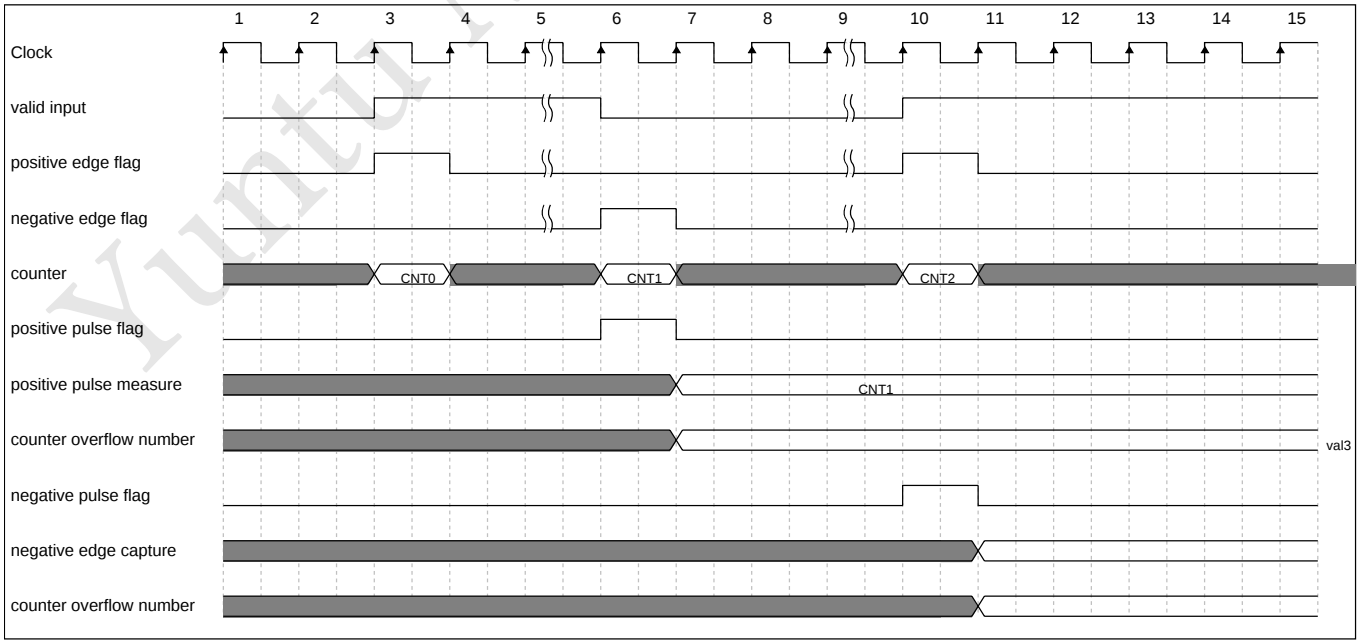


Figure 30.15: eTMR Pulse Measure

Yuntu Microelectronics

30.4.6.3 Capture Test

When CTRL[CAPTST]=1 and all channels are in input capture mode, the capture test mode is enabled. In this mode, the counter stops and can write a new value to the counter. After the capture mode is disabled, the counter will continue counting based on the new value. When writing the counter in capture test mode, the channel flag (STS[CHxF]) will be set, and the current counter value is saved to the CVAL register.

30.4.7 Output Polarity

In PWM mode or output compare mode, the channel output is controlled by the polarity defined by CH_CTRL[CHPOL].

- When CH_CTRL[CHPOL]=0, the corresponding channel outputs the raw signal
- When CH_CTRL[CHPOL]=1, the corresponding channel outputs the inverted signal.

30.4.8 Channels Initialization

In PWM mode or output compare mode, when the eTMR is disabled, the channels output the initial value defined by the CHINITV register. When the eTMR is enabled, write value 1 to CHINIT[CHxINIT] also results in the corresponding channel outputting the initial value. (x=0-7)

30.4.9 Loading Mechanism

Some registers are designed with buffer registers, which are CHMASK, INIT, MID, MOD, CH_VAL0, and CH_VAL1. When the eTMR is enabled, these registers' value is locked and they can only change at the appropriate opportunity. The loading mechanism is applied to manage the update of these special registers. As shown in Figure eTMR Loading Mechanism.

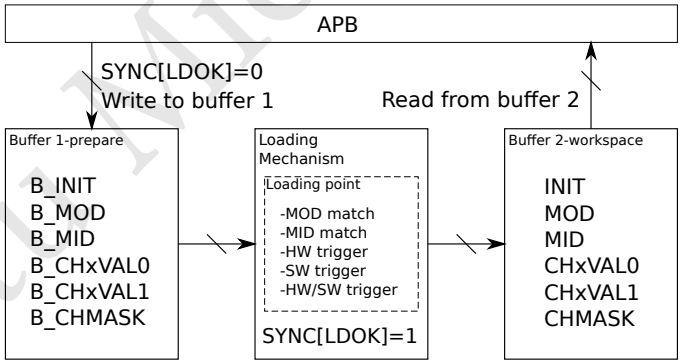


Figure 30.16: eTMR Loading Mechanism

Registers loading configuration:

- Select loading opportunity
- Configure trigger if need
- Configure loading frequency if need

Yuntu Microelectronics

- Write buffer registers
- Set SYNC[LDOK]=1

The reload flag (STS[RF]) will be set if registers load successfully.

30.4.9.1 General Registers Loading

The general registers include INIT, MOD, MID, CH_VAL0 and CH_VAL1, and they load with respective buffer registers at the same time point. These registers can update when the counter matches with MOD[MOD] or MID[MID]. As shown in Figure eTMR Loading Mechanism Timing. In addition, These registers can update when a loading trigger occurs. As shown in Figure eTMR Loading by Trigger Timing. Configure SYNC[REGLD] to select the loading opportunity of the general registers. Set CH_CTRL[LDEN]=1 to enable the loading of CH_VAL0 and CH_VAL1.

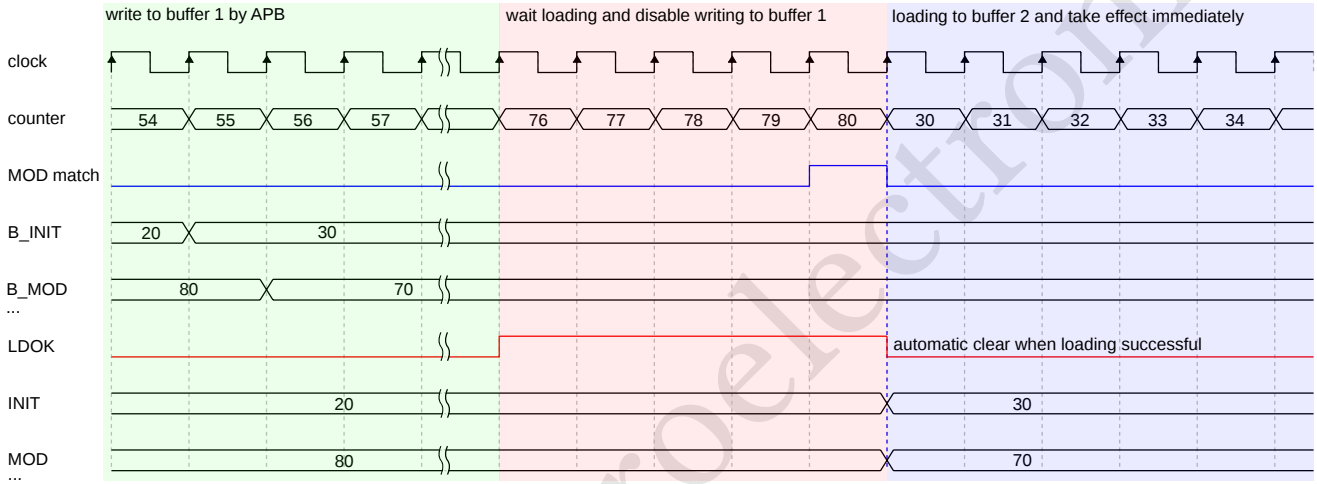


Figure 30.17: eTMR Loading Mechanism Timing

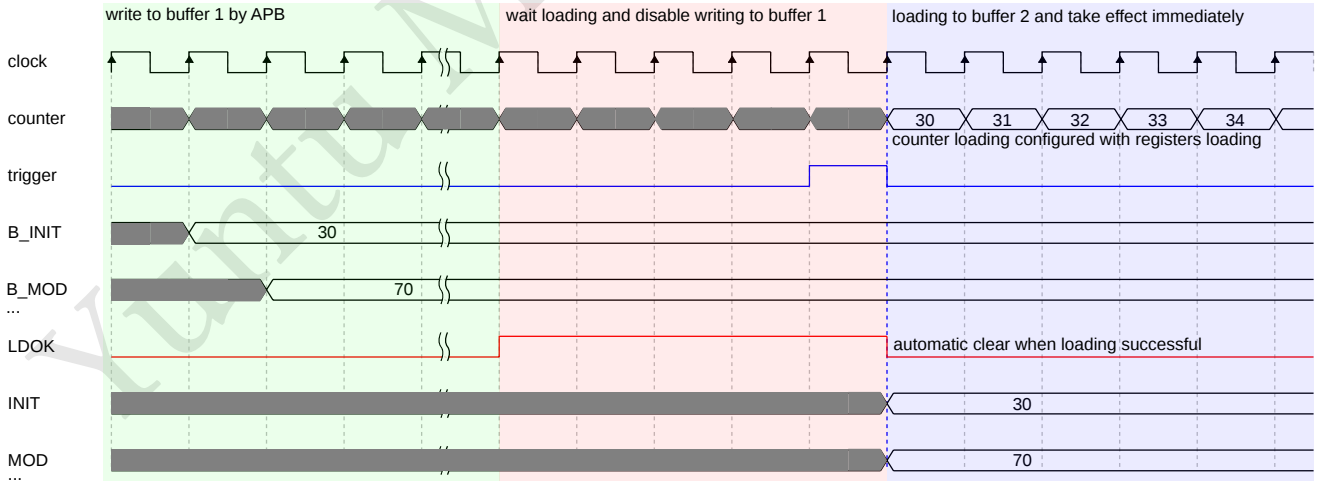


Figure 30.18: eTMR Loading by Trigger Timing

30.4.9.2 Special Registers Loading

The special register refers to CHMASK, and it updates at the configurable time point. The CHMASK register can update when the counter matches with MOD[MOD], MID[MID] or a loading trigger occurs,

Yuntu Microelectronics

and it can also update together with the general registers. Configure SYNC[MASKLD] to select the loading opportunity of the CHMASK register.

30.4.9.3 Counter Loading

The counter can update when the counter matches with MID[MID] or a loading trigger occurs, and it can also update together with the general registers. Configure SYNC[CNTLD] to select the loading opportunity of the CNT register.

NOTE: The CNT register always load value from INIT register when a loading event occurs.

30.4.9.4 Loading trigger

There are three hardware triggers and one software trigger used for registers' loading. Configure SYNC[TRIGSx] to select the trigger source which is used for registers' loading.

The hardware triggers are from the TMU (trigger 0), CIM (trigger 1), and pad (trigger 2), and they are enabled by setting SYNC[TRIGxEN]=1. Any valid hardware trigger x will generate a trigger event (x=0,1,2).

Writing value 1 to SYNC[SWTRIG] can generate a software trigger.

30.4.9.5 Loading Frequency

The loading frequency is applied for executing registers' loading when the loading opportunity comes at specific times. If the loading opportunity counts less than SYNC[LDFRQ], the registers' loading won't be executed.

30.4.10 Dithering

Dithering is used to achieve fine resolution on the generated PWM signals. The dither accuracy is 1/32 counter clock source before clock division. This module contains three kinds of dithering by CTRL[DITHMD] configuration:

- PWM period dithering
- PWM edge dithering
- PWM period and edge dithering

When CTRL[DITHMD]=0x1, period dither for more resolution by DITHMOD[MODINT] and DITHMOD[MODFRAC] configuration. The integer part from DITHMOD[MODINT] needs less than CTRL[CLKPRS] which guaranteed to be usable even in the case of clock division. The fractional part is from DITHMOD[MODFRAC].

When CTRL[DITHMD]=0x2, edge dither for more resolution by CH_DITHVAL1[VAL1INT] and CH_DITHVAL1[VAL1FRAC] configuration. The integer part from CH_DITHVAL1[VAL1INT] needs less than CTRL[CLKPRS] which guaranteed to be usable even in the case of clock division. The fractional part is from CH_DITHVAL1[VAL1FRAC].

When CTRL[DITHMD]=0x3, period and edge dither for more resolution by the combination of DITHMOD[MODINT], DITHMOD[MODFRAC], CH_DITHVAL1[VAL1INT] and CH_DITHVAL1[VAL1FRAC] configurations.

Yuntu Microelectronics

30.4.11 Quadrature Decoder

The eTMR1 supports the quadrature decoder mode, the mode is enabled when QDCTRL[QDEN]=1. In this mode, the independent 16-bit counter will increase or decrease according to the input signals from the pad. As shown in Figure eTMR Quadrature Decoder. The STS[QDDIR] flag indicates the counting direction of the counter. The STS[QDTOF] flag will be set if the counter overflow occurs.

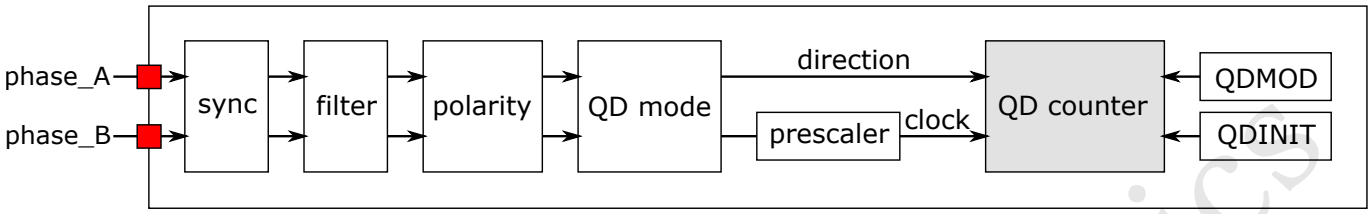


Figure 30.19: eTMR Quadrature Decoder

30.4.11.1 Decoder Prescaler

In the quadrature decoder mode, a 3-bit prescaler is available for the independent counter. The prescaler can be configured by QDCTRL[QDCLKPRS].

30.4.11.2 Count and Direction Decoder Mode

Configure QDCTRL[QDMODE]=2 or QDCTRL[QDMODE]=3 to enable the direction decoding mode. In this mode, one of the input indicates the counting direction, the other input defines the counting rate. As shown in Figure eTMR Quadrature Decoder Count and Direction Mode.

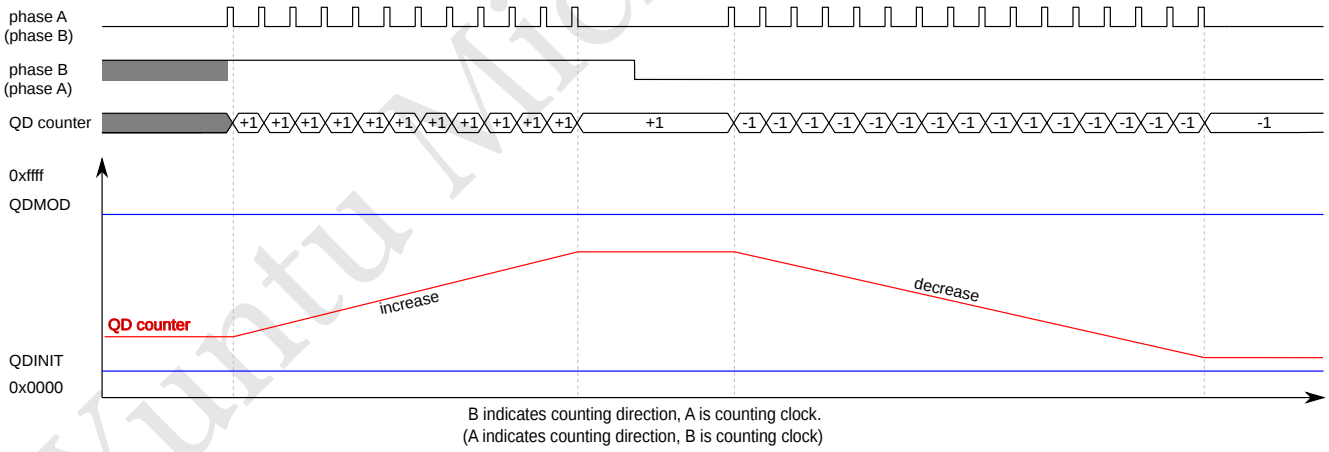


Figure 30.20: eTMR Quadrature Decoder Count and Direction Mode

30.4.11.3 Phase A and phase B Decoder Mode

Configure QDCTRL[QDMODE]=0 or QDCTRL[QDMODE]=1 to enable the phase decoding mode. In this mode, the relationship between two input signals indicates the counting direction, the two input signals define the counting rate. As shown in Figure eTMR Quadrature Decoder PhaseA and PhaseB Mode.

Yuntu Microelectronics

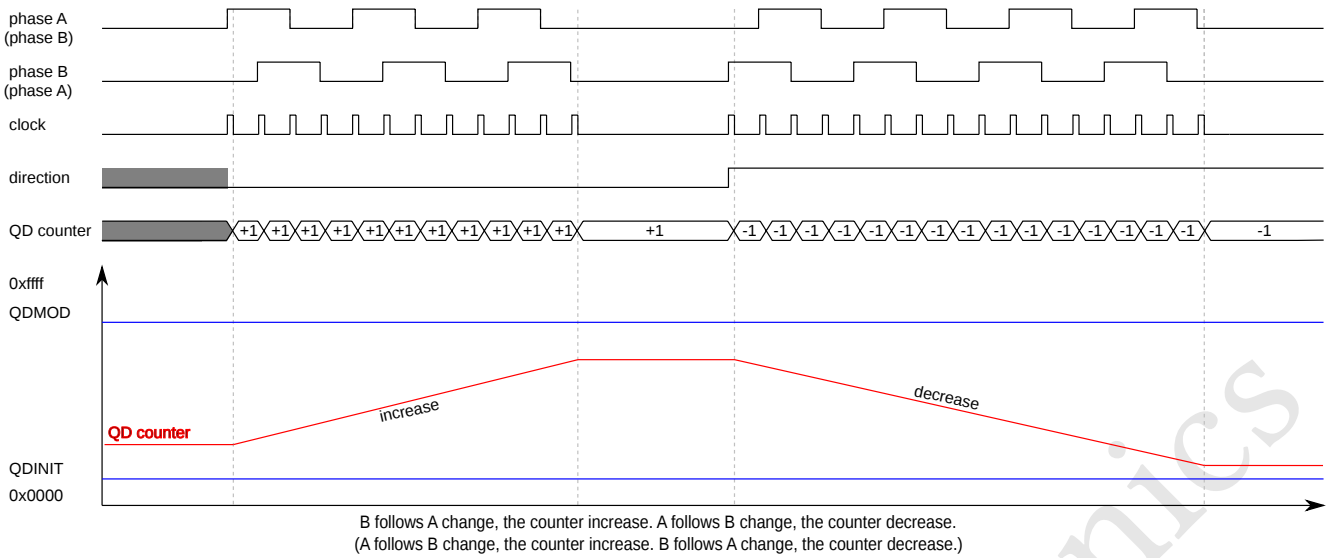


Figure 30.21: eTMR Quadrature Decoder PhaseA and PhaseB Mode

30.4.11.4 Decoder Input Polarity

The polarity of the decoder input is defined by QDCTRL[PHxPOL]. When QDCTRL[PHxPOL]=0, the decoder input is the raw input signal. When QDCTRL[PHxPOL]=1, the decoder input is the inverted input signal.

30.4.11.5 Decoder Filter

The quadrature decoder inputs are filtered by the digital filter. The valid input pulse width is defined by QDCTRL[QDFCNT] and QDCTRL[QDFPER], the input will be ignored when its valid width is less than QDCTRL[QDFCNT] × QDCTRL[QDFPER].

30.4.11.6 Decoder Counter Overflow

Quadrature Decoder counter contains up counting overflow and down counting overflow. As shown in Figure eTMR Quadrature Decoder Counter Overflow.

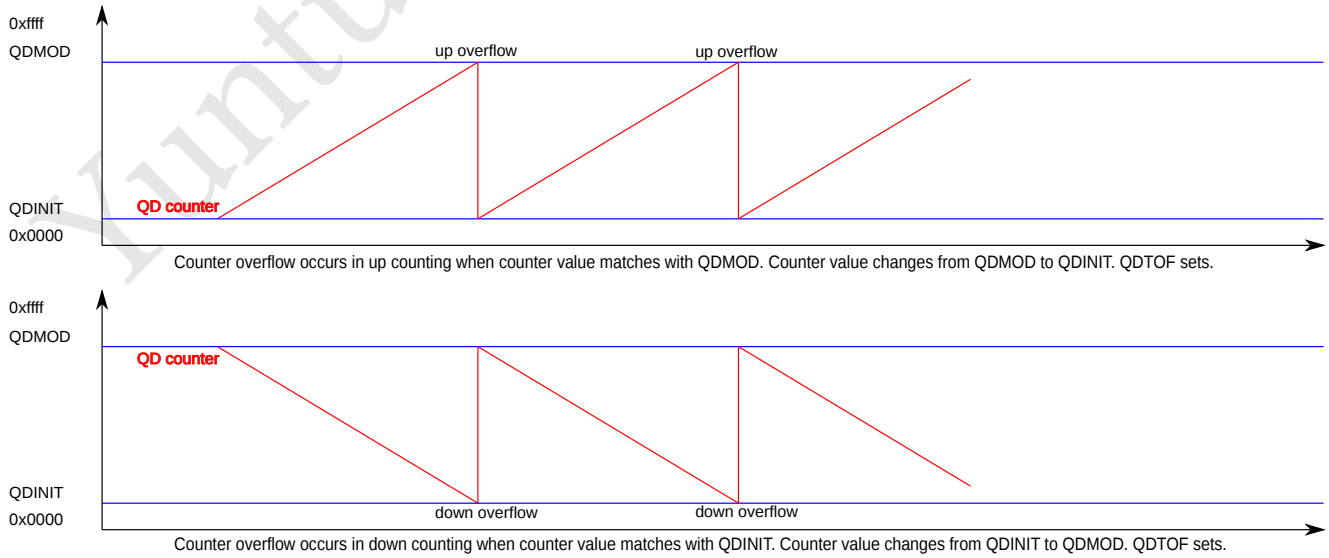


Figure 30.22: eTMR Quadrature Decoder Counter Overflow

Yuntu Microelectronics

30.4.12 Filter

Filter used for input capture mode, fault input and phase A/phase B in quadrature decoder mode. As shown in Figure eTMR Filter Timing. Filter clock division and number of samples can be configured.

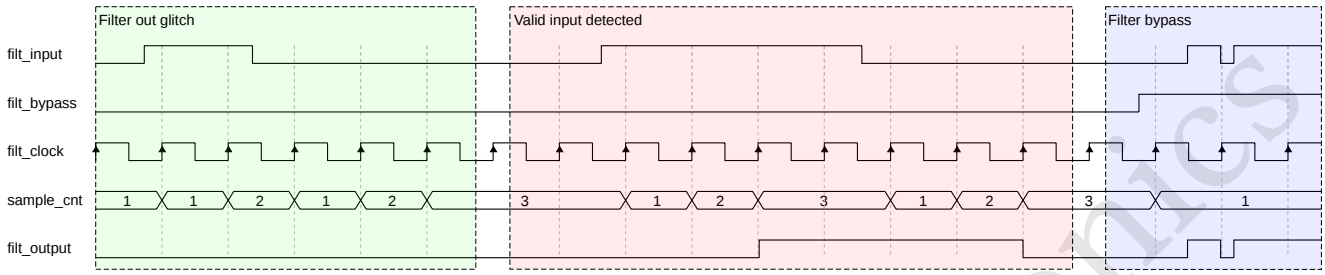


Figure 30.23: eTMR Filter Timing

30.4.13 Output Trigger

The eTMR can output triggers that send to the TMU module. The output trigger can be width fixed pulse or width adjustable pulse.

30.4.13.1 Width Fixed Pulse

Configure OTRIG[TSRC]=0 to enable the generating of width fixed pulse. The width fixed pulse is generated by counter matching, and there are five available match points: INIT, MID, MOD, VAL0, and VAL1. The generated pulses are all one bus clock cycle width.

- When the counter reaches INIT[INIT] and OTRIG[INITTEN]=1, the INIT match pulse generates;
- When the counter reaches MOD[MOD] and OTRIG[MODTEN]=1, the MOD match pulse generates;
- When the counter reaches MID[MID] and OTRIG[MIDTEN]=1, the MID match pulse generates;
- When the counter reaches CH_VAL0[VAL0] and CH_CTRL[VAL0TEN]=1, the VAL0 match pulse generates;
- When the counter reaches CH_VAL1[VAL1] and CH_CTRL[VAL1TEN]=1, the VAL1 match pulse generates.

As shown in Figure eTMR Output Trigger.

Yuntu Microelectronics

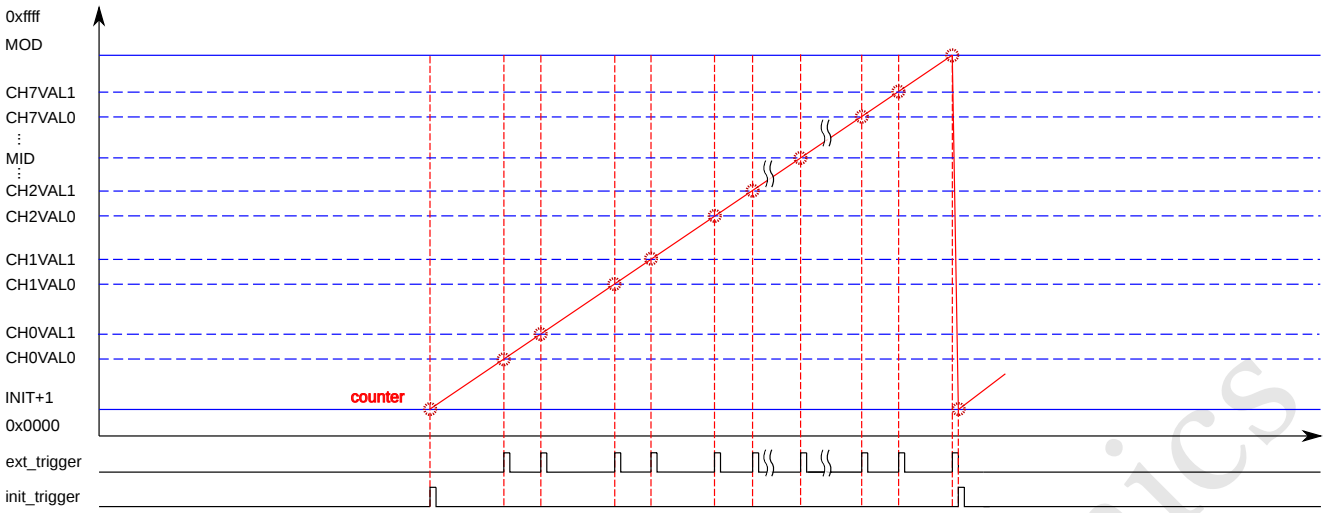


Figure 30.24: eTMR Output Trigger

30.4.13.2 Width Adjustable Pulse

Configure OTRIG[TSRC]=1 to enable the generating of width adjustable pulse. The width adjustable pulse is generated by channel in PWM mode, the PWM output can be used as the output trigger. Configure OTRIG[PULSESRC] to select the PWM source.

30.4.13.3 Output Frequency

The output frequency is applied for outputting pulse when the output opportunity comes at specific times. If the output opportunity counts less than OTRIG[TFRQ], the pulse won't be generate. The output frequency is only used to generate width fixed pulse.

30.4.14 Input Capture Source

The eTMR1 channel 0 and eTMR2 channel 0 input capture sources are selected by CIM_ETMROPT1[ETMRxCH0SEL]. Their input capture signal can be from the ACMP output. See ETMR1 Channel 0 Select and ETMR2 Channel 0 Select for details.

30.4.15 Hall Sensor Support

To simplify the calculations required by the CPU on each hall sensor's input, each eTMRx channel 0-3 supports hall sensor defined in CIM.

30.4.16 Modulation Implementation

The eTMR0 and supports channel outputs modulated(two channels '&' logic) by eTMR1 channel 1 output. The eTMR3 supports channel outputs modulated by eTMR2 channel 1 output. Any of the 8 channels of eTMR0 and eTMR3 can be configured to support this modulation function. This function is enabled by configuring CIM_ETMROPT1[ETMR0_OUTSEL] and CIM_ETMROPT1[ETMR3_OUTSEL]. See ETMR0 Output Modulation Select and ETMR3 Output Modulation Select for details.

30.4.17 Power Mode

The eTMRs can work in Active mode and Sleep mode.

Yuntu Microelectronics

30.4.18 Interrupts

The timeout interrupt and channel interrupt are available in all eTMRs. The timeout interrupt is generated by STS[TOF] when INTE[TOIE]=1. Write value 1 to STS[TOF] to clear the timeout interrupt. The channel interrupt is generated by STS[CHxF] when INTE[CHxIE]=1. Write value 1 to STS[CHxF] to clear the channel interrupt.

The fault interrupt is available in all eTMRs. The fault interrupt is generated by STS[FxF] when INTE[FIE]=1. Write value 1 to STS[FxF] to clear the fault interrupt.

The quadrature decoder counter timeout interrupt is only available in eTMR1 and eTMR2. The quadrature decoder counter timeout interrupt is generated by STS[QDTOF] when INTE[QDTOIE]=1. Write value 1 to STS[QDTOF] to clear the quadrature decoder counter timeout interrupt.

30.4.19 DMA

All eTMRs support DMA, this function is enabled by setting CH_CTRL[DMAEN]=1. The DMA request is generated when STS[CHxF]=1. The DMA request is cleared by clearing STS[CHxF]. The DMA priority is higher than the interrupt.

30.4.20 Reset

During a reset, the eTMRs are in the default mode and disabled.

30.5 Glossary

eTMR	Enhanced Timer
PWM	Pulse Width Modulation
TMU	Trigger Multiplexer
GTB	Global Time Base
DMA	Direct Memory Access
QD	Quadrature Decoder

Yuntu Microelectronics

Chapter 31

Multiple Pulse Width Modulation (MPWM)

31.1 Chip-specific Information

31.1.1 Instantiation Information

YTM32B1HA0x contains 3 MPWM instances.

Table 31.1: MPWM Instances and Features

Instances	Channels
MPWM0	16
MPWM1	16
MPWM2	16

31.1.2 Signal Description

Table 31.2: MPWM Signal Table

Signal Name	Description	Direction
MPWMx_CH0	MPWMx channel 0	I/O
MPWMx_CH1	MPWMx channel 1	I/O
MPWMx_CH2	MPWMx channel 2	I/O
MPWMx_CH3	MPWMx channel 3	I/O
MPWMx_CH4	MPWMx channel 4	I/O
MPWMx_CH5	MPWMx channel 5	I/O
MPWMx_CH6	MPWMx channel 6	I/O
MPWMx_CH7	MPWMx channel 7	I/O
MPWMx_CH8	MPWMx channel 8	I/O
MPWMx_CH9	MPWMx channel 9	I/O
MPWMx_CH10	MPWMx channel 10	I/O
MPWMx_CH11	MPWMx channel 11	I/O
MPWMx_CH12	MPWMx channel 12	I/O
MPWMx_CH13	MPWMx channel 13	I/O
MPWMx_CH14	MPWMx channel 14	I/O
MPWMx_CH15	MPWMx channel 15	I/O

Yuntu Microelectronics

31.1.3 Clock Description

Table 31.3: MPWM Clock Table

Clock Name	Description
Bus clock	Bus clock used for register access and function logic.

31.1.4 DMA Description

Table 31.4: MPWM DMA Table

Instances	No.	Request Source	Description
MPWM0	108	Compare event or capture event	channels 0-15 compare event or capture event generates DMA request
MPWM1	109	Compare event or capture event	channels 0-15 compare event or capture event generates DMA request
MPWM2	110	Compare event or capture event	channels 0-15 compare event or capture event generates DMA request

31.1.5 Interrupt Description

Table 31.5: MPWM Interrupt Table

Vector	IRQn	Request Source	Description	Wakeup
239	223	MPWM0 channels 0~7 shared interrupt	The interrupt can be generated when MPWM0 channels 0~7 timers overflow event or channels event occurs.	Y
240	224	MPWM0 channels 8~15 shared interrupt	The interrupt can be generated when MPWM0 channels 8~15 timers overflow event or channels event occurs.	Y
241	225	MPWM1 channels 0~7 shared interrupt	The interrupt can be generated when MPWM1 channels 0~7 timers overflow event or channels event occurs.	Y
242	226	MPWM1 channels 8~15 shared interrupt	The interrupt can be generated when MPWM1 channels 8~15 timers overflow event or channels event occurs.	Y
243	227	MPWM2 channels 0~7 shared interrupt	The interrupt can be generated when MPWM2 channels 0~7 timers overflow event or channels event occurs.	Y
244	228	MPWM2 channels 8~15 shared interrupt	The interrupt can be generated when MPWM2 channels 8~15 timers overflow event or channels event occurs.	Y

Yuntu Microelectronics

31.1.6 Power mode Description

Table 31.6: MPWM Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Keeps working
Deepsleep	Disable
Standby	Disable
Powerdown	Disable

31.2 Introduction

31.2.1 Overview

The MPWM contains a variety of functions, such as a common timer, PWM output, input capture, output compare, pulse count and so on.

31.2.2 Features

This chip contains 3 MPWMs(MPWM0, MPWM1, MPWM2). The features are listed below:

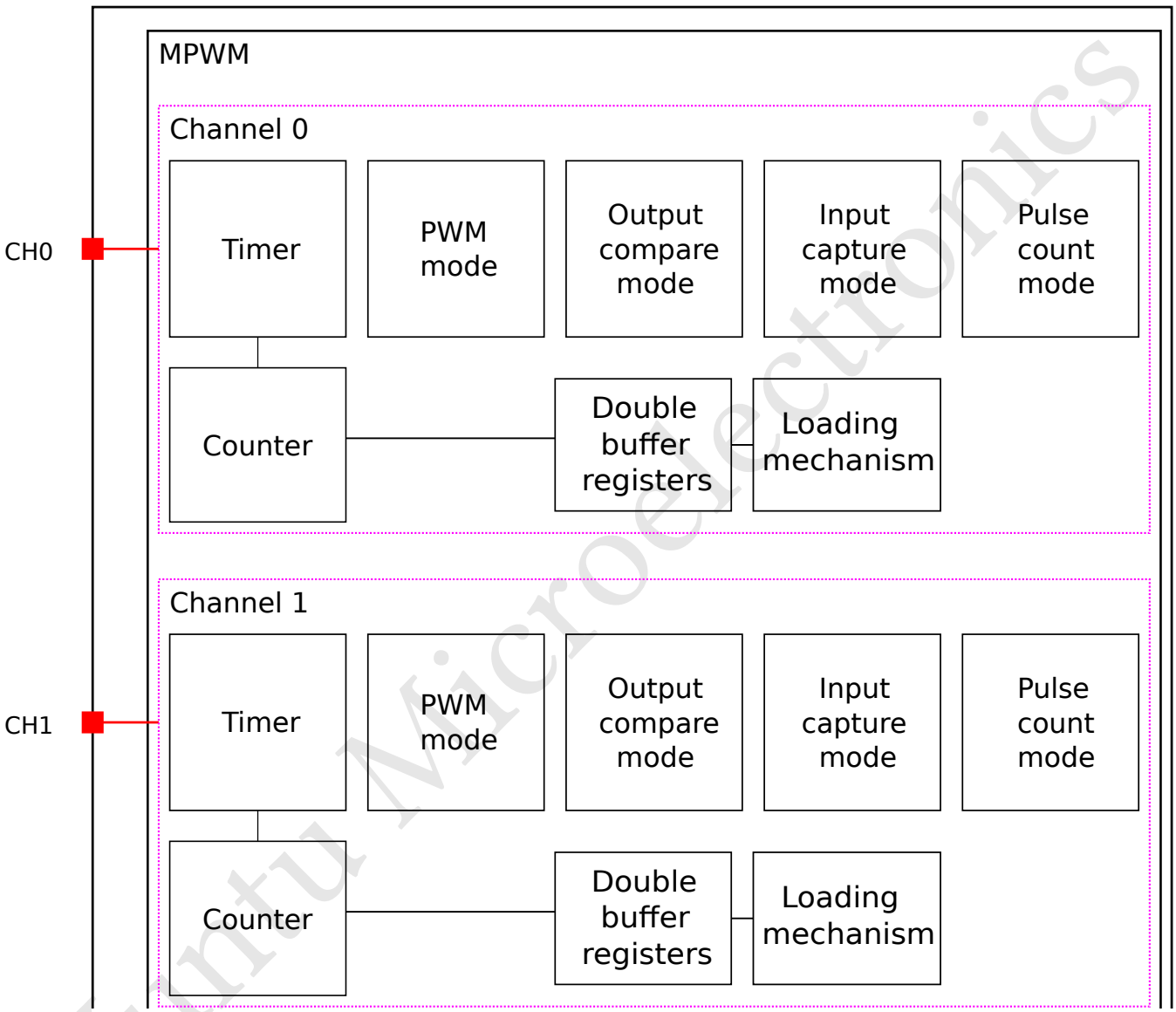
- Each MPWM supports 16 channels
- Each channel contains a 16-bit counter independently
  - Configurable comparison value
  - Configurable period value
- Support clock prescaler(1, 2, 4, 8, 16, 32, 64, 128) independently
- Counting modes
  - Continuous mode
  - One-shot mode
- Support counter starts counting by hardware trigger in one-shot mode
- Support four channel modes
  - Common timer
  - PWM mode
    - Independent mode for each channel
    - Support edge-aligned PWM
    - Relevant registers have double buffer registers and support loading mechanism
    - Write operations to double buffer registers take effect immediately
  - Output Compare mode
    - The output can be configured to set, clear or toggle on the match point
  - Input Capture mode
    - Support rising edge capture
    - Support falling edge capture
    - Support any edge capture
  - Pulse count mode
- Support generating triggers on match point
- Polarity control is available for each channel
- Support several interrupts
  - Channel event interrupt

Yuntu Microelectronics

  - Timer overflow event interrupt
- Support DMA
- Support counter running under debug mode

31.2.3 Block diagram

The following figure shows the MPWM block diagram.



Yuntu Microelectronics

Figure 31.1: MPWM Block Diagram

31.3 Register Definition

31.3.1 MPWM Memory Map

MPWM0 base address 40144000h
MPWM1 base address 40148000h
MPWM2 base address 4014c000h

Table 31.7: MPWM Memory Map

Offset	Register	Width	Access	Reset Value
0h	IO Status Register (IOSTS)	32	RW	00000000h
4h	Status 0 Register (STS0)	32	RW	00000000h
8h	Status 1 Register (STS1)	32	RW	00000000h
ch	Interrupt Enable 0 Register (INTE0)	32	RW	00000000h
10h	Interrupt Enable 1 Register (INTE1)	32	RW	00000000h
14h	Control Register (CTRL)	32	RW	00000000h
18h	Software Trigger Register (SWT)	32	RW	00000000h
1ch	Channel Initial Register (INIT)	32	RW	00000000h
20h	Channel Polarity Register (POL)	32	RW	00000000h
40h	Channel Configuration Register (CH0_CFG)	32	RW	00000000h
44h	Counter Status Register (CH0_CNT)	32	RO	00000000h
48h	Capture Status Register (CH0_CAP)	32	RW	00000000h
50h	Period Register (CH0_PERIOD)	32	RW	00000000h
54h	Compare Register (CH0_CMP)	32	RW	00000000h
60h	Channel Configuration Register (CH1_CFG)	32	RW	00000000h
64h	Counter Status Register (CH1_CNT)	32	RO	00000000h
68h	Capture Status Register (CH1_CAP)	32	RW	00000000h
70h	Period Register (CH1_PERIOD)	32	RW	00000000h
74h	Compare Register (CH1_CMP)	32	RW	00000000h
80h	Channel Configuration Register (CH2_CFG)	32	RW	00000000h
84h	Counter Status Register (CH2_CNT)	32	RO	00000000h
88h	Capture Status Register (CH2_CAP)	32	RW	00000000h
90h	Period Register (CH2_PERIOD)	32	RW	00000000h
94h	Compare Register (CH2_CMP)	32	RW	00000000h
a0h	Channel Configuration Register (CH3_CFG)	32	RW	00000000h
a4h	Counter Status Register (CH3_CNT)	32	RO	00000000h
a8h	Capture Status Register (CH3_CAP)	32	RW	00000000h
b0h	Period Register (CH3_PERIOD)	32	RW	00000000h
b4h	Compare Register (CH3_CMP)	32	RW	00000000h
c0h	Channel Configuration Register (CH4_CFG)	32	RW	00000000h
c4h	Counter Status Register (CH4_CNT)	32	RO	00000000h
c8h	Capture Status Register (CH4_CAP)	32	RW	00000000h
d0h	Period Register (CH4_PERIOD)	32	RW	00000000h
d4h	Compare Register (CH4_CMP)	32	RW	00000000h
e0h	Channel Configuration Register (CH5_CFG)	32	RW	00000000h
e4h	Counter Status Register (CH5_CNT)	32	RO	00000000h
e8h	Capture Status Register (CH5_CAP)	32	RW	00000000h
f0h	Period Register (CH5_PERIOD)	32	RW	00000000h
f4h	Compare Register (CH5_CMP)	32	RW	00000000h
100h	Channel Configuration Register (CH6_CFG)	32	RW	00000000h
104h	Counter Status Register (CH6_CNT)	32	RO	00000000h
108h	Capture Status Register (CH6_CAP)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
110h	Period Register (CH6_PERIOD)	32	RW	00000000h
114h	Compare Register (CH6_CMP)	32	RW	00000000h
120h	Channel Configuration Register (CH7_CFG)	32	RW	00000000h
124h	Counter Status Register (CH7_CNT)	32	RO	00000000h
128h	Capture Status Register (CH7_CAP)	32	RW	00000000h
130h	Period Register (CH7_PERIOD)	32	RW	00000000h
134h	Compare Register (CH7_CMP)	32	RW	00000000h
140h	Channel Configuration Register (CH8_CFG)	32	RW	00000000h
144h	Counter Status Register (CH8_CNT)	32	RO	00000000h
148h	Capture Status Register (CH8_CAP)	32	RW	00000000h
150h	Period Register (CH8_PERIOD)	32	RW	00000000h
154h	Compare Register (CH8_CMP)	32	RW	00000000h
160h	Channel Configuration Register (CH9_CFG)	32	RW	00000000h
164h	Counter Status Register (CH9_CNT)	32	RO	00000000h
168h	Capture Status Register (CH9_CAP)	32	RW	00000000h
170h	Period Register (CH9_PERIOD)	32	RW	00000000h
174h	Compare Register (CH9_CMP)	32	RW	00000000h
180h	Channel Configuration Register (CH10_CFG)	32	RW	00000000h
184h	Counter Status Register (CH10_CNT)	32	RO	00000000h
188h	Capture Status Register (CH10_CAP)	32	RW	00000000h
190h	Period Register (CH10_PERIOD)	32	RW	00000000h
194h	Compare Register (CH10_CMP)	32	RW	00000000h
1a0h	Channel Configuration Register (CH11_CFG)	32	RW	00000000h
1a4h	Counter Status Register (CH11_CNT)	32	RO	00000000h
1a8h	Capture Status Register (CH11_CAP)	32	RW	00000000h
1b0h	Period Register (CH11_PERIOD)	32	RW	00000000h
1b4h	Compare Register (CH11_CMP)	32	RW	00000000h
1c0h	Channel Configuration Register (CH12_CFG)	32	RW	00000000h
1c4h	Counter Status Register (CH12_CNT)	32	RO	00000000h
1c8h	Capture Status Register (CH12_CAP)	32	RW	00000000h
1d0h	Period Register (CH12_PERIOD)	32	RW	00000000h
1d4h	Compare Register (CH12_CMP)	32	RW	00000000h
1e0h	Channel Configuration Register (CH13_CFG)	32	RW	00000000h
1e4h	Counter Status Register (CH13_CNT)	32	RO	00000000h
1e8h	Capture Status Register (CH13_CAP)	32	RW	00000000h
1f0h	Period Register (CH13_PERIOD)	32	RW	00000000h
1f4h	Compare Register (CH13_CMP)	32	RW	00000000h
200h	Channel Configuration Register (CH14_CFG)	32	RW	00000000h
204h	Counter Status Register (CH14_CNT)	32	RO	00000000h
208h	Capture Status Register (CH14_CAP)	32	RW	00000000h
210h	Period Register (CH14_PERIOD)	32	RW	00000000h
214h	Compare Register (CH14_CMP)	32	RW	00000000h
220h	Channel Configuration Register (CH15_CFG)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
224h	Counter Status Register (CH15_CNT)	32	RO	00000000h
228h	Capture Status Register (CH15_CAP)	32	RW	00000000h
230h	Period Register (CH15_PERIOD)	32	RW	00000000h
234h	Compare Register (CH15_CMP)	32	RW	00000000h

31.3.1.1 MPWM IOSTS Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH15IO	CH14IO	CH13IO	CH12IO	CH11IO	CH10IO	CH9IO	CH8IO	CH7IO	CH6IO	CH5IO	CH4IO	CH3IO	CH2IO	CH1IO	CH0IO
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.8: MPWM IOSTS Register Description

Field	Function
15 CH15IO	**Channel 15 IO Status** Indicate channel 15 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
14 CH14IO	**Channel 14 IO Status** Indicate channel 14 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
13 CH13IO	**Channel 13 IO Status** Indicate channel 13 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
12	**Channel 12 IO Status**

Yuntu Microelectronics

Field	Function
CH12IO	Indicate channel 12 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
11 CH11IO	**Channel 11 IO Status** Indicate channel 11 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
10 CH10IO	**Channel 10 IO Status** Indicate channel 10 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
9 CH9IO	**Channel 9 IO Status** Indicate channel 9 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
8 CH8IO	**Channel 8 IO Status** Indicate channel 8 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
7 CH7IO	**Channel 7 IO Status** Indicate channel 7 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
6 CH6IO	**Channel 6 IO Status** Indicate channel 6 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
5 CH5IO	**Channel 5 IO Status** Indicate channel 5 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
4	**Channel 4 IO Status**

Yuntu Microelectronics

Field	Function
CH4IO	Indicate channel 4 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
3 CH3IO	**Channel 3 IO Status** Indicate channel 3 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
2 CH2IO	**Channel 2 IO Status** Indicate channel 2 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
1 CH1IO	**Channel 1 IO Status** Indicate channel 1 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.
0 CH0IO	**Channel 0 IO Status** Indicate channel 0 input/output status in pwm mode, compare mode and input capture mode. This bit is read only and the write operation is invalid. 0b - Channel status is 0. 1b - Channel status is 1.

31.3.1.2 MPWM STS0 Register

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH15TOF	CH14TOF	CH13TOF	CH12TOF	CH11TOF	CH10TOF	CH9TOF	CH8TOF	CH7TOF	CH6TOF	CH5TOF	CH4TOF	CH3TOF	CH2TOF	CH1TOF	CH0TOF
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 31.9: MPWM STS0 Register Description

Field	Function
15 CH15TOF	**Channel 15 Timer Overflow Flag** Indicate channel 15 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
14 CH14TOF	**Channel 14 Timer Overflow Flag** Indicate channel 14 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
13 CH13TOF	**Channel 13 Timer Overflow Flag** Indicate channel 13 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
12 CH12TOF	**Channel 12 Timer Overflow Flag** Indicate channel 12 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
11 CH11TOF	**Channel 11 Timer Overflow Flag** Indicate channel 11 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
10 CH10TOF	**Channel 10 Timer Overflow Flag** Indicate channel 10 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
9 CH9TOF	**Channel 9 Timer Overflow Flag** Indicate channel 9 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
8 CH8TOF	**Channel 8 Timer Overflow Flag** Indicate channel 8 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
7 CH7TOF	**Channel 7 Timer Overflow Flag** Indicate channel 7 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
6	**Channel 6 Timer Overflow Flag**

Yuntu Microelectronics

Field	Function
CH6TOF	Indicate channel 6 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
5 CH5TOF	**Channel 5 Timer Overflow Flag** Indicate channel 5 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
4 CH4TOF	**Channel 4 Timer Overflow Flag** Indicate channel 4 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
3 CH3TOF	**Channel 3 Timer Overflow Flag** Indicate channel 3 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
2 CH2TOF	**Channel 2 Timer Overflow Flag** Indicate channel 2 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
1 CH1TOF	**Channel 1 Timer Overflow Flag** Indicate channel 1 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.
0 CH0TOF	**Channel 0 Timer Overflow Flag** Indicate channel 0 timer overflow flag. Write 1 to clear the bit. 0b - No timer overflow event occurs. 1b - Timer overflow event occurs.

Yuntu Microelectronics

31.3.1.3 MPWM STS1 Register

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH15F	CH14F	CH13F	CH12F	CH11F	CH10F	CH9F	CH8F	CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.10: MPWM STS1 Register Description

Field	Function
15 CH15F	**Channel 15 Flag** Indicate channel 15 matching event flag in pwm mode and compare mode. Indicate channel 15 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
14 CH14F	**Channel 14 Flag** Indicate channel 14 matching event flag in pwm mode and compare mode. Indicate channel 14 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
13 CH13F	**Channel 13 Flag** Indicate channel 13 matching event flag in pwm mode and compare mode. Indicate channel 13 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
12 CH12F	**Channel 12 Flag** Indicate channel 12 matching event flag in pwm mode and compare mode. Indicate channel 12 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
11	**Channel 11 Flag**

Yuntu Microelectronics

Field	Function
CH11F	Indicate channel 11 matching event flag in pwm mode and compare mode. Indicate channel 11 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
10 CH10F	**Channel 10 Flag** Indicate channel 10 matching event flag in pwm mode and compare mode. Indicate channel 10 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
9 CH9F	**Channel 9 Flag** Indicate channel 9 matching event flag in pwm mode and compare mode. Indicate channel 9 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
8 CH8F	**Channel 8 Flag** Indicate channel 8 matching event flag in pwm mode and compare mode. Indicate channel 8 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
7 CH7F	**Channel 7 Flag** Indicate channel 7 matching event flag in pwm mode and compare mode. Indicate channel 7 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
6 CH6F	**Channel 6 Flag** Indicate channel 6 matching event flag in pwm mode and compare mode. Indicate channel 6 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
5 CH5F	**Channel 5 Flag** Indicate channel 5 matching event flag in pwm mode and compare mode. Indicate channel 5 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.

Yuntu Microelectronics

Field	Function
4 CH4F	**Channel 4 Flag** Indicate channel 4 matching event flag in pwm mode and compare mode. Indicate channel 4 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
3 CH3F	**Channel 3 Flag** Indicate channel 3 matching event flag in pwm mode and compare mode. Indicate channel 3 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
2 CH2F	**Channel 2 Flag** Indicate channel 2 matching event flag in pwm mode and compare mode. Indicate channel 2 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
1 CH1F	**Channel 1 Flag** Indicate channel 1 matching event flag in pwm mode and compare mode. Indicate channel 1 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.
0 CH0F	**Channel 0 Flag** Indicate channel 0 matching event flag in pwm mode and compare mode. Indicate channel 0 capture event flag in input capture mode. Write 1 to clear the bit. 0b - No channel event occurs. 1b - Channel event occurs.

Yuntu Microelectronics

31.3.1.4 MPWM INTE0 Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15 TOIE	CH14 TOIE	CH13 TOIE	CH12 TOIE	CH11 TOIE	CH10 TOIE	CH9T OIE	CH8T OIE	CH7T OIE	CH6T OIE	CH5T OIE	CH4T OIE	CH3T OIE	CH2T OIE	CH1T OIE	CH0T OIE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.11: MPWM INTE0 Register Description

Field	Function
15 CH15TOIE	**Channel 15 Timer Overflow Interrupt Enable** Channel 15 timer overflow interrupt enable/disable. 0b - Channel 15 timer overflow interrupt is disabled. 1b - Channel 15 timer overflow interrupt is enabled.
14 CH14TOIE	**Channel 14 Timer Overflow Interrupt Enable** Channel 14 timer overflow interrupt enable/disable. 0b - Channel 14 timer overflow interrupt is disabled. 1b - Channel 14 timer overflow interrupt is enabled.
13 CH13TOIE	**Channel 13 Timer Overflow Interrupt Enable** Channel 13 timer overflow interrupt enable/disable. 0b - Channel 13 timer overflow interrupt is disabled. 1b - Channel 13 timer overflow interrupt is enabled.
12 CH12TOIE	**Channel 12 Timer Overflow Interrupt Enable** Channel 12 timer overflow interrupt enable/disable. 0b - Channel 12 timer overflow interrupt is disabled. 1b - Channel 12 timer overflow interrupt is enabled.
11 CH11TOIE	**Channel 11 Timer Overflow Interrupt Enable** Channel 11 timer overflow interrupt enable/disable. 0b - Channel 11 timer overflow interrupt is disabled. 1b - Channel 11 timer overflow interrupt is enabled.
10 CH10TOIE	**Channel 10 Timer Overflow Interrupt Enable** Channel 10 timer overflow interrupt enable/disable. 0b - Channel 10 timer overflow interrupt is disabled. 1b - Channel 10 timer overflow interrupt is enabled.
9	**Channel 9 Timer Overflow Interrupt Enable**

Yuntu Microelectronics

Field	Function
CH9TOIE	Channel 9 timer overflow interrupt enable/disable. 0b - Channel 9 timer overflow interrupt is disabled. 1b - Channel 9 timer overflow interrupt is enabled.
8 CH8TOIE	**Channel 8 Timer Overflow Interrupt Enable** Channel 8 timer overflow interrupt enable/disable. 0b - Channel 8 timer overflow interrupt is disabled. 1b - Channel 8 timer overflow interrupt is enabled.
7 CH7TOIE	**Channel 7 Timer Overflow Interrupt Enable** Channel 7 timer overflow interrupt enable/disable. 0b - Channel 7 timer overflow interrupt is disabled. 1b - Channel 7 timer overflow interrupt is enabled.
6 CH6TOIE	**Channel 6 Timer Overflow Interrupt Enable** Channel 6 timer overflow interrupt enable/disable. 0b - Channel 6 timer overflow interrupt is disabled. 1b - Channel 6 timer overflow interrupt is enabled.
5 CH5TOIE	**Channel 5 Timer Overflow Interrupt Enable** Channel 5 timer overflow interrupt enable/disable. 0b - Channel 5 timer overflow interrupt is disabled. 1b - Channel 5 timer overflow interrupt is enabled.
4 CH4TOIE	**Channel 4 Timer Overflow Interrupt Enable** Channel 4 timer overflow interrupt enable/disable. 0b - Channel 4 timer overflow interrupt is disabled. 1b - Channel 4 timer overflow interrupt is enabled.
3 CH3TOIE	**Channel 3 Timer Overflow Interrupt Enable** Channel 3 timer overflow interrupt enable/disable. 0b - Channel 3 timer overflow interrupt is disabled. 1b - Channel 3 timer overflow interrupt is enabled.
2 CH2TOIE	**Channel 2 Timer Overflow Interrupt Enable** Channel 2 timer overflow interrupt enable/disable. 0b - Channel 2 timer overflow interrupt is disabled. 1b - Channel 2 timer overflow interrupt is enabled.
1 CH1TOIE	**Channel 1 Timer Overflow Interrupt Enable** Channel 1 timer overflow interrupt enable/disable. 0b - Channel 1 timer overflow interrupt is disabled. 1b - Channel 1 timer overflow interrupt is enabled.
0	**Channel 0 Timer Overflow Interrupt Enable**

Yuntu Microelectronics

Field	Function
CH0TOIE	Channel 0 timer overflow interrupt enable/disable. 0b - Channel 0 timer overflow interrupt is disabled. 1b - Channel 0 timer overflow interrupt is enabled.

31.3.1.5 MPWM INTE1 Register

Offset: 10h

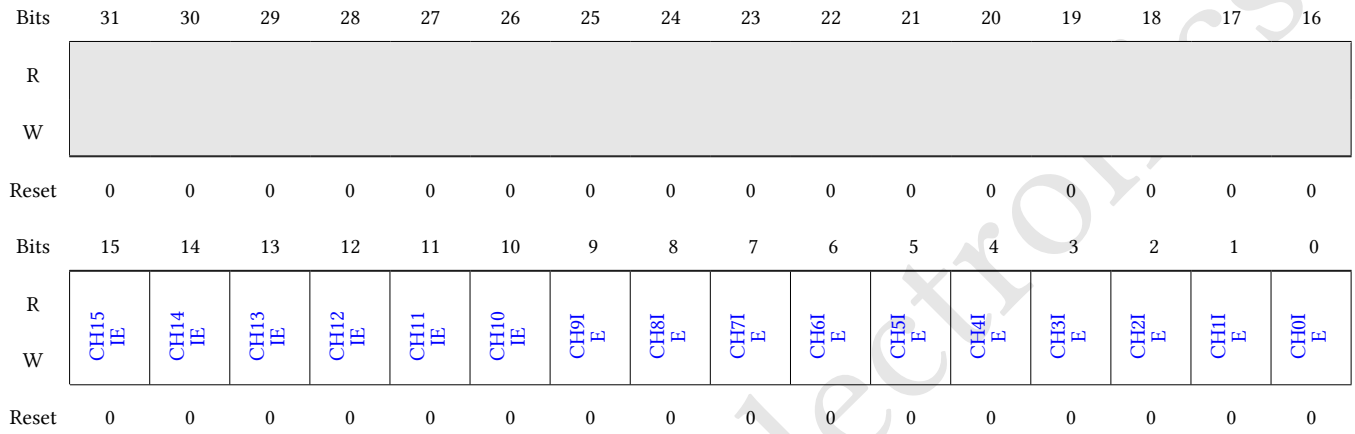


Table 31.12: MPWM INTE1 Register Description

Field	Function
15 CH15IE	**Channel 15 Interrupt Enable** Channel 15 interrupt enable/disable. 0b - Channel 15 interrupt is disabled. 1b - Channel 15 interrupt is enabled.
14 CH14IE	**Channel 14 Interrupt Enable** Channel 14 interrupt enable/disable. 0b - Channel 14 interrupt is disabled. 1b - Channel 14 interrupt is enabled.
13 CH13IE	**Channel 13 Interrupt Enable** Channel 13 interrupt enable/disable. 0b - Channel 13 interrupt is disabled. 1b - Channel 13 interrupt is enabled.
12 CH12IE	**Channel 12 Interrupt Enable** Channel 12 interrupt enable/disable. 0b - Channel 12 interrupt is disabled. 1b - Channel 12 interrupt is enabled.
11	**Channel 11 Interrupt Enable**

Yuntu Microelectronics

Field	Function
CH11IE	Channel 11 interrupt enable/disable. 0b - Channel 11 interrupt is disabled. 1b - Channel 11 interrupt is enabled.
10 CH10IE	**Channel 10 Interrupt Enable** Channel 10 interrupt enable/disable. 0b - Channel 10 interrupt is disabled. 1b - Channel 10 interrupt is enabled.
9 CH9IE	**Channel 9 Interrupt Enable** Channel 9 interrupt enable/disable. 0b - Channel 9 interrupt is disabled. 1b - Channel 9 interrupt is enabled.
8 CH8IE	**Channel 8 Interrupt Enable** Channel 8 interrupt enable/disable. 0b - Channel 8 interrupt is disabled. 1b - Channel 8 interrupt is enabled.
7 CH7IE	**Channel 7 Interrupt Enable** Channel 7 interrupt enable/disable. 0b - Channel 7 interrupt is disabled. 1b - Channel 7 interrupt is enabled.
6 CH6IE	**Channel 6 Interrupt Enable** Channel 6 interrupt enable/disable. 0b - Channel 6 interrupt is disabled. 1b - Channel 6 interrupt is enabled.
5 CH5IE	**Channel 5 Interrupt Enable** Channel 5 interrupt enable/disable. 0b - Channel 5 interrupt is disabled. 1b - Channel 5 interrupt is enabled.
4 CH4IE	**Channel 4 Interrupt Enable** Channel 4 interrupt enable/disable. 0b - Channel 4 interrupt is disabled. 1b - Channel 4 interrupt is enabled.
3 CH3IE	**Channel 3 Interrupt Enable** Channel 3 interrupt enable/disable. 0b - Channel 3 interrupt is disabled. 1b - Channel 3 interrupt is enabled.
2	**Channel 2 Interrupt Enable**

Yuntu Microelectronics

Field	Function
CH2IE	Channel 2 interrupt enable/disable. 0b - Channel 2 interrupt is disabled. 1b - Channel 2 interrupt is enabled.
1 CH1IE	**Channel 1 Interrupt Enable** Channel 1 interrupt enable/disable. 0b - Channel 1 interrupt is disabled. 1b - Channel 1 interrupt is enabled.
0 CH0IE	**Channel 0 Interrupt Enable** Channel 0 interrupt enable/disable. 0b - Channel 0 interrupt is disabled. 1b - Channel 0 interrupt is enabled.

31.3.1.6 MPWM CTRL Register

Offset: 14h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15EN	CH14EN	CH13EN	CH12EN	CH11EN	CH10EN	CH9EN	CH8EN	CH7EN	CH6EN	CH5EN	CH4EN	CH3EN	CH2EN	CH1EN	CH0EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.13: MPWM CTRL Register Description

Field	Function
15 CH15EN	**Channel 15 Enable** Channel 15 enable/disable. 0b - Channel 15 is disabled. 1b - Channel 15 is enabled.
14 CH14EN	**Channel 14 Enable** Channel 14 enable/disable. 0b - Channel 14 is disabled. 1b - Channel 14 is enabled.
13	**Channel 13 Enable**

Yuntu Microelectronics

Field	Function
CH13EN	Channel 13 enable/disable. 0b - Channel 13 is disabled. 1b - Channel 13 is enabled.
12 CH12EN	**Channel 12 Enable** Channel 12 enable/disable. 0b - Channel 12 is disabled. 1b - Channel 12 is enabled.
11 CH11EN	**Channel 11 Enable** Channel 11 enable/disable. 0b - Channel 11 is disabled. 1b - Channel 11 is enabled.
10 CH10EN	**Channel 10 Enable** Channel 10 enable/disable. 0b - Channel 10 is disabled. 1b - Channel 10 is enabled.
9 CH9EN	**Channel 9 Enable** Channel 9 enable/disable. 0b - Channel 9 is disabled. 1b - Channel 9 is enabled.
8 CH8EN	**Channel 8 Enable** Channel 8 enable/disable. 0b - Channel 8 is disabled. 1b - Channel 8 is enabled.
7 CH7EN	**Channel 7 Enable** Channel 7 enable/disable. 0b - Channel 7 is disabled. 1b - Channel 7 is enabled.
6 CH6EN	**Channel 6 Enable** Channel 6 enable/disable. 0b - Channel 6 is disabled. 1b - Channel 6 is enabled.
5 CH5EN	**Channel 5 Enable** Channel 5 enable/disable. 0b - Channel 5 is disabled. 1b - Channel 5 is enabled.
4	**Channel 4 Enable**

Yuntu Microelectronics

Field	Function
CH4EN	Channel 4 enable/disable. 0b - Channel 4 is disabled. 1b - Channel 4 is enabled.
3 CH3EN	**Channel 3 Enable** Channel 3 enable/disable. 0b - Channel 3 is disabled. 1b - Channel 3 is enabled.
2 CH2EN	**Channel 2 Enable** Channel 2 enable/disable. 0b - Channel 2 is disabled. 1b - Channel 2 is enabled.
1 CH1EN	**Channel 1 Enable** Channel 1 enable/disable. 0b - Channel 1 is disabled. 1b - Channel 1 is enabled.
0 CH0EN	**Channel 0 Enable** Channel 0 enable/disable. 0b - Channel 0 is disabled. 1b - Channel 0 is enabled.

31.3.1.7 MPWM SWT Register

Offset: 18h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15SWT	CH14SWT	CH13SWT	CH12SWT	CH11SWT	CH10SWT	CH9SWT	CH8SWT	CH7SWT	CH6SWT	CH5SWT	CH4SWT	CH3SWT	CH2SWT	CH1SWT	CH0SWT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.14: MPWM SWT Register Description

Field	Function
15	**Channel 15 Software Trigger**

Yuntu Microelectronics

Field	Function
CH15SWT	Channel 15 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
14 CH14SWT	**Channel 14 Software Trigger** Channel 14 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
13 CH13SWT	**Channel 13 Software Trigger** Channel 13 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
12 CH12SWT	**Channel 12 Software Trigger** Channel 12 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
11 CH11SWT	**Channel 11 Software Trigger** Channel 11 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
10 CH10SWT	**Channel 10 Software Trigger** Channel 10 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
9 CH9SWT	**Channel 9 Software Trigger** Channel 9 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
8 CH8SWT	**Channel 8 Software Trigger** Channel 8 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
7 CH7SWT	**Channel 7 Software Trigger** Channel 7 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
6 CH6SWT	**Channel 6 Software Trigger** Channel 6 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
5 CH5SWT	**Channel 5 Software Trigger** Channel 5 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
4	**Channel 4 Software Trigger**

Yuntu Microelectronics

Field	Function
CH4SWT	Channel 4 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
3 CH3SWT	**Channel 3 Software Trigger** Channel 3 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
2 CH2SWT	**Channel 2 Software Trigger** Channel 2 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
1 CH1SWT	**Channel 1 Software Trigger** Channel 1 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.
0 CH0SWT	**Channel 0 Software Trigger** Channel 0 software trigger by writing 1 to this bit and keep a bus clock cycle. The bit is write only. Read operation is invalid and always returns 0.

31.3.1.8 MPWM INIT Register

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15 INIT	CH14 INIT	CH13 INIT	CH12 INIT	CH11 INIT	CH10 INIT	CH9I NIT	CH8I NIT	CH7I NIT	CH6I NIT	CH5I NIT	CH4I NIT	CH3I NIT	CH2I NIT	CH1I NIT	CH0I NIT
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.15: MPWM INIT Register Description

Field	Function
15 CH15INIT	**Channel 15 Initialization** Channel 15 initial value in PWM mode and compare mode before Channel 15 is enabled. 0b - Channel 15 initial value is 0. 1b - Channel 15 initial value is 1.
14	**Channel 14 Initialization**

Yuntu Microelectronics

Field	Function
CH14INIT	Channel 14 initial value in PWM mode and compare mode before Channel 14 is enabled. 0b - Channel 14 initial value is 0. 1b - Channel 14 initial value is 1.
13 CH13INIT	**Channel 13 Initialization** Channel 13 initial value in PWM mode and compare mode before Channel 13 is enabled. 0b - Channel 13 initial value is 0. 1b - Channel 13 initial value is 1.
12 CH12INIT	**Channel 12 Initialization** Channel 12 initial value in PWM mode and compare mode before Channel 12 is enabled. 0b - Channel initial value is 0. 1b - Channel 12 initial value is 1.
11 CH11INIT	**Channel 11 Initialization** Channel 11 initial value in PWM mode and compare mode before Channel 11 is enabled. 0b - Channel initial value is 0. 1b - Channel 11 initial value is 1.
10 CH10INIT	**Channel 10 Initialization** Channel 10 initial value in PWM mode and compare mode before Channel 10 is enabled. 0b - Channel 10 initial value is 0. 1b - Channel 10 initial value is 1.
9 CH9INIT	**Channel 9 Initialization** Channel 9 initial value in PWM mode and compare mode before Channel 9 is enabled. 0b - Channel 9 initial value is 0. 1b - Channel 9 initial value is 1.
8 CH8INIT	**Channel 8 Initialization** Channel 8 initial value in PWM mode and compare mode before Channel 8 is enabled. 0b - Channel 8 initial value is 0. 1b - Channel 8 initial value is 1.
7 CH7INIT	**Channel 7 Initialization** Channel 7 initial value in PWM mode and compare mode before Channel 7 is enabled. 0b - Channel 7 initial value is 0. 1b - Channel 7 initial value is 1.
6	**Channel 6 Initialization**

Yuntu Microelectronics

Field	Function
CH6INIT	Channel 6 initial value in PWM mode and compare mode before Channel 6 is enabled. 0b - Channel 6 initial value is 0. 1b - Channel 6 initial value is 1.
5 CH5INIT	**Channel 5 Initialization** Channel 5 initial value in PWM mode and compare mode before Channel 5 is enabled. 0b - Channel 5 initial value is 0. 1b - Channel 5 initial value is 1.
4 CH4INIT	**Channel 4 Initialization** Channel 4 initial value in PWM mode and compare mode before Channel 4 is enabled. 0b - Channel 4 initial value is 0. 1b - Channel 4 initial value is 1.
3 CH3INIT	**Channel 3 Initialization** Channel 3 initial value in PWM mode and compare mode before Channel 3 is enabled. 0b - Channel 3 initial value is 0. 1b - Channel 3 initial value is 1.
2 CH2INIT	**Channel 2 Initialization** Channel 2 initial value in PWM mode and compare mode before Channel 2 is enabled. 0b - Channel 2 initial value is 0. 1b - Channel 2 initial value is 1.
1 CH1INIT	**Channel 1 Initialization** Channel 1 initial value in PWM mode and compare mode before Channel 1 is enabled. 0b - Channel 1 initial value is 0. 1b - Channel 1 initial value is 1.
0 CH0INIT	**Channel 0 Initialization** Channel 0 initial value in PWM mode and compare mode before Channel 0 is enabled. 0b - Channel 0 initial value is 0. 1b - Channel 0 initial value is 1.

Yuntu Microelectronics

31.3.1.9 MPWM POL Register

Offset: 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CH15POL	CH14POL	CH13POL	CH12POL	CH11POL	CH10POL	CH9POL	CH8POL	CH7POL	CH6POL	CH5POL	CH4POL	CH3POL	CH2POL	CH1POL	CH0POL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.16: MPWM POL Register Description

Field	Function
15 CH15POL	**Channel 15 Polarity** Channel 15 polarity in PWM mode and compare mode. 0b - Not invert the channel 15 output. 1b - Invert the channel 15 output.
14 CH14POL	**Channel 14 Polarity** Channel 14 polarity in PWM mode and compare mode. 0b - Not invert the channel 14 output. 1b - Invert the channel 14 output.
13 CH13POL	**Channel 13 Polarity** Channel 13 polarity in PWM mode and compare mode. 0b - Not invert the channel 13 output. 1b - Invert the channel 13 output.
12 CH12POL	**Channel 12 Polarity** Channel 12 polarity in PWM mode and compare mode. 0b - Not invert the channel 12 output. 1b - Invert the channel 12 output.
11 CH11POL	**Channel 11 Polarity** Channel 11 polarity in PWM mode and compare mode. 0b - Not invert the channel 11 output. 1b - Invert the channel 11 output.
10 CH10POL	**Channel 10 Polarity** Channel 10 polarity in PWM mode and compare mode. 0b - Not invert the channel 10 output. 1b - Invert the channel 10 output.
9	**Channel 9 Polarity**

Yuntu Microelectronics

Field	Function
CH9POL	Channel 9 polarity in PWM mode and compare mode. 0b - Not invert the channel 9 output. 1b - Invert the channel 9 output.
8 CH8POL	**Channel 8 Polarity** Channel 8 polarity in PWM mode and compare mode. 0b - Not invert the channel 8 output. 1b - Invert the channel 8 output.
7 CH7POL	**Channel 7 Polarity** Channel 7 polarity in PWM mode and compare mode. 0b - Not invert the channel 7 output. 1b - Invert the channel 7 output.
6 CH6POL	**Channel 6 Polarity** Channel 6 polarity in PWM mode and compare mode. 0b - Not invert the channel 6 output. 1b - Invert the channel 6 output.
5 CH5POL	**Channel 5 Polarity** Channel 5 polarity in PWM mode and compare mode. 0b - Not invert the channel 5 output. 1b - Invert the channel 5 output.
4 CH4POL	**Channel 4 Polarity** Channel 4 polarity in PWM mode and compare mode. 0b - Not invert the channel 4 output. 1b - Invert the channel 4 output.
3 CH3POL	**Channel 3 Polarity** Channel 3 polarity in PWM mode and compare mode. 0b - Not invert the channel 3 output. 1b - Invert the channel 3 output.
2 CH2POL	**Channel 2 Polarity** Channel 2 polarity in PWM mode and compare mode. 0b - Not invert the channel 2 output. 1b - Invert the channel 2 output.
1 CH1POL	**Channel 1 Polarity** Channel 1 polarity in PWM mode and compare mode. 0b - Not invert the channel 1 output. 1b - Invert the channel 1 output.
0	**Channel 0 Polarity**

Yuntu Microelectronics

Field	Function
CH0POL	Channel 0 polarity in PWM mode and compare mode. 0b - Not invert the channel 0 output. 1b - Invert the channel 0 output.

31.3.1.10 MPWM CH_CFG Register

Offset: 40h + n * 20h

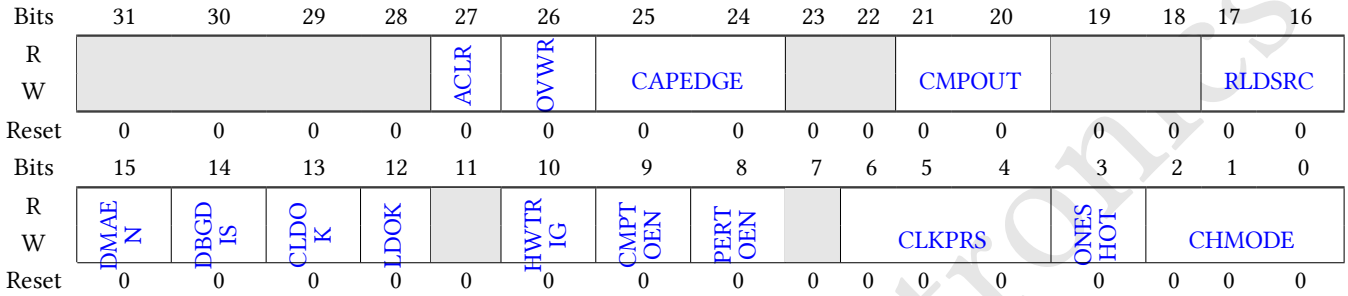


Table 31.17: MPWM CH_CFG Register Description

Field	Function
27 ACLR	**Auto clear** Counter is automatically reset to 0 when capture event occurs. 0b - Auto clear is disabled. 1b - Auto clear is enabled.
26 OVWR	**Over Write** Over write the CAP register from counter when new capture event occurs. 0b - Over write is disabled. 1b - Over write is enabled.
25 - 24 CAPEDGE	**Capture Edge Selection** Select the capture edge in the input capture mode. 00b - Capture is disabled. 01b - Capture rising edge. 10b - Capture falling edge. 11b - Capture both rising edge and falling edge.
21 - 20 CMPOUT	**Compare Output** Select the output value when counter matches with CMP in compare mode. 00b - Compare Output disabled. 01b - Output value 0. 10b - Output value 1. 11b - Output value toggle.
17 - 16	**Reloading Source**

Yuntu Microelectronics

Field	Function
RLDSRC	Double Buffer Registers reloading sources. 00b - PERIOD matching point reload. 01b - Hardware trigger point reload. 10b - Software trigger point reload. 11b - Double buffer mechanism is disabled.
15 DMAEN	**DMA Enable** When DMA is enabled and STS1[CHxF] is set, generate a DMA request. 0b - DMA request is disabled. 1b - DMA request is enabled.
14 DBGDIS	**Debug Mode Disabled** Enable/disable the debug mode. 0b - Counter stops under debug mode. 1b - Counter keeps running under debug mode.
13 CLDOK	**Clear LDOK** Write 1 to this bit to clear CH_CFG[LDOK], and it automatically clear when CH_CFG[LDOK]=0.
12 LDOK	**Registers Loading OK** Set LDOK to indicate registers(CMP and PERIOD) is ready for loading, it is cleared automatically when registers load successfully. 0b - Registers is not ready for loading. 1b - Registers is ready for loading and wait for loading opportunity.
10 HWTRIG	**Hardware Trigger** Hardware trigger enable/disable. 0b - Hardware trigger is disabled. 1b - Hardware trigger is enabled.
9 CMPTOEN	**CMP Matching Trigger Output Enable** Registers CMP matching trigger output enable/disable. 0b - Registers CMP matching trigger output is disabled. 1b - Registers CMP matching trigger output is enabled.
8 PERTOEN	**PERIOD Matching Trigger Output Enable** Registers PERIOD matching trigger output enable/disable. 0b - Registers PERIOD matching trigger output is disabled. 1b - Registers PERIOD matching trigger output is enabled.
6 - 4	**Clock Prescaler**

Yuntu Microelectronics

Field	Function
CLKPRS	Clock prescaler for counter. 000b - Divide by 1. 001b - Divide by 2. 010b - Divide by 4. 011b - Divide by 8. 100b - Divide by 16. 101b - Divide by 32. 110b - Divide by 64. 111b - Divide by 128.
3 ONESHOT	**One Shot** Counter counting mode. 0b - Continuous mode. 1b - One shot mode.
2 - 0 CHMODE	**Channel Mode** Channel mode. 000b - Common timer 001b - PWM mode. 010b - Output compare mode. 011b - Input capture mode. 100b - Pulse counter mode.

31.3.1.11 MPWM CH_CNT Register

Offset: 44h + n * 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 31.18: MPWM CH_CNT Register Description

Field	Function
15 - 0 CNT	**Counter Status** Reading this register gets the counter value, and writing to this register is invalid.

31.3.1.12 MPWM CH_CAP Register

Offset: 48h + n * 20h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 31.19: MPWM CH_CAP Register Description

Field	Function
15 - 0 CAP	**Capture Status** In capture mode, store counter value when a capture event occurs.

31.3.1.13 MPWM CH_PERIOD Register

Offset: 50h + n * 20h

Yuntu Microelectronics

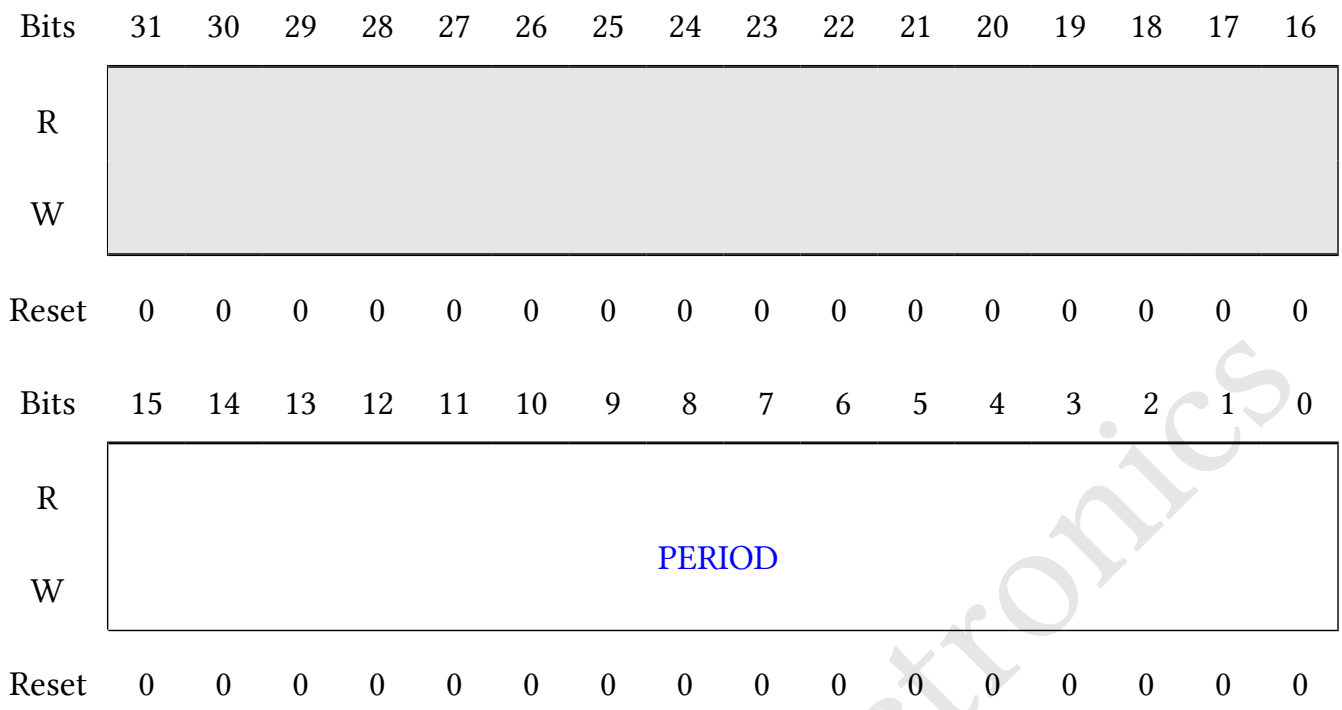


Table 31.20: MPWM CH_PERIOD Register Description

Field	Function
15 - 0 PERIOD	**Counter Counting Period** Counter counting period.

Yuntu Microelectronics

31.3.1.14 MPWM CH_CMP Register

Offset: 54h + n * 20h

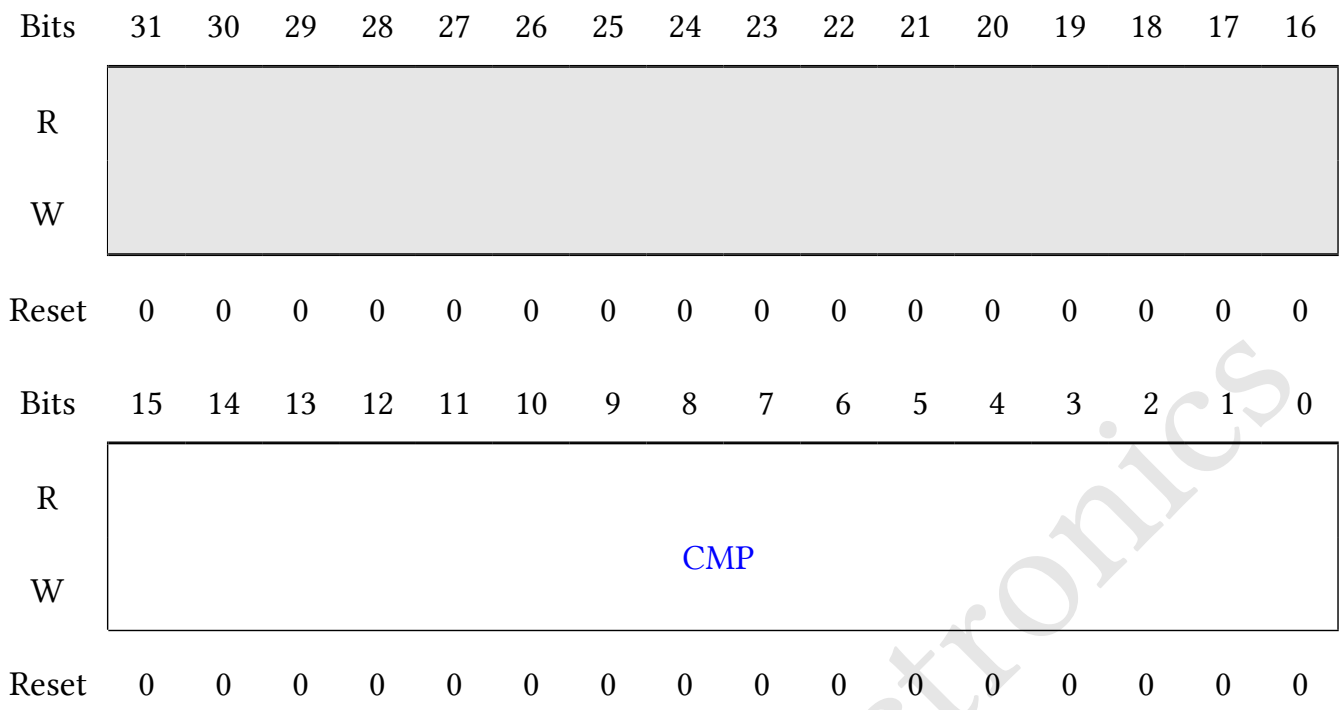


Table 31.21: MPWM CH_CMP Register Description

Field	Function
15 - 0 CMP	**Counter Compare Value** Counter compare value.

31.4 Function

31.4.1 Clock

The MPWM module works in bus clock domain. Each channel contains a counter and clock prescaler independently. The clock prescaler is used for dividing the counter clock. The prescaler supports 1, 2, 4, 8, 16, 32,64, 128 frequency division.

31.4.2 Common Timer

When CH_CFG[CHMODE]=0x0, the channel is in common timer mode. Each channel has a 16-bit counter. The Counter counts from 1 to CH_PERIOD[PERIOD] and supports two counting mode:

- Continuous mode
- One-shot mode. the Continuous mode means counter keeps counting. When the timer overflow event occurs, the counter will count from 1 again. But for one-shot mode, the counter will stop counting until the next time counting trigger comes. In one-shot mode, counting starts with a hardware trigger(from TMU) or software trigger(CTRL[CHxEN], x=0-15).

Anyway, when the counter reaches CH_PERIOD[PERIOD], the timer overflow event occurs. The timeout flag (STS0[CHxTOF], x=0-15) will be set. The timeout flag (STS0[CHxTOF], x=0-15) will be cleared by writing 1 to this bit. As shown in Figure MPWM Timer Overflow.

Yuntu Microelectronics

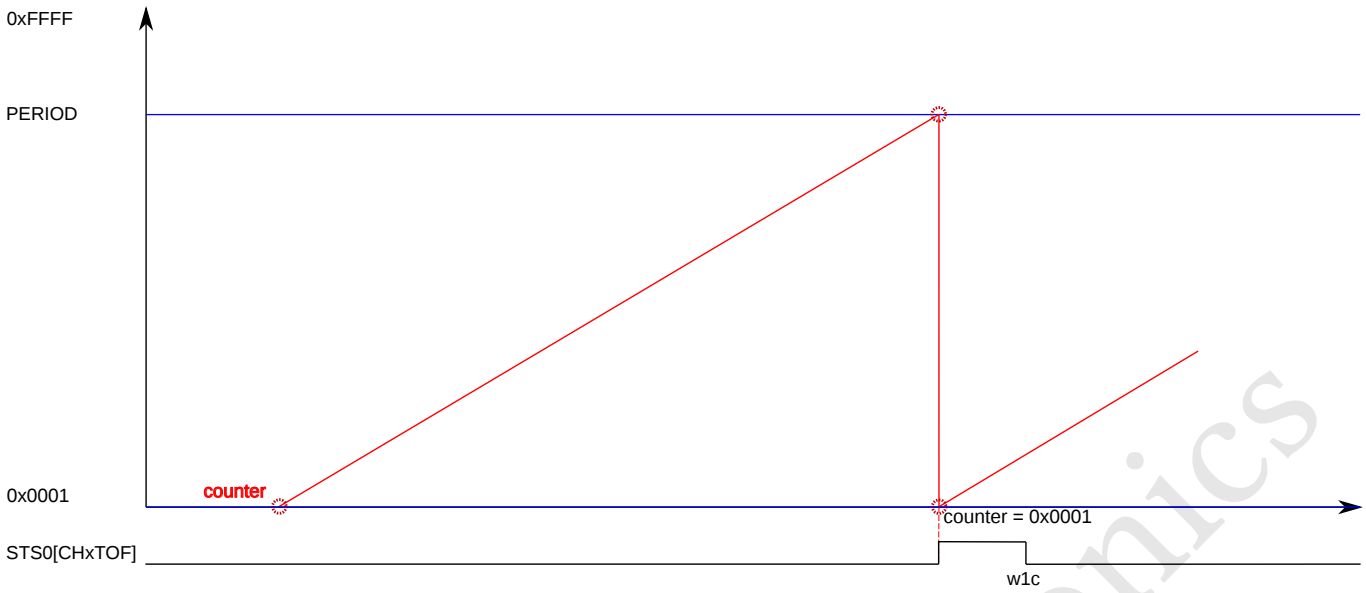


Figure 31.2: MPWM Timer Overflow

31.4.3 PWM Mode

When CH_CTRL[CHMODE]=0x1, the channel x is in PWM mode. Each channel is capable of generating edge aligned PWM. All channels support independent period and duty cycle configuration. As shown in Figure MPWM PWM Generation Block Diagram.

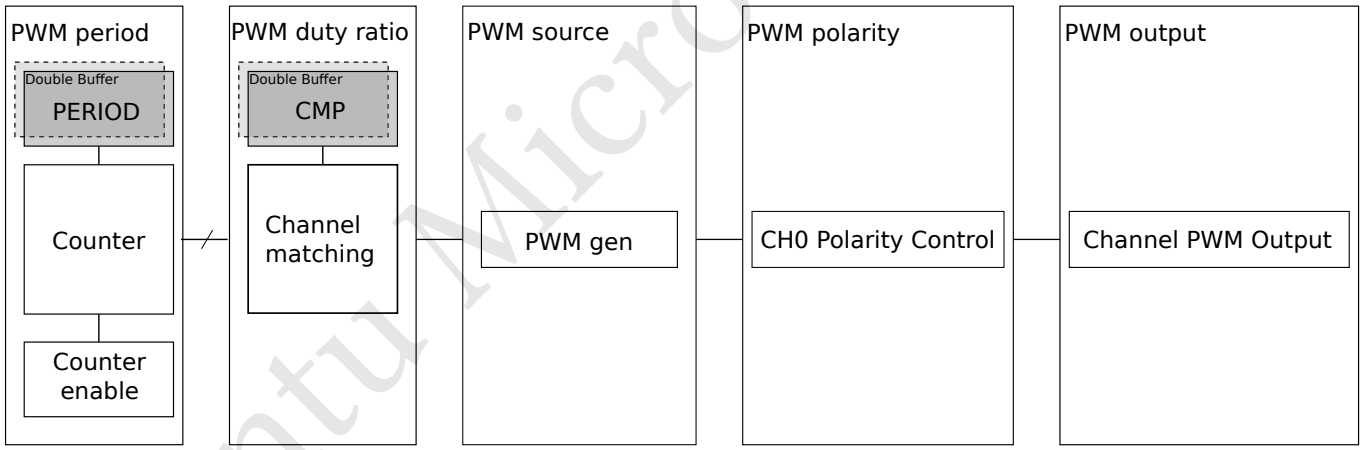


Figure 31.3: MPWM PWM Generation Block Diagram

The period of output PWM is up to counter clock and CH_PERIOD[PERIOD]:

$$PER_{PWM} = \frac{CH_PERIOD[PERIOD]}{CLK_{CNT}}$$

When the period of output PWM is determined, the duty cycle of output PWM is up to CH_CMP[CMP], PWM is set when the counter value is more than CH_CMP[CMP] and the counter value is less than CH_PERIOD[PERIOD], PWM clear when the counter value is more than CH_PERIOD[PERIOD]:

CH_CMP[CMP] < CH_PERIOD[PERIOD]:

Yuntu Microelectronics

$$DUTY_{PWM} = \frac{CH_PERIOD[PERIOD] - CH_CMP[CMP]}{PER_{PWM}} \times 100\%$$

31.4.3.1 Edge Aligned PWM

Edge aligned PWM contains left edge aligned PWM and right edge aligned PWM. As shown in Figures MPWM Left Edge Aligned PWM and MPWM Right Edge Aligned PWM

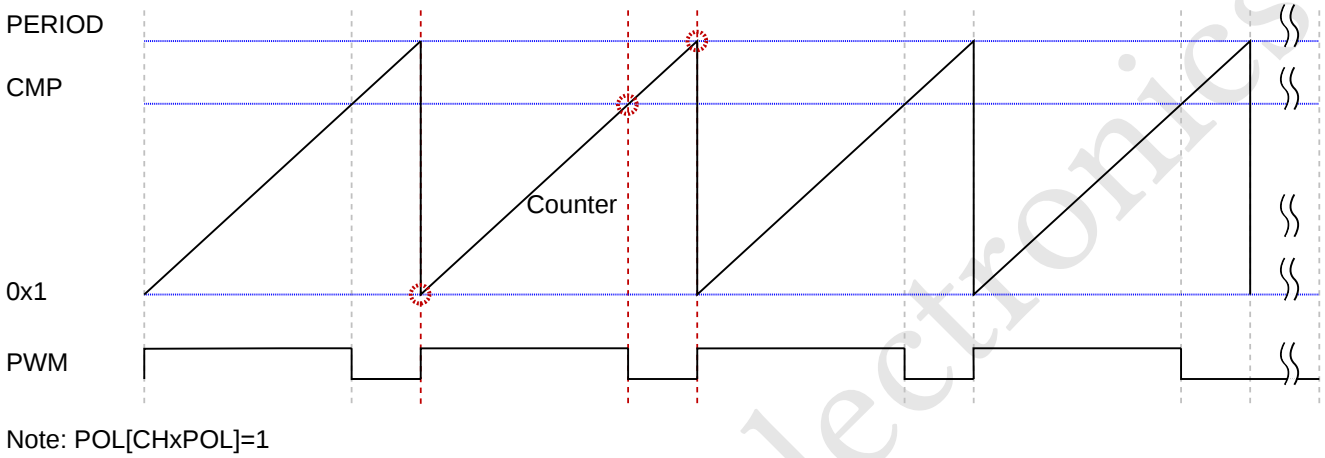


Figure 31.4: MPWM Left Edge Aligned PWM

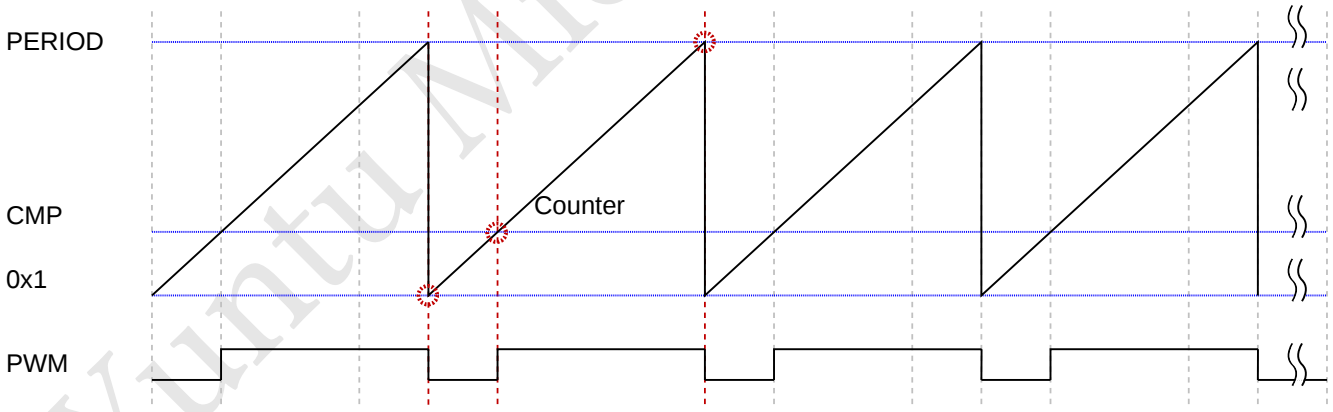


Figure 31.5: MPWM Right Edge Aligned PWM

Left edge aligned PWM configuration:

- Determine CH_PERIOD[PERIOD] and $DUTY_{PWM}$ as needed
- Calculate CH_CMP[CMP]
- Configuring POL[CHxPOL]=1, x=0-15

Yuntu Microelectronics

$$CH_CMP[CMP] = DUTY_{PWM} \times CH_PERIOD[PERIOD]$$

Right edge aligned PWM configuration:

- Determine CH_PERIOD[PERIOD] and $DUTY_{PWM}$ as needed
- Calculate CH_CMP[CMP]
- Configuring POL[CHxPOL]=0, x=0-15

$$CH_CMP[CMP] = ((1 - DUTY_{PWM}) \times CH_PERIOD[PERIOD])$$

31.4.4 Output Compare Mode

When CH_CFG[CHMODE]=0x2, the channel is in output compare mode. In this mode, each channel can hold, set, clear, or toggle the output when the counter reaches the specific value defined by CH_CMP[CMP]. Configure CH_CFG[CMPOUT] to select the action of the corresponding channel when the counter reaches the CH_CMP[CMP]. As shown in Figure MPWM Output Compare. The channel flag (STS[CHxF], x=0-15) will be set when a compare event occurs.

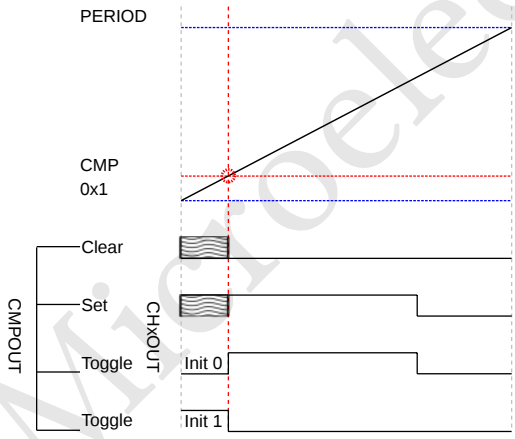


Figure 31.6: MPWM Output Compare

31.4.5 Input Capture Mode

When CH_CFG[CHMODE]=0x3, the channel x is in input capture mode. In this mode, each channel can detect the input signal change defined by CH_CTRL[CAPEDGE]. When the capture event occurs, the current counter value is saved to the CAP register. As shown in Figure MPWM Input Capture. The channel flag (STS[CHxF], x=0-15) will be set when a capture event occurs.

When CH_CFG[ONESHOT] is set and one capture event occurs, current channel x is disabled(CTRL[CHxEN]=0, x=0-15) automatically.

When CH_CFG[OVWR] is set and a new capture event occurs, CH_CAP[CAP] is overwritten from a new counter value. When CH_CFG[OVWR] is cleared and a new capture event occurs, CH_CAP[CAP] will hold value.

Yuntu Microelectronics

When CH_CFG[ACLR] is set and one capture event occurs, the counter is reset to 0 and continues counting. When a new capture event occurs, the counter is reset to 0 and continues counting again. When CH_CFG[ACLR] is cleared and one capture event occurs, the counter is not reset to 0 and continues counting.

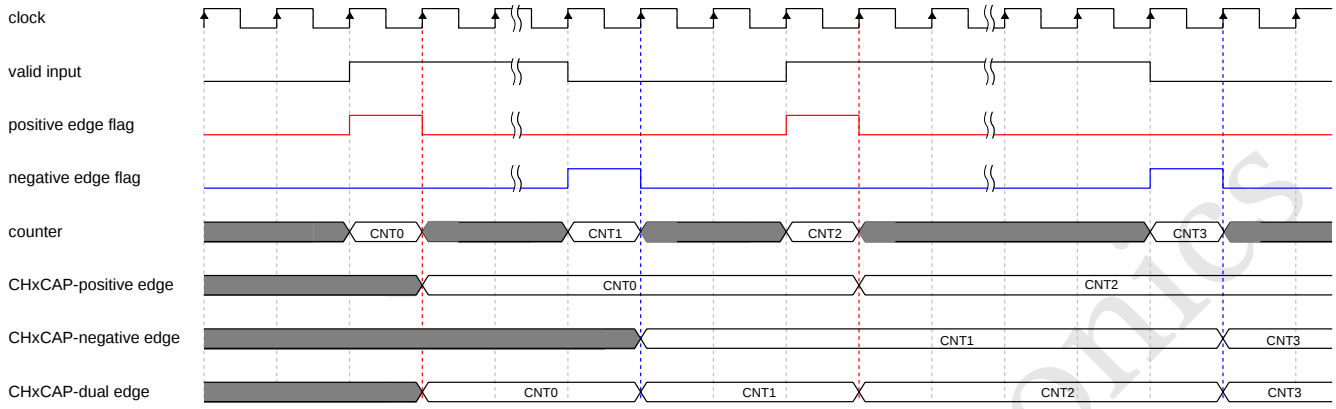


Figure 31.7: MPWM Input Capture

31.4.6 Pulse Counter Mode

When CH_CFG[CHMODE]=0x4, the channel is in pulse counter mode. In this mode, each channel can detect the input signal change defined by CH_CFG[CAPEDGE]. When the valid edge comes, the channel counter accumulates. Reading CH_CNT[CNT] gets pulse number. When the counter value matchs with CMP[CMP], flag STS1[CHxF] is set. If INTE1[CHxIE] is enabled and the matching event interrruput generates.

31.4.7 Output Polarity

In PWM mode or output compare mode, the channel output is controlled by the polarity defined by POLCHxPOL. When POL[CHxPOL]=0(x=0-15), the corresponding channel outputs the raw signal; When POL[CHxPOL]=1(x=0-15), the corresponding channel outputs the inverted signal.

31.4.8 Channels Initialization

When the MPWM is disabled in PWM mode or output compare mode, the channels output the initial value defined by the INITCHxINIT register. When the MPWM is enabled, write value 1 to INITCHxINIT also results in the corresponding channel output the initial value.

31.4.9 Loading Mechanism

Some registers are designed with shadow registers CH_PERIOD[PERIOD] and CH_CMP[CMP]. When the channel is disabled, these registers' value is unlocked and they can be read and written in real time. When the channel is enabled, these registers' value is locked and they can only change at the appropriate load opportunity. Load opportunity contains hardware trigger, software trigger and period load(counter=CH_PERIOD[PERIOD]). The loading mechanism is applied to manage the update of these special registers. As shown in Figure MPWM Loading Mechanism.

Yuntu Microelectronics

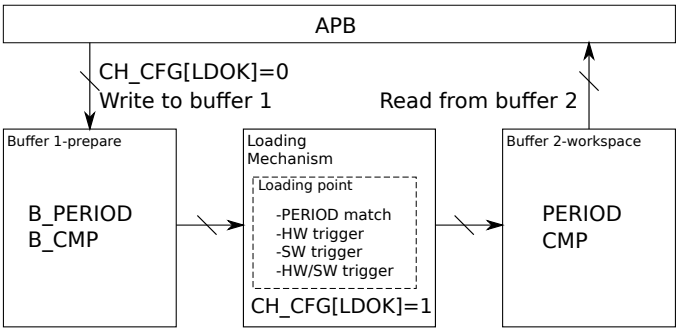


Figure 31.8: MPWM Loading Mechanism

Registers loading configuration:

- Select the loading opportunity
- Configure trigger if need
- Clear CH_CFG[LDOK] by CH_CFG[CLDOK]
- Write buffer registers
- Set CH_CFG[LDOK]=1

The CH_CFG[LDOK] will be cleared automatically if the registers load successfully. So, the CH_CFG[LDOK] indicate loading status too.

Loading mechanism implements by hardware trigger, software trigger and period load(counter=CHxPERIOD[PERIOD]). As shown in Figure MPWM Loading by Trigger Timing.

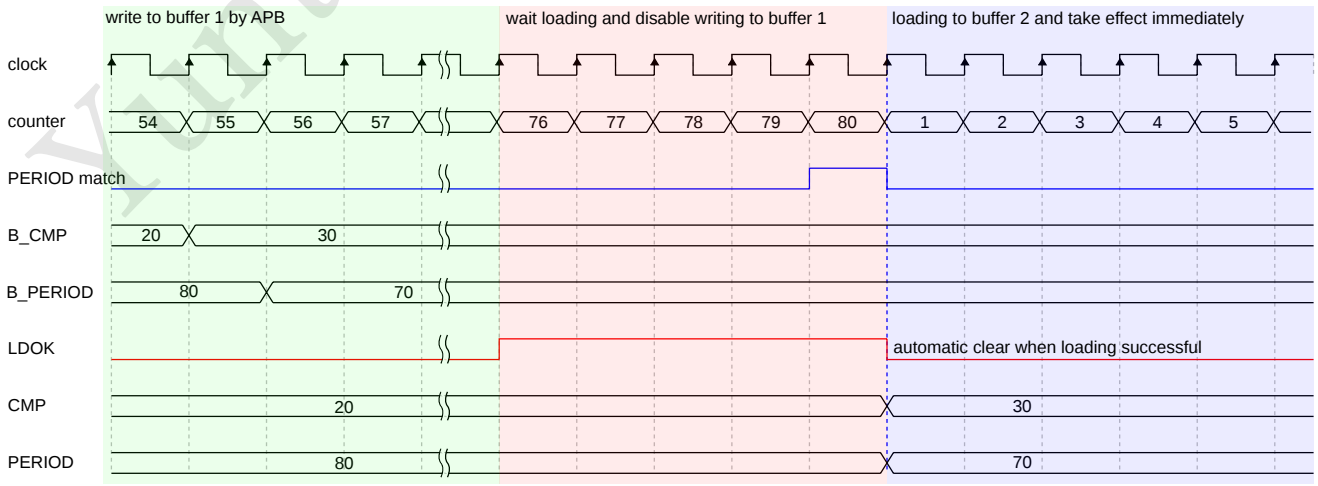


Figure 31.9: MPWM Loading Mechanism Timing

Yuntu Microelectronics

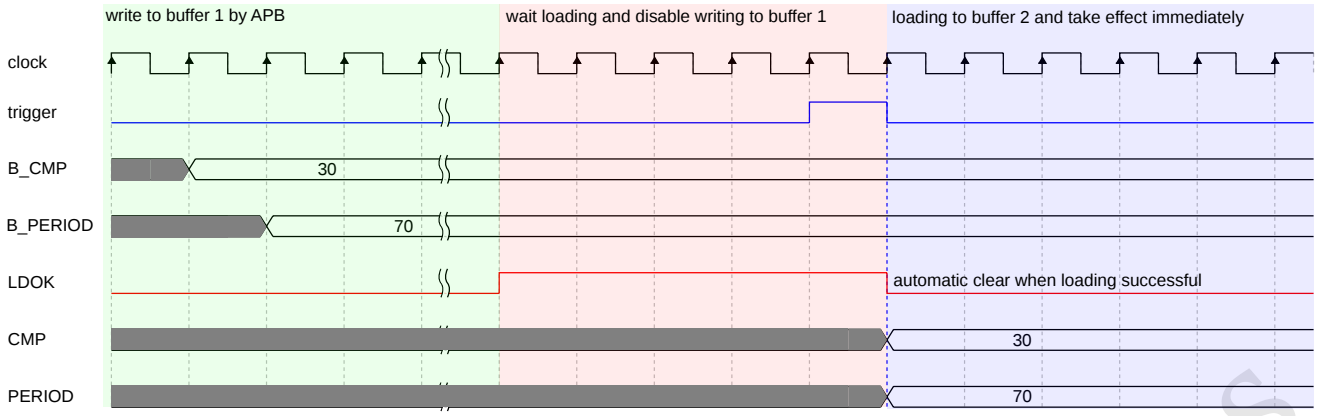


Figure 31.10: MPWM Loading by Trigger Timing

Yuntu Microelectronics

31.4.9.1 Loading Trigger

There is one hardware trigger and one software trigger used for registers' loading. Configure CH_CFG[RLDSRC] to select the trigger source which is used for registers' loading.

The hardware trigger is from the TMU and enabled by configuring CH_CFG[RLDSRC]=1. Writing value 1 to SWTCHxSWT can generate a software trigger and enable it by configuring CH_CFG[RLDSRC]=2.

31.4.10 Output Trigger

The MPWM can output triggers to the TMU module.

Configure CH_CFG[CMPTOEN]=1 and CH_CFG[PERTOEN]=1 to enable the generation of triggers. There are two available match points: CH_CMP[CMP] and CH_PERIOD[PERIOD]. The generated pulses are all one bus clock cycle width. As shown in figure MPWM Output Trigger.

- When the counter value matches with CH_CMP[CMP] and CH_CFG[CMPTOEN] is set, the trigger generates and sustains a bus clock cycle.
- When the counter value matches with CH_PERIOD[PERIOD] and CH_CFG[PERTOEN] is set, the trigger generates and sustains a bus clock cycle.

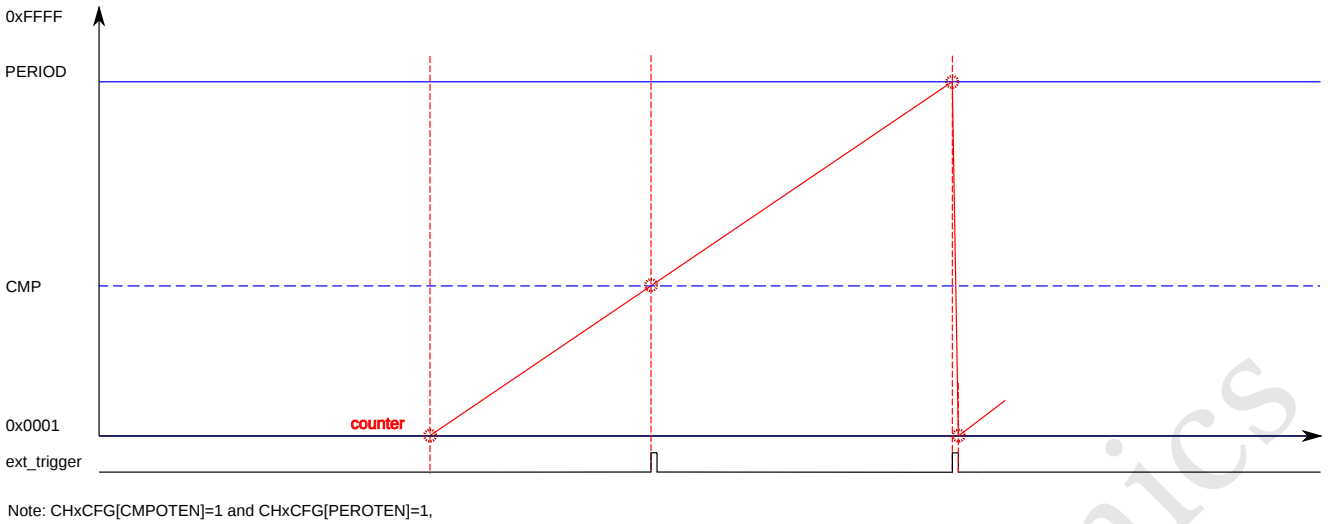


Figure 31.11: MPWM Output Trigger

31.4.11 Power Mode

The MPWMs can work in active mode and sleep mode.

31.4.12 Interrupts

The timer overflow event interrupt and channel event interrupt are available for every channel of MPWMx.

The timer overflow event interrupt is generated by STS0[CHxTOF] when INTE0[CHxTOIE](x=0-15) is set. Write value 1 to clear STS0[CHxTOF](x=0-15).

The channel event interrupt is generated by STS1[CHxF](x=0-15) when INTE1[CHxIE](x=0-15) is set. Write value 1 to clear STS1[CHxF](x=0-15).

31.4.13 DMA

MPWMx supports DMA, this function is enabled by setting CH_CFG[DMAEN]. All channels of MPWMx share the same DMA request. The DMA request is generated when any STS1[CHxF]=1(x=0-15). The DMA request is cleared by clearing STS1[CHxF](x=0-15). The DMA priority is higher than the interrupt.

31.4.14 Reset

During a reset, the MPWMs are in the default mode and disabled.

31.5 Glossary

MPWM	Multiple Pulse Width Modulation
PWM	Pulse Width Modulation
DMA	Direct Memory Access
TMU	Trigger Multiplexer

Yuntu Microelectronics

Chapter 32

Real Time Clock (RTC)

32.1 Introduction

The Real Time Clock (RTC) provides a real time counter which is present in always on domain, hence available in all power modes of MCU. RTC generate interrupts when RTC counter reaches the values of ALM register.

32.1.1 Features

- Register write protection for RTC enable register
- 32-bit seconds counter with overflow flag and optional interrupt
- Configurable 32-bit alarm
- Configurable 1, 2, 4, 8, 16, 32, 64 or 128 Hz square wave output with optional interrupt

32.2 Register Definition

This subsection provides a description of the memory map and registers.

32.2.1 RTC Memory Map

RTC base address 40180000h

Table 32.1: RTC Memory Map

Offset	Register	Width	Access	Reset Value
0h	Second Register (SEC)	32	RW	00000000h
8h	Compensation Register (CMP)	32	RW	00000000h
ch	Alarm Register (ALM)	32	RW	00000000h
10h	Enable Register (EN)	32	RW	00000000h
14h	Control Register (CTRL)	32	RW	00000000h
18h	Interrupt Flag Register (INTF)	32	RW	00000000h
1ch	Interrupt Enable Register (INTE)	32	RW	00000003h

32.2.1.1 RTC SEC Register

RTC Second Register

Yuntu Microelectronics

Offset: 0h

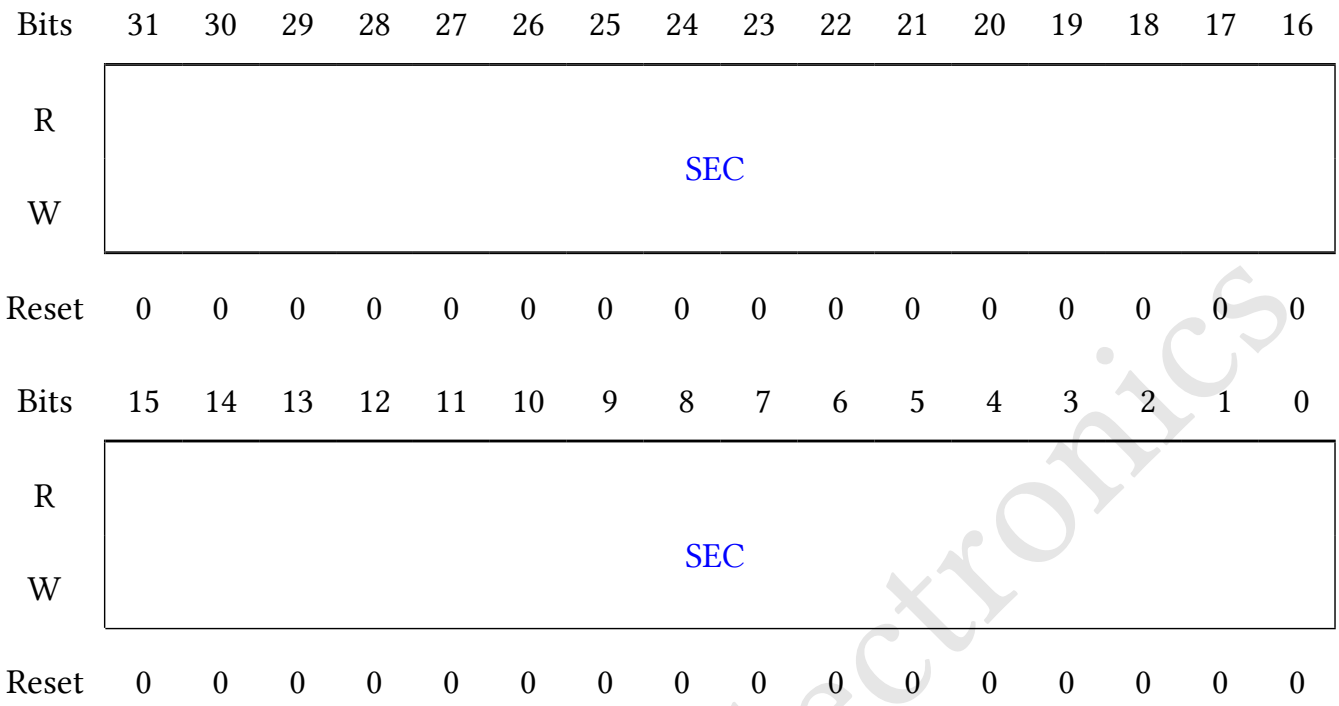


Table 32.2: RTC SEC Register Description

Field	Function
31 - 0 SEC	**Second Value** SEC is read-only when RTC is enabled. The SEC increments once a second provided INTF[OVFIF] is not set. The time counter will read as zero when INTF[OVFIF] is set. When RTC is disabled, SEC can be read and written. Writing to this bit will clear the INTF[OVFIF]. Writing to SEC with zero is not recommended because SEC will read as zero when INTF[OVFIF] is set.

Yuntu Microelectronics

32.2.1.2 RTC CMP Register

RTC Compensation Register

Offset: 8h

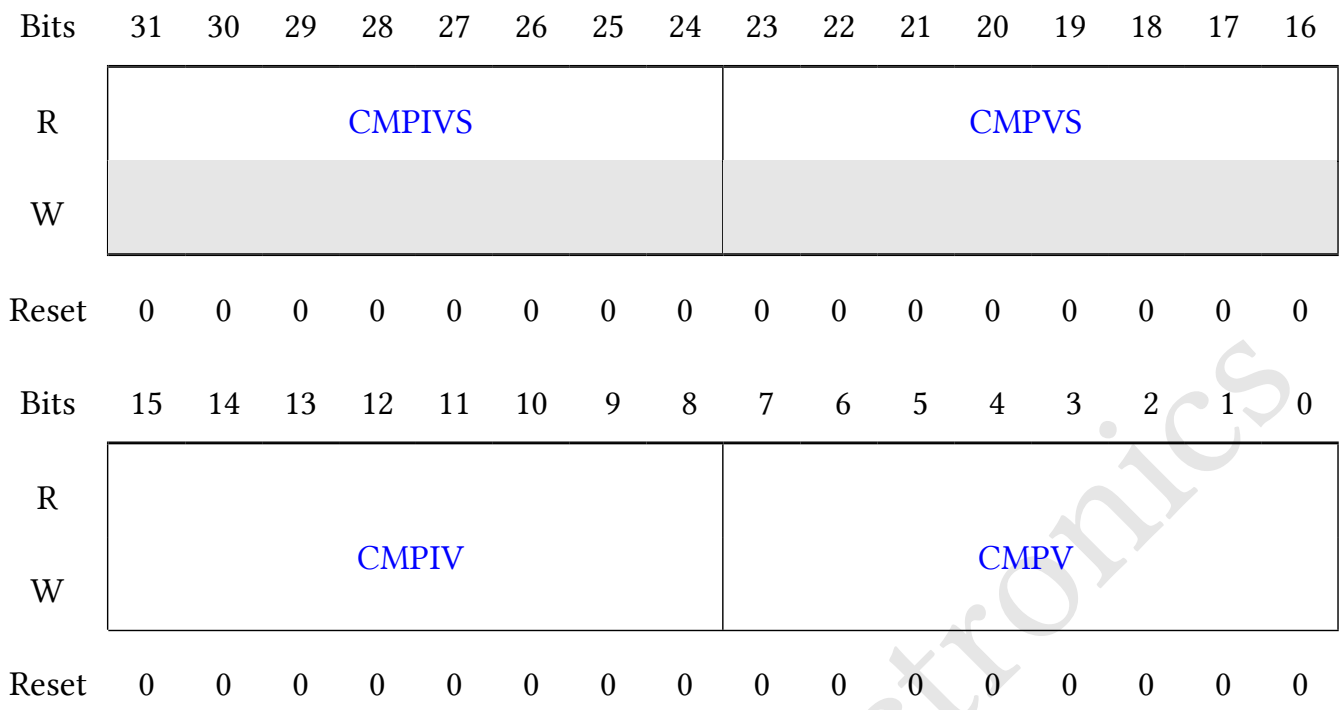


Table 32.3: RTC CMP Register Description

Field	Function
31 - 24 CMPIVS	**Compensation Interval Status** Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CMPIV. If the CMPV does not equal zero then it is decremented once a second. Reading this register at the same time as the seconds counter is incrementing can result in an incorrect value being read.
23 - 16 CMPVS	**Compensation Value Status** Current value used by the compensation logic for the present second interval. Updated once a second if the CMPIVS equals 0 with the contents of the CMPV field. If the CMPIVS does not equal zero then it is loaded with zero (compensation is not enabled for that second increment). Reading this register at the same time as the seconds counter is incrementing can result in an incorrect value being read.
15 - 8 CMPIV	**Compensation Interval Value** Configures the compensation interval in seconds from 1 to 256 to control how frequently the CMPV should adjust the number of 32.768 kHz cycles in each second. The value which is written should be one less than the number of seconds. For example, write zero to configure for a compensation interval of one second. This register is double-buffered and writes do not take effect until the end of the current compensation interval.
7 - 0	**Compensation Value**

Yuntu Microelectronics

Field	Function
CMPV	Configures the number of 32.768 kHz clock cycles in each second, equal to 32,768 CMPV.CMPV is a twos complement sign-extended value. This register is double-buffered and writes do not take effect until the end of the current compensation interval. For example: 00000000b - The number of clock cycles in each second is 32768. 00000001b - The number of clock cycles in each second is 32767. 01111110b - The number of clock cycles in each second is 32642. 01111111b - The number of clock cycles in each second is 32641. 11111111b - The number of clock cycles in each second is 32769. 10000000b - The number of clock cycles in each second is 32896. 10000001b - The number of clock cycles in each second is 32895.

32.2.1.3 RTC ALM Register

RTC Alarm Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ALM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ALM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 32.4: RTC ALM Register Description

Field	Function
31 - 0 ALM	**Alarm Value** If RTC is enabled, when ALM equals the SEC[SEC] and the SEC[SEC] increments, INTF[ALMIF] will set.

32.2.1.4 RTC EN Register

RTC Enable Register

Offset: 10h

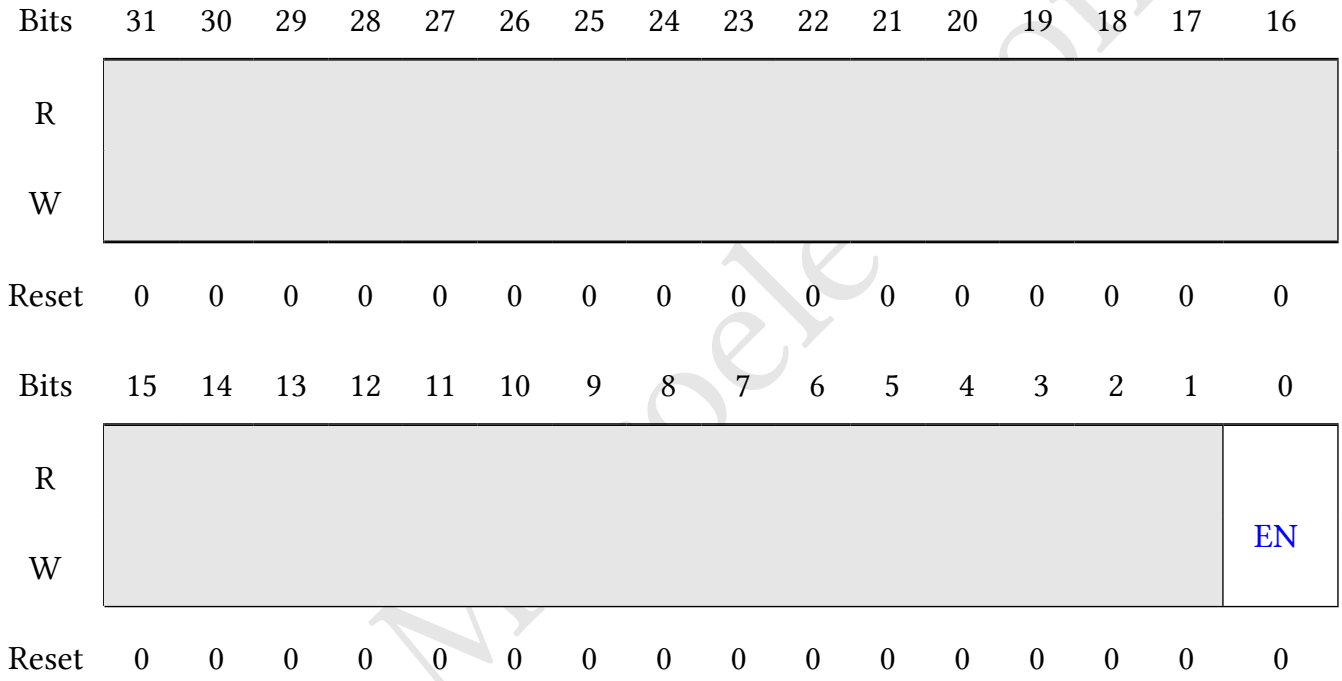


Table 32.5: RTC EN Register Description

Field	Function
0 EN	**RTC Module Enable** When RTC is disabled, SEC is writable. When RTC is enabled, SEC is not writable. 0b - RTC disable 1b - RTC enable

32.2.1.5 RTC CTRL Register

RTC Control Register

Offset: 14h

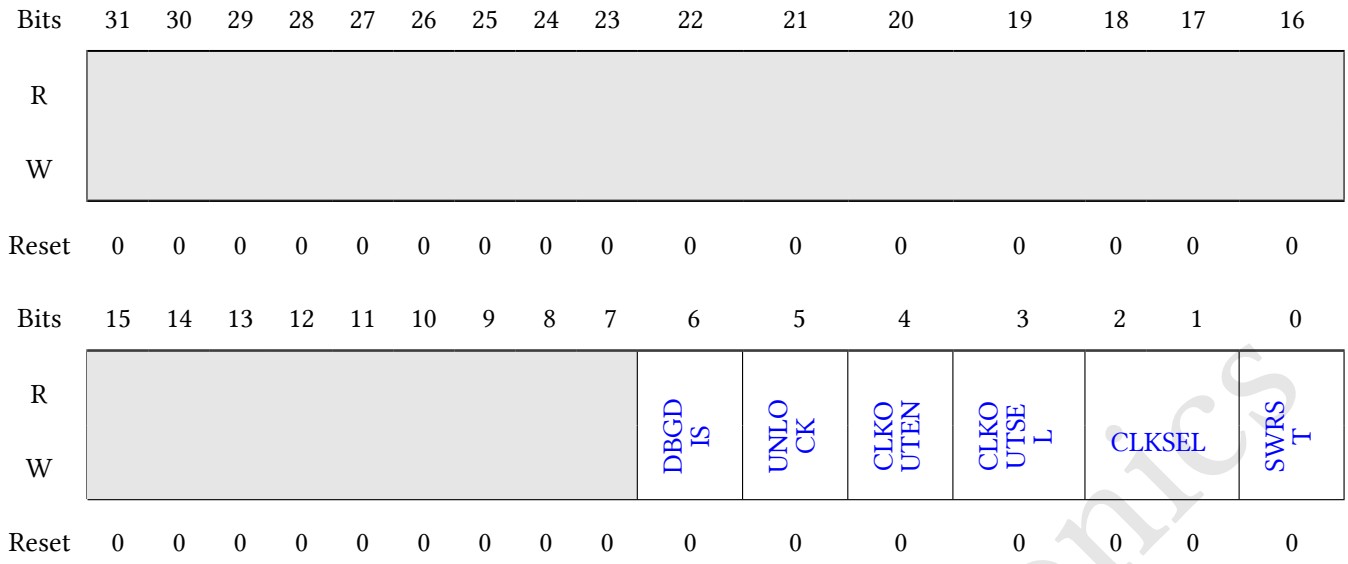


Table 32.6: RTC CTRL Register Description

Field	Function
6 DBGDIS	**Debug Mode Disable** Default 0, run at debug mode. 0b - RTC keeps running in debug mode 1b - RTC stops in debug mode
5 UNLOCK	**Unlock Mode** Unlock register access bit, allows RTC_EN register to be written 0b - Don't allow RTC_EN register to be written 1b - Allow RTC_EN register to be written
4 CLKOUTEN	**RTC Clock Output Enable** Enable the RTC output clock, when CLKOUTEN is not set, rtc output clock is zero. 0b - RTC clock output disable 1b - RTC clock output enable
3 CLKOUTSEL	**RTC Clock Output Select** 0b - RTC clock output is selected by configuration of CTRL[CLKSEL]. 1b - RTC clock output source is 1Hz clock, and actual output clock is determined by INTE[SECOCS], please see INTE[SECOCS]
2 - 1 CLKSEL	**RTC Clock Select** 00b - SXOSC_CLK 01b - RTC_CLKIN 10b - SIRC_CLK_DIV4 11b - Bus clock
0	**Software Reset**

Yuntu Microelectronics

Field	Function
SWRST	0b - No effect 1b - Resets all RTC registers except SWRST which only cleared by POR and by software

32.2.1.6 RTC INTF Register

RTC Interrupt Flag Register

Offset: 18h

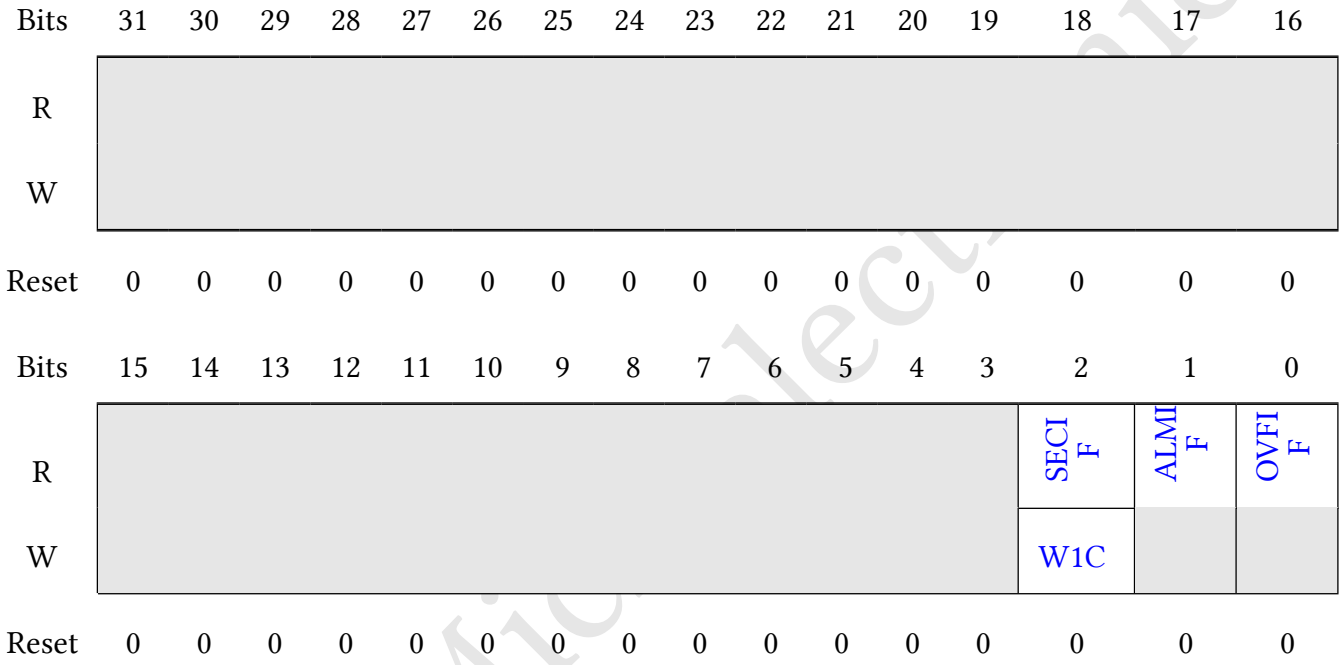


Table 32.7: RTC INTF Register Description

Field	Function
2 SECIF	**Second Interrupt Flag** RTC second interrupt flag. Write 1 and then write 0 to clear this flag. 0b - No second interrupt is pending 1b - The Second interrupt is pending
1 ALMIF	**Alarm Interrupt Flag.** RTC alarm interrupt flag. Write ALM register to clear this flag. 0b - No alarm interrupt is pending 1b - Alarm interrupt is pending
0 OVFIF	**Overflow Interrupt Flag** RTC overflow interrupt flag. Write SEC register to clear this flag. 0b - No overflow interrupt is pending 1b - Overflow interrupt is pending

Yuntu Microelectronics

32.2.1.7 RTC INTE Register

RTC Interrupt Enable Register

Offset: 1ch

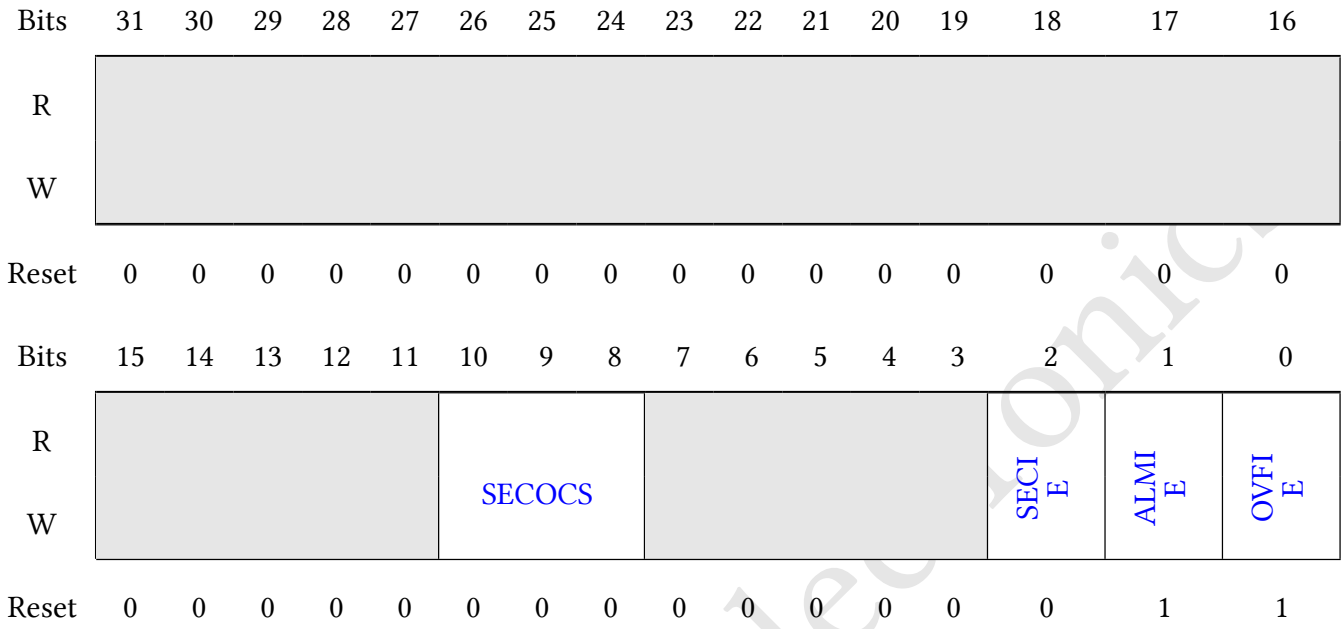


Table 32.8: RTC INTE Register Description

Field	Function
10 - 8 SECOCS	**Second Output Clock Select** RTC Second output clock select 000b - Second output at 1Hz 001b - Second output at 2Hz 010b - Second output at 4Hz 011b - Second output at 8Hz 100b - Second output at 16Hz 101b - Second output at 32Hz 110b - Second output at 64Hz 111b - Second output at 128Hz
2 SECIE	**Second Interrupt Enable** RTC second interrupt enable bit 0b - Second interrupt disable 1b - Second interrupt enable
1 ALMIE	**Alarm Interrupt Enable** 0b - Alarm interrupt disable 1b - Alarm interrupt enable
0	**Overflow Interrupt Enable**

Yuntu Microelectronics

Field	Function
OVFIE	0b - Overflow interrupt disable 1b - Overflow interrupt enable

32.3 Functional Description

32.3.1 General

This section gives detailed information on the internal operation of the module. The RTC is active in all low power modes. The time counter within the RTC is clocked by default from a 32.768KHz clock.

32.3.1.1 Software Reset

Writing 1 to CTRL[SWRST] forces the equivalent of a reset of the RTC module. CTRL[SWRST] is not affected by the software reset and must be cleared by software.

32.3.1.2 Time Counter

The RTC module contains a 32-bit second counter that increments one every second and a 16-bit prescaler register that increments once every 32.768KHz clock. If the SEC register overflow the the INTF[OVFIF] will set and the prescaler register will stop incrementing.

32.3.1.3 Compensation

The CMP register provide a way to accurate correct the time errors. The compensation factor is provided and calculated before write to CMP register. The RTC itself does not have the function to calculate the value of compensation factors.

The compensation factor is calculated with information from 32.768KHz crystal by firmware for operation situations like temperature and voltage, etc.

The compensation logic alters the number of 32.768KHz clock cycles it takes for the prescaler register to overflow and increment the time seconds counter. The time compensation value is used to adjust the number of clock cycles between -127 and +128. Cycles are added or subtracted from the prescaler register when the prescaler register equals 0x3FFF and then increments. The compensation interval is used to adjust the frequency at which the time compensation value is used, that is, from once a second to once every 256 seconds.

Updates to the time compensation register will not take effect until the next time the time seconds register increments and provided the previous compensation interval has expired. When the compensation interval is set to other than once a second then the compensation is applied in the first second interval and the remaining second intervals receive no compensation.

Compensation is disabled by configuring the time compensation register to zero.

32.3.1.4 Register Lock

When CTRL[UNLOCK] bit is cleared, Enable Register cannot be written.

Yuntu Microelectronics

32.3.2 Interrupts

RTC module supports three interrupt. Please refer to the table of interrupt vectors. RTC interrupts is asserted whenever a status flag and the corresponding interrupt enable bit are both set. The RTC interrupt can be used to wakeup the chip from any low-power mode.

If RTC is enabled, when ALMR[ALM] equals the RTCSEC[SEC] and the RTCSEC[SEC] increments, RTCS[ALMIF] will set and Alarm interrupt is generated. When the RTC is enabled and overflows, Overflow interrupt is generated.

If RTC is enabled, when INTE[SECIE] is set,Second interrupt is generated once a second.The frequency of the second Interrupt defaults to 1 Hz, but can be configured by INTE[SECIE].

32.3.3 Operation Modes

RTC is implemented in always-on domain, it keeps working if 32.768kHZ crystal oscillator is enabled even in Deepsleep, Standby or Powerdown mode.

If RTC is required to keep working, it needs to configure RTC, clock sources and WKU(just for Powerdown mode) before entering low power modes.

Yuntu Microelectronics

Chapter 33

Programmable Trigger Unit (PTU)

33.1 Chip-specific PTU Information

33.1.1 Instantiation Information

YTM32B1HA0x contains two 8-channel PTUs.

33.2 Introduction

33.2.1 Overview

The Programmable Trigger Unit (PTU) provides controllable delays from either an internal or external trigger. The PTU provides programmable delays from internal triggers, external triggers or software trigger to the ADCs. The PTU can provides pulse outputs that are used as the sample window of ACMP block.

33.2.2 Features

- Contain two PTU instances
- 8 Configurable PTU channels for ADC hardware trigger
- Each trigger outputs can be enabled/disabled independently
- Configurable delay per pre-trigger output
- Support Bypass of pre-trigger delay
- Support software trigger source
- Support Continuous Mode
- Support interrupt delay
- Support pulse output as ACMP's sample window

33.2.3 Block Diagram

The following figure shows the PTU block diagram.

Yuntu Microelectronics

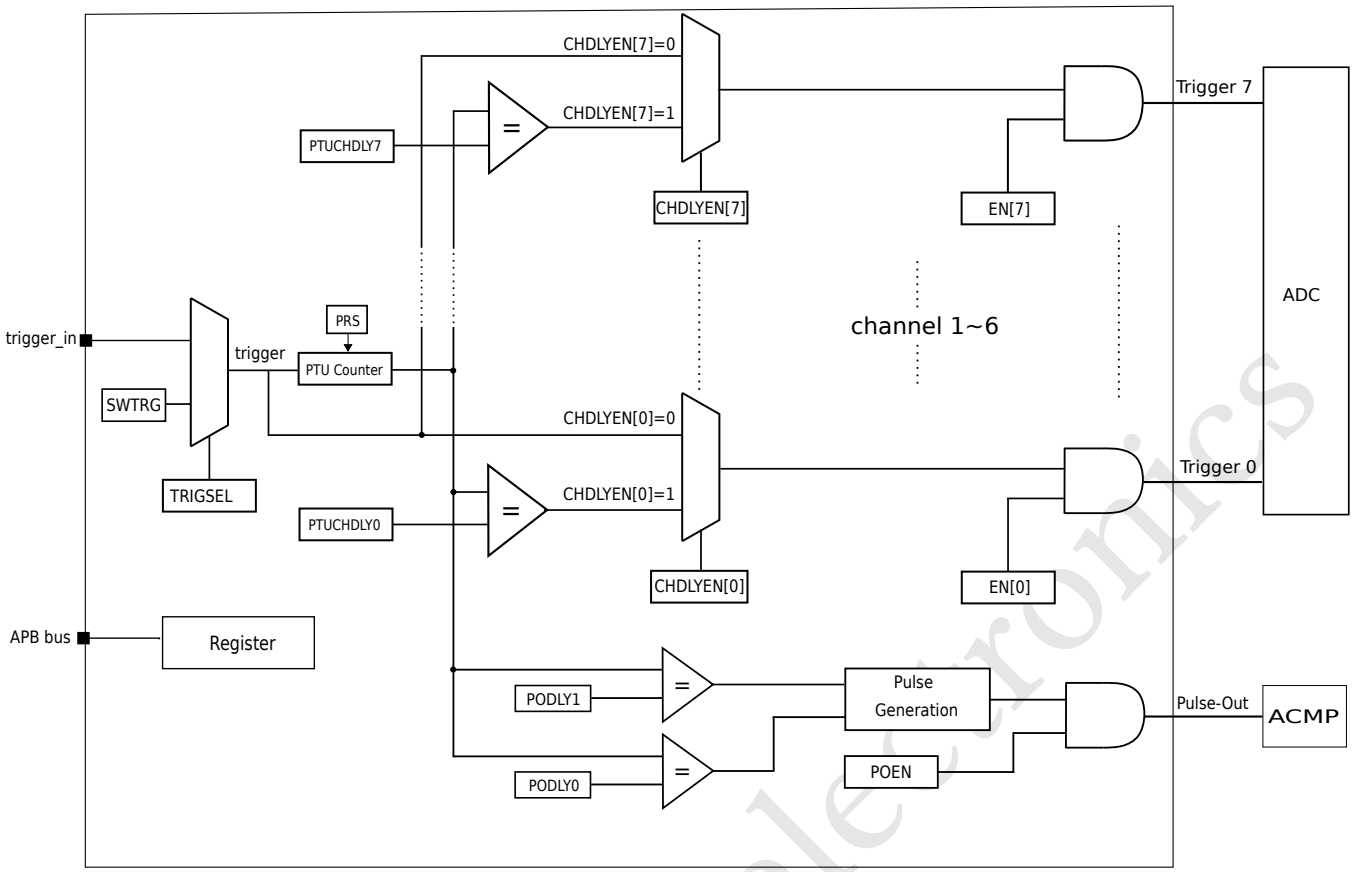


Figure 33.1: Block Diagram of PTU

33.3 Register Definition

This subsection provides a description of the memory map and registers.

33.3.1 PTU Memory Map

PTU0 base address 40118000h
PTU1 base address 4011c000h

Table 33.1: PTU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Prescaler Select Register (PRS)	32	RW	00000000h
8h	Interrupt Enable Register (INTE)	32	RW	00000000h
ch	Interrupt Flag Register (STS)	32	RW	00000000h
10h	Modulus Register (MOD)	32	RW	0000FFFFh
14h	Counter Current Value Register (CNT)	32	RW	00000000h
18h	Interrupt Delay Register (INTDLY)	32	RW	0000FFFFh
20h	Channel Control Register (CHCTRL)	32	RW	00000000h
24h	Channel Status Register (CHSTS)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
28h	Channel Delay n Register (CHDLYn0)	32	RW	00000000h
2ch	Channel Delay n Register (CHDLYn1)	32	RW	00000000h
30h	Channel Delay n Register (CHDLYn2)	32	RW	00000000h
34h	Channel Delay n Register (CHDLYn3)	32	RW	00000000h
38h	Channel Delay n Register (CHDLYn4)	32	RW	00000000h
3ch	Channel Delay n Register (CHDLYn5)	32	RW	00000000h
40h	Channel Delay n Register (CHDLYn6)	32	RW	00000000h
44h	Channel Delay n Register (CHDLYn7)	32	RW	00000000h
190h	Pulse-Out Enable Register (POEN)	32	RW	00000000h
194h	Pulse-Out Delay Register (PODLY)	32	RW	00h

33.3.1.1 PTU CTRL Register

Control Register

Offset: 0h

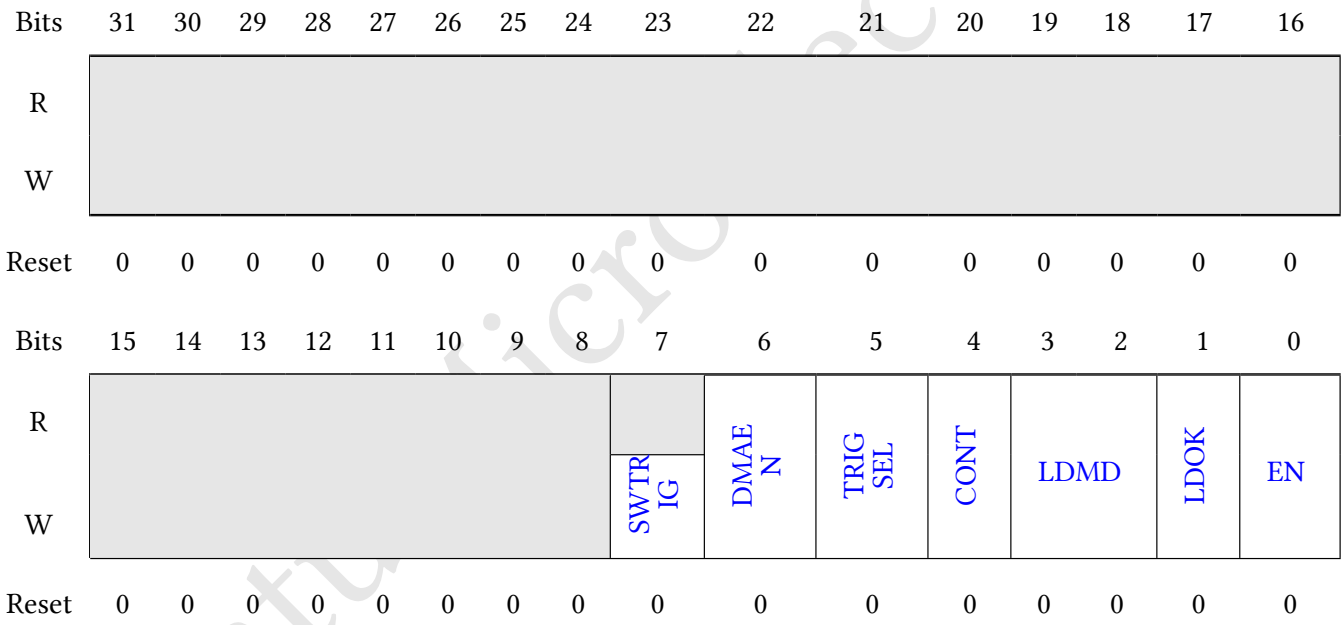


Table 33.2: PTU CTRL Register Description

Field	Function
7 SWTRIG	**Software Trigger** Trigger PTU by software. 0b - No effects. 1b - If PTU is enabled and the trigger input source is software trigger, write 1 to this bit to generate a trigger input event.
6	**DMA Enable**

Field	Function
DMAEN	This bit is set and cleared by software to enable the generation of DMA requests. This allows using the DMA controller to manage automatically the converted data. 0b - DMA disabled. 1b - DMA enabled.
5 TRIGSEL	**Trigger Selection** This bit selects the trigger input source for the PTU. 0b - External trigger is selected as the trigger input source. 1b - Software trigger is selected as the trigger input source.
4 CONT	**Continuous Mode Enable** Enables continuous mode. 0b - PTU operation in normal mode. 1b - PTU operation in continuous mode.
3 - 2 LDMD	**Load Mode Select** Selects the mode to load the MOD, INTDLY, CHDLYn and PODLY registers, after 1 is written to LDOK. 00b - The internal registers are loaded with the values from their buffers immediately after 1 is written to LDOK. 01b - The internal registers are loaded with the values from their buffers when the PTU counter reaches the MOD register value after 1 is written to LDOK. 10b - The internal registers are loaded with the values from their buffers when a trigger input event is detected after 1 is written to LDOK. 11b - The internal registers are loaded with the values from their buffers when either the PTU counter reaches the MOD register value or a trigger input event is detected after 1 is written to LDOK.
1 LDOK	**Load OK** LDOK can be written only when CTRL[EN] is set or it can be written at the same time with CTRL[EN] being written to 1. it is automatically cleared when the values in buffers are loaded into the internal registers or the CTRL[EN] is cleared. 0b - Writing 0 to it has no effect. 1b - Loading internal registers as selected load mode as value of CTRL[LDMD]
0	**PTU Enable**

Yuntu Microelectronics

Field	Function
EN	Writing 1 to this bit updates the internal registers of MOD, INTDLY and CHDLYn with the values written to their buffers. The MOD, INTDLY and CHDLYn will take effect immediately if LDMOD is cleared; or when the counter rolls over in Continuous mode, or a rising edge is detected on the selected trigger input source, or software trigger is selected and CTRL[SWTRIG] is written with 1, if CTRL[LDMD] is set. After 1 is written to LDOK bit, the values in the buffers of the above registers are not effective and the buffers cannot be written until the values in buffers are loaded into their internal registers. 0b - PTU disabled. The Counter is off. 1b - PTU enabled.

33.3.1.2 PTU PRS Register

Prescaler select register

Offset: 4h

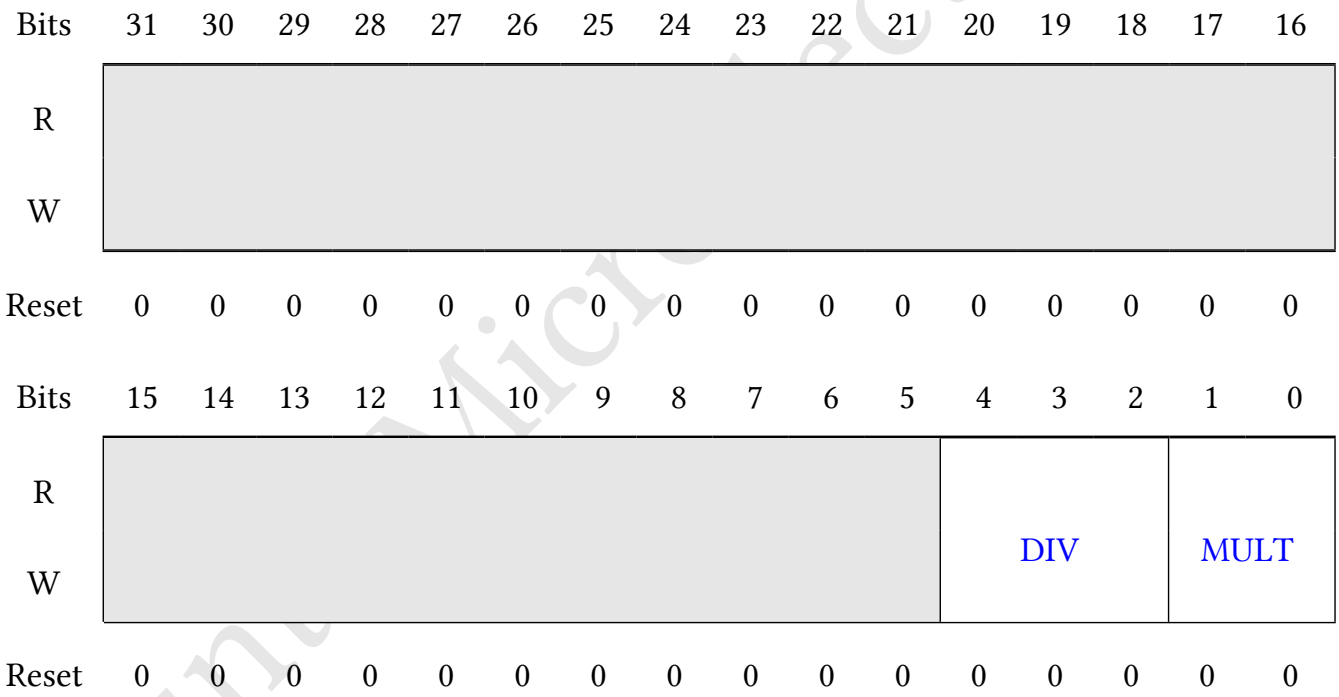


Table 33.3: PTU PRS Register Description

Field	Function
4 - 2	**Divider Select**

Yuntu Microelectronics

Field	Function
DIV	000b - Counting uses the peripheral clock divided by multiplication factor selected by PRS[MULT] 001b - Counting uses the peripheral clock divided by twice of the multiplication factor selected by PRS[MULT] 010b - Counting uses the peripheral clock divided by 4 times of the multiplication factor selected by PRS[MULT] 011b - Counting uses the peripheral clock divided by 8 times of the multiplication factor selected by PRS[MULT] 100b - Counting uses the peripheral clock divided by 16 times of the multiplication factor selected by PRS[MULT] 101b - Counting uses the peripheral clock divided by 32 times of the multiplication factor selected by PRS[MULT] 110b - Counting uses the peripheral clock divided by 64 times of the multiplication factor selected by PRS[MULT] 111b - Counting uses the peripheral clock divided by 128 times of the multiplication factor selected by PRS[MULT]
1 - 0 MULT	**Multiplication Factor Select for Prescaler** This bit selects the multiplication factor of the prescaler divider for the counter clock. 00b - Multiplication factor is 1. 01b - Multiplication factor is 10. 10b - Multiplication factor is 20. 11b - Multiplication factor is 40.

Yuntu Microelectronics

33.3.1.3 PTU INTE Register

Interrupt enable register

Offset: 8h

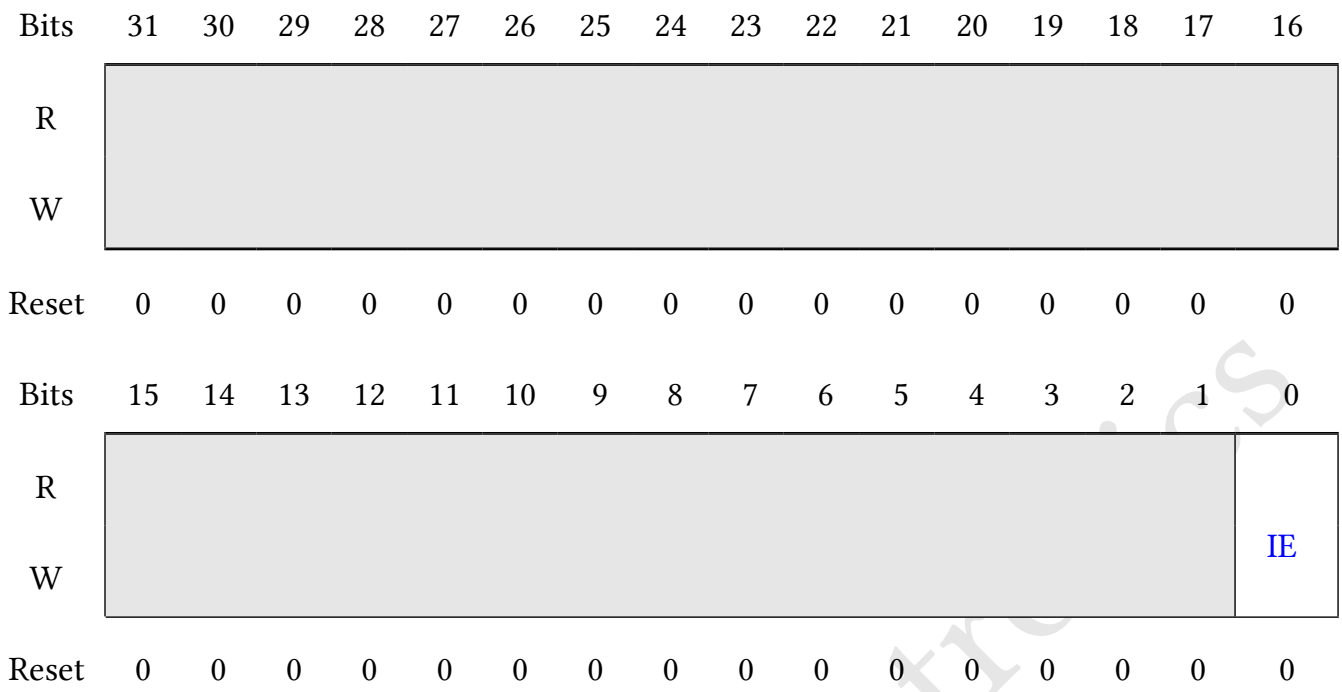


Table 33.4: PTU INTE Register Description

Field	Function
0 IE	**PTU Interrupt Enable** This bit enables the PTU interrupt. When this bit is set, STS[IF] generates a PTU interrupt to the CPU. 0b - PTU interrupt disabled. 1b - PTU interrupt enabled.

Yuntu Microelectronics

33.3.1.4 PTU STS Register

Interrupt flag register

Offset: ch

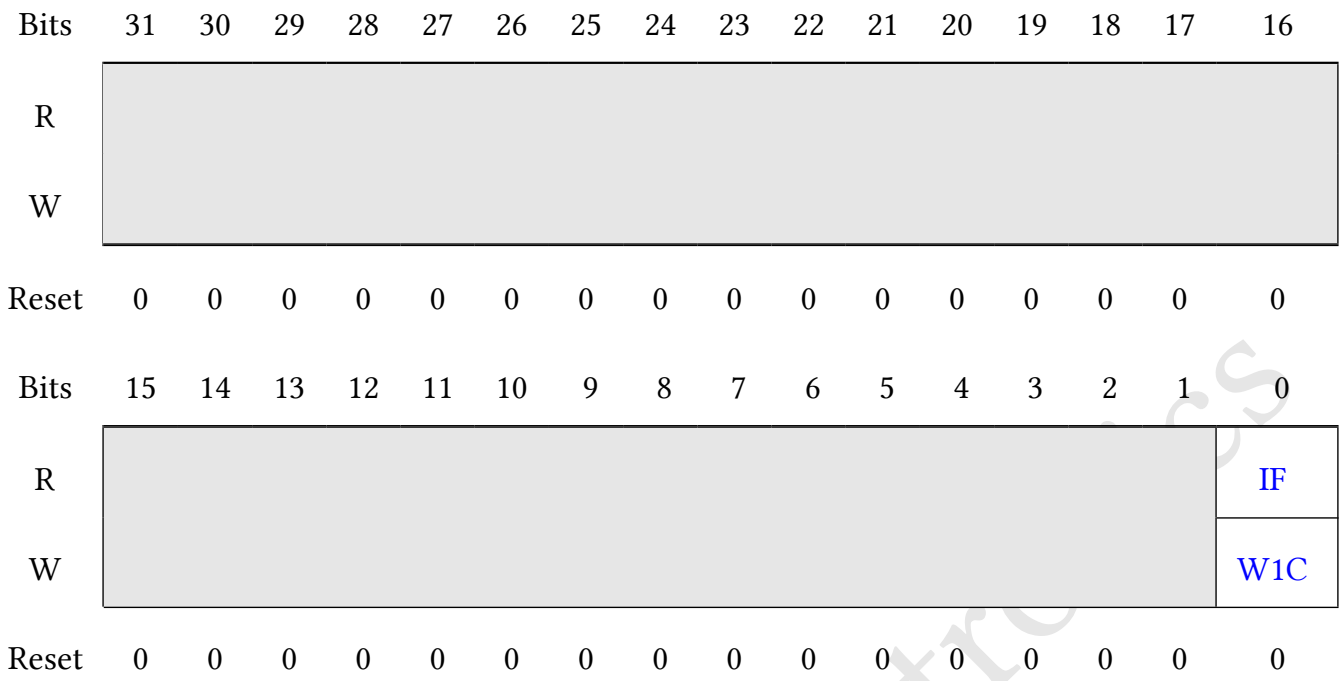


Table 33.5: PTU STS Register Description

Field	Function
0 IF	**PTU Interrupt Flag** This bit is set when the counter value is equal to the internal buffer of the INTDLY register. Writing 1 clear this bit. 0b - This bit is 0 before the counter value is equal to the internal buffer of the INTDLY register. 1b - This bit is set when the counter value is equal to the internal buffer of the INTDLY register.

Yuntu Microelectronics

33.3.1.5 PTU MOD Register

Modulus Register

Offset: 10h

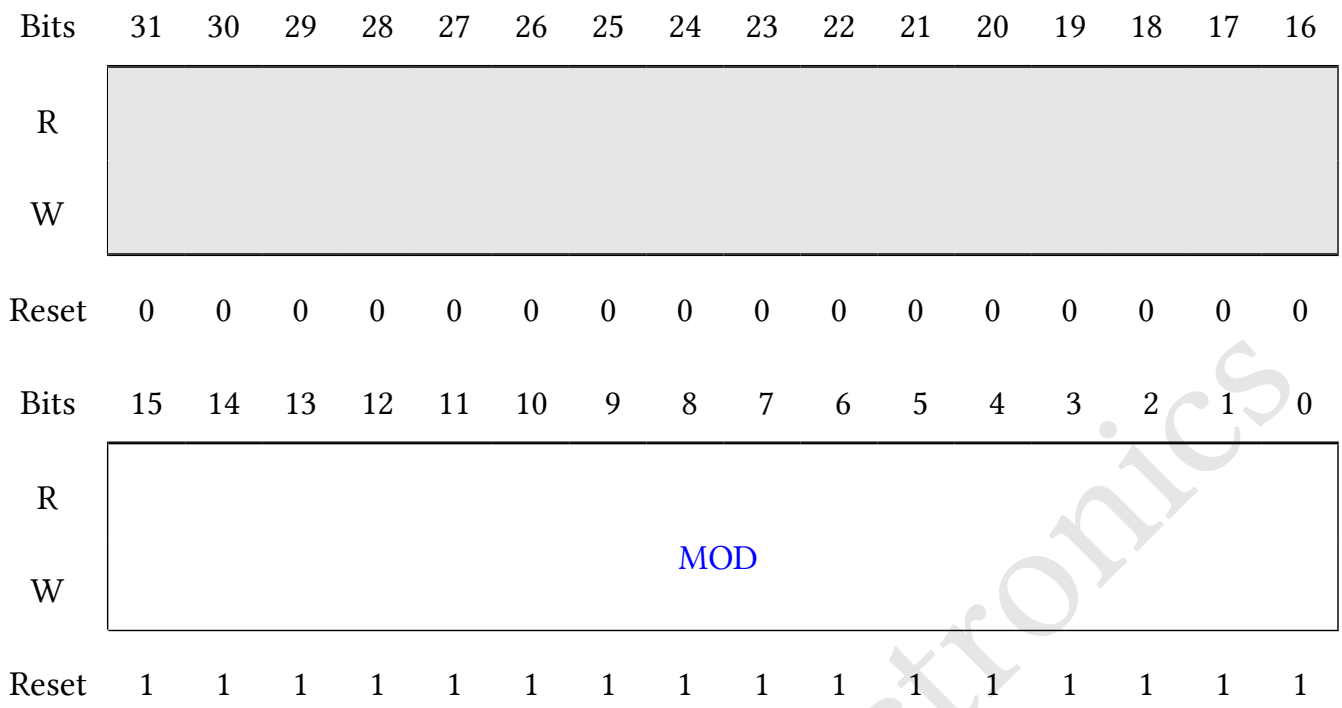


Table 33.6: PTU MOD Register Description

Field	Function
15 - 0 MOD	**Modulus** This modulus register controls the counter operation, when the counter reaches modulus value, it returns to zero. It always operate in Continuous mode. Reading this register returns the effective value of counter modulus.

Yuntu Microelectronics

33.3.1.6 PTU CNT Register

Counter current value

Offset: 14h

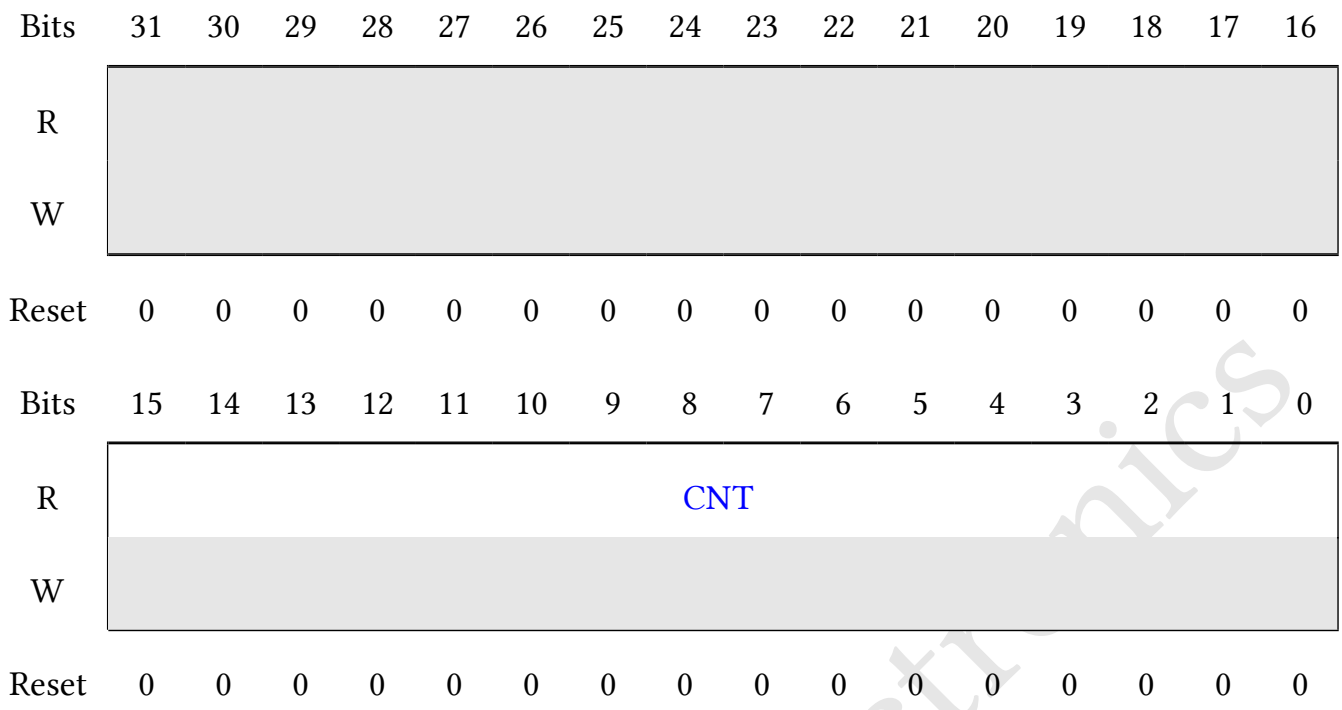


Table 33.7: PTU CNT Register Description

Field	Function
15 - 0 CNT	**PTU Counter Current Value** Reading this register returns the current value of PTU counter.

Yuntu Microelectronics

33.3.1.7 PTU INTDLY Register

Interrupt delay register

Offset: 18h

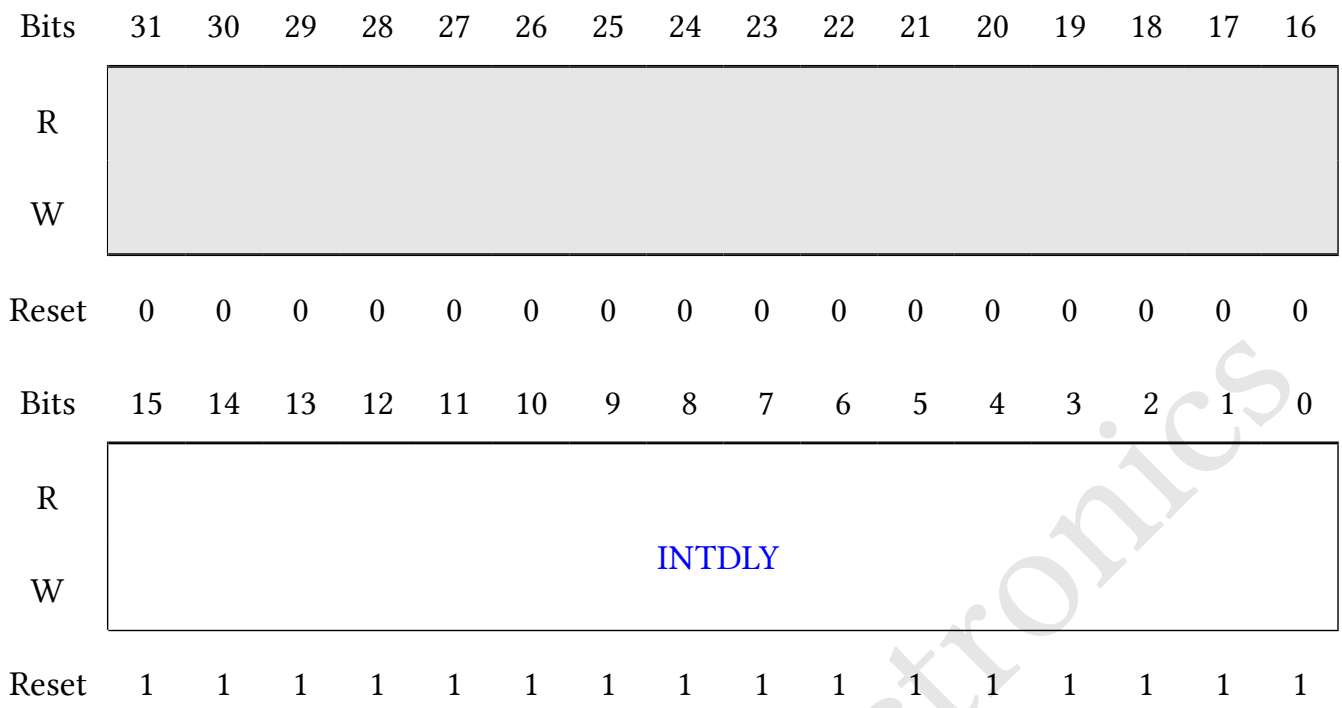


Table 33.8: PTU INTDLY Register Description

Field	Function
15 - 0 INTDLY	**PTU Interrupt Delay** This register specify the delay of generating PTU interrupt. A PTU interrupt is generated when the counter reaches the value of INTDLY[INTDLY].

Yuntu Microelectronics

33.3.1.8 PTU CHCTRL Register

Channel control register

Offset: 20h

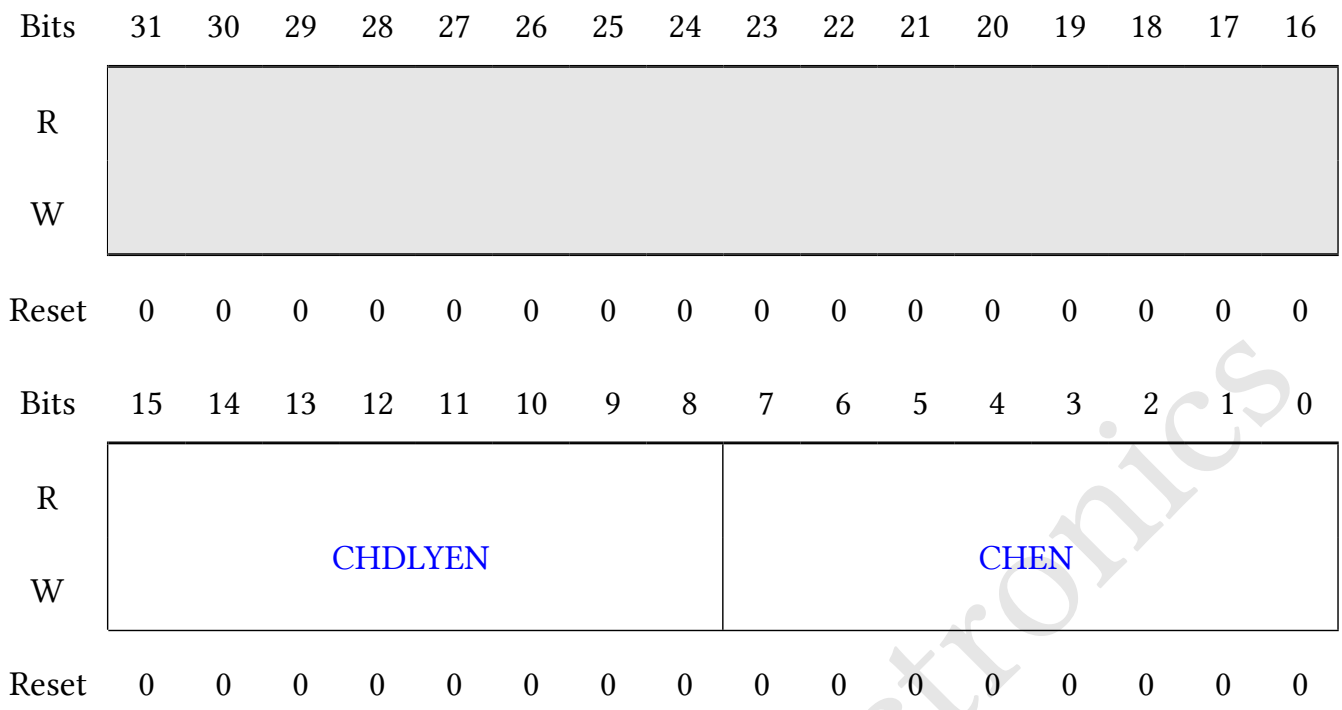


Table 33.9: PTU CHCTRL Register Description

Field	Function
15 - 8 CHDLYEN	**Channel Trigger n Scale Delay Disable** 00000000b (0x00) - Channel trigger number n scale delay disable. 00000001b (0x01) - Chanel trigger number n scale delay enable.
7 - 0 CHEN	**Channel Trigger n Enable** Channel trigger number n enable 00000000b (0x00) - trigger number n disabled. 00000001b (0x01) - trigger number n enabled.

Yuntu Microelectronics

33.3.1.9 PTU CHSTS Register

Channel Status Register

Offset: 24h

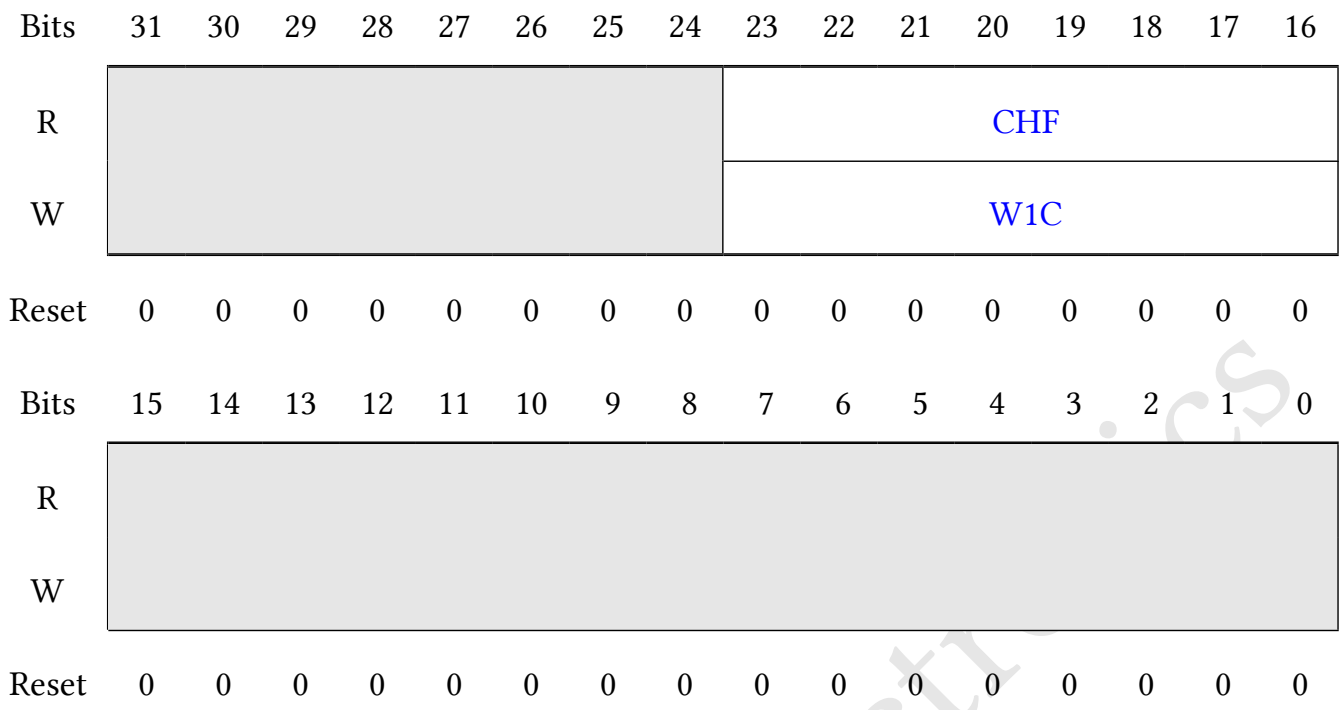


Table 33.10: PTU CHSTS Register Description

Field	Function
23 - 16 CHF	**PTU Channel Flags** The CHF[n] bit is set when the PTU counter matches the CHDLYn. Write 1 to clear these bits. 00000000b (0x00) - Before the PTU counter matches the CHDLYn, CHF[n] bit is 0. 00000001b (0x01) - When the PTU counter matches the CHDLYn, CHF[n] bit is set.

33.3.1.10 PTU CHDLYn Register

PTU Channel Delay n Register

Offset: 28h + n * 4h

Yuntu Microelectronics

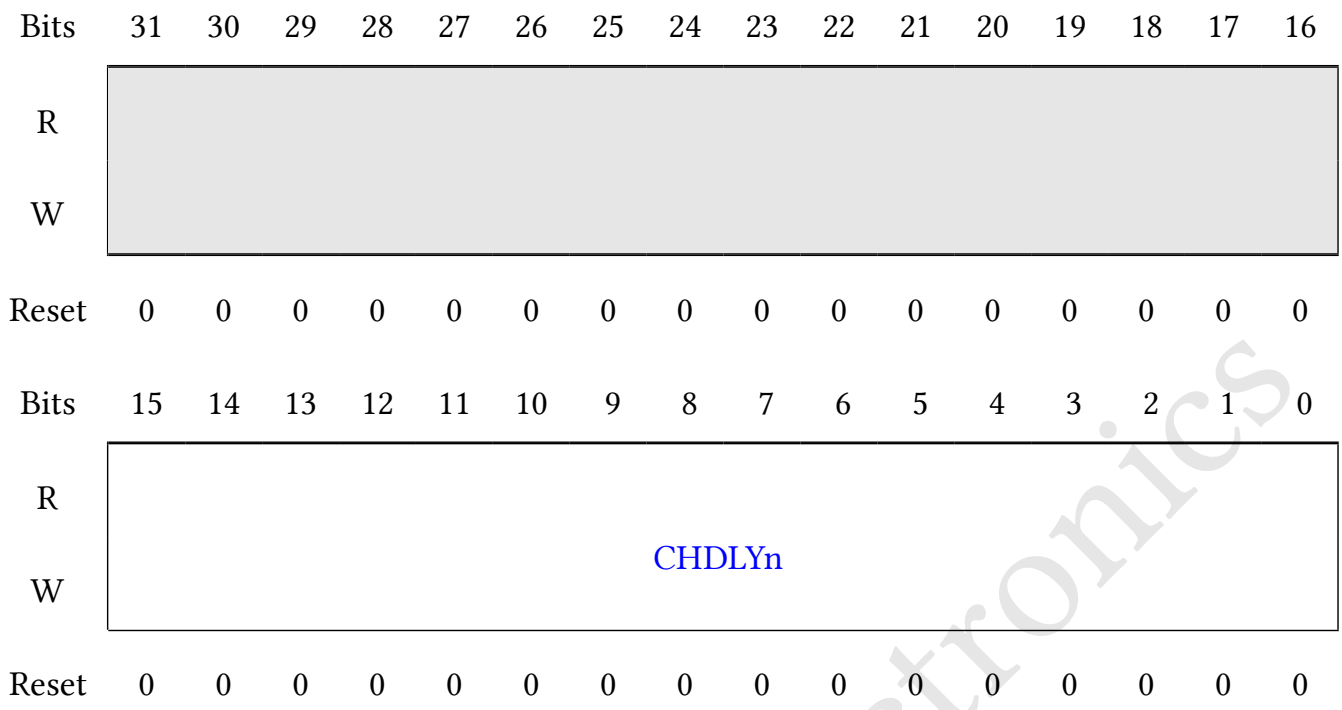


Table 33.11: PTU CHDLYn Register Description

Field	Function
15 - 0 CHDLYn	**Channel Delay n** This register specify the delay value for each of channels to generate the triggers which will start a ADC conversion when PTU counter reaches CHDLYn[CHDLYn] value. Reading this register returns the effective value.

Yuntu Microelectronics

33.3.1.11 PTU POEN Register

PTU pulse-out enable register

Offset: 190h

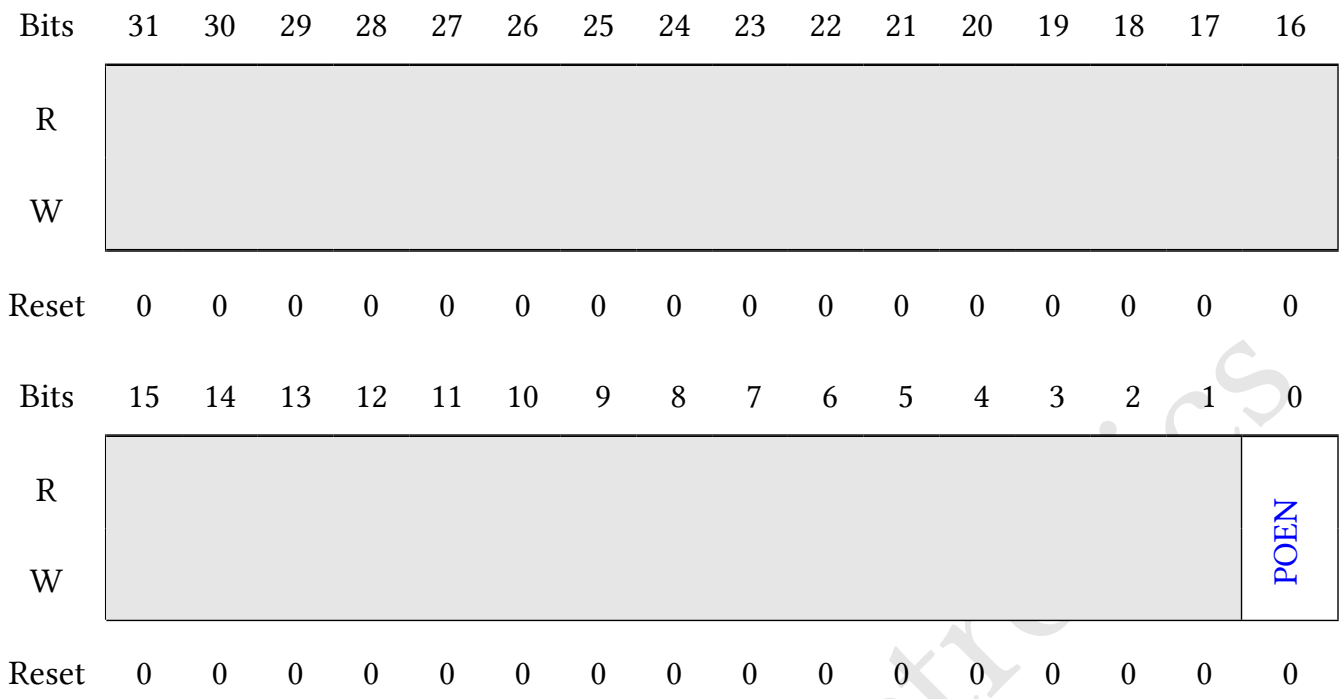


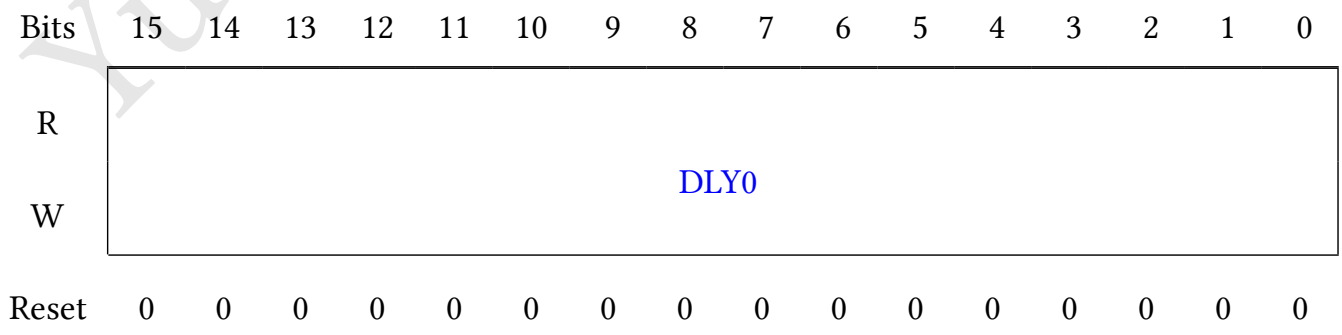
Table 33.12: PTU POEN Register Description

Field	Function
0 POEN	**PTU Pulse-out Enable** This bit enable the pulse output. 0b - PTU pulse-out disabled. 1b - PTU pulse-out enabled.

33.3.1.12 PTU PODLY_PODLY0 Register

These bits specify the delay 0 value for the PTU Pulse-Out. Pulse-out goes high when the PTU counter is equal to the PODLY0. Reading these bits returns the value of internal register that is effective for the current PTU cycle.

Offset: 194h



Yuntu Microelectronics

Table 33.13: PTU PODLY_PODLY0 Register Description

Field	Function
15 - 0 DLY0	**DLY0**

33.3.1.13 PTU PODLY_PODLY1 Register

These bits specify the delay 1 value for the PTU pulse-out. Pulse-out goes low when the PTU counter is equal to the PODLY1. Reading these bits returns the value of internal register that is effective for the current PTU cycle.

Offset: 196h

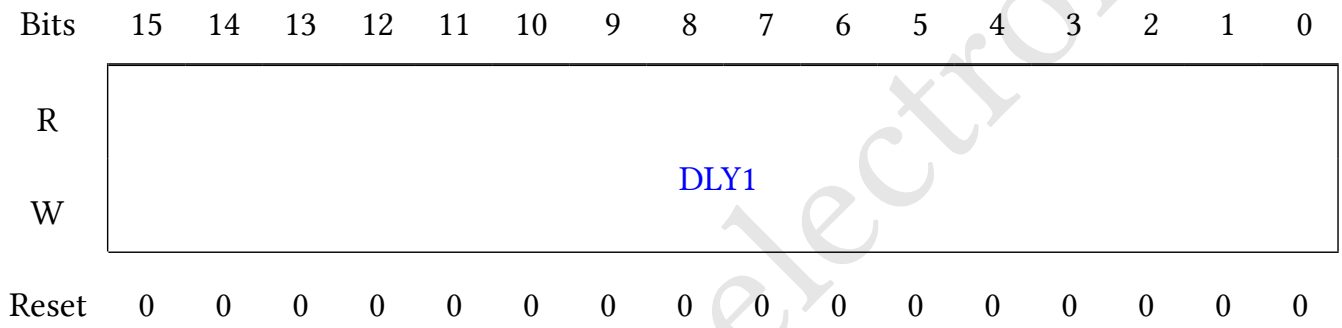


Table 33.14: PTU PODLY_PODLY1 Register Description

Field	Function
15 - 0 DLY1	**DLY1**

33.4 Functional Description

33.4.1 PTU Counter

PTU counter is shared by input delay compare logic, ADC channel trigger logic and ACMP pulse out logic with same time base. The counter increments on every output of prescaler output.

33.4.2 PTU Trigger

When PTU is enabled, the PTU counter start count when a trigger input event happens. The PTU counter's output is compared with INTDLY and CHDLYn values. The trigger input event can be software trigger or hardware trigger which is controlled by CTRL[TRIGSEL] bit.

When PTU counter value reaches INTDLY, an input delay interrupt is generated if INTE[IE] is set.

When PTU counter value reaches CHDLYn, trigger n is generated to start ADC conversion.

Yuntu Microelectronics

33.4.3 PTU Pulse Out

PTU generates pulse out for ACMP operation window. When PTU counter reaches PODLY[PODLY0], pulse out signal goes high. When PTU counter reaches PODLY[PODLY1], pulse out signal goes low.

33.4.4 Operation Modes

PTU supports the chip operation modes of operation as follows:

- Active: The PTU operates normally.
- Sleep: The PTU continues to operate normally and can be configured to generate trigger, pulse output and input delay interrupt.
- Deepsleep, Standby, Powerdown: The PTU stops operation as it is clocked by fast bus clock which is stopped in Deepsleep, Standby and Powerdown mode.

33.5 Glossary

PTU	Programmable Trigger Unit
ADC	Analog-to-Digital Converter
ACMP	Analog Comparator
TMU	Trigger Multiplexer

Yuntu Microelectronics

Part VIII

Analog

Yuntu Microelectronics

Chapter 34

Analog-to-Digital Converter (ADC)

34.1 Chip-specific ADC information

34.1.1 Instantiation Information

YTM32B1HA0x contains two 12-bit ADC. The 12-bit ADC is a successive approximation analog-to-digital converter.

- ADC0: 32 external input channels and 8 internal channels
- ADC1: 32 external input channels

34.1.2 Signal Description

Table 34.1: ADC0 Signal Table

Channel selection	Signal Name	Description
0	ADC0_SE0	The analog channel 0 for external signal
1	ADC0_SE1	The analog channel 1 for external signal
2	ADC0_SE2	The analog channel 2 for external signal
3	ADC0_SE3	The analog channel 3 for external signal
4	ADC0_SE4	The analog channel 4 for external signal
5	ADC0_SE5	The analog channel 5 for external signal
6	ADC0_SE6	The analog channel 6 for external signal
7	ADC0_SE7	The analog channel 7 for external signal
8	ADC0_SE8	The analog channel 8 for external signal
9	ADC0_SE9	The analog channel 9 for external signal
10	ADC0_SE10	The analog channel 10 for external signal
11	ADC0_SE11	The analog channel 11 for external signal
12	ADC0_SE12	The analog channel 12 for external signal
13	ADC0_SE13	The analog channel 13 for external signal
14	ADC0_SE14	The analog channel 14 for external signal
15	ADC0_SE15	The analog channel 15 for external signal
16	ADC0_SE16	The analog channel 16 for external signal
17	ADC0_SE17	The analog channel 17 for external signal
18	ADC0_SE18	The analog channel 18 for external signal
19	ADC0_SE19	The analog channel 19 for external signal

Yuntu Microelectronics

Table 34.1 continued from previous page

Channel selection	Signal Name	Description
20	ADC0_SE20	The analog channel 20 for external signal
21	ADC0_SE21	The analog channel 21 for external signal
22	ADC0_SE22	The analog channel 22 for external signal
23	ADC0_SE23	The analog channel 23 for external signal
24	ADC0_SE24	The analog channel 24 for external signal
25	ADC0_SE25	The analog channel 25 for external signal
26	ADC0_SE26	The analog channel 26 for external signal
27	ADC0_SE27	The analog channel 27 for external signal
28	ADC0_SE28	The analog channel 28 for external signal
29	ADC0_SE29	The analog channel 29 for external signal
30	ADC0_SE30	The analog channel 30 for external signal
31	ADC0_SE31	The analog channel 31 for external signal
32	Temperature senser	Temperature senser voltage
33	PMC VREF	PMC 1.2V reference voltage
34	VREFH	High reference voltage
35	VREFL	Low reference voltage
36	-	Reserve
37	VDD	Supply VDD
38	VSS	Supply VSS
39	VDD25	Supply VDD25

Table 34.2: ADC1 Signal Table

Channel selection	Signal Name	Description
0	ADC1_SE0	The analog channel 0 for external signal
1	ADC1_SE1	The analog channel 1 for external signal
2	ADC1_SE2	The analog channel 2 for external signal
3	ADC1_SE3	The analog channel 3 for external signal
4	ADC1_SE4	The analog channel 4 for external signal
5	ADC1_SE5	The analog channel 5 for external signal
6	ADC1_SE6	The analog channel 6 for external signal
7	ADC1_SE7	The analog channel 7 for external signal
8	ADC1_SE8	The analog channel 8 for external signal
9	ADC1_SE9	The analog channel 9 for external signal
10	ADC1_SE10	The analog channel 10 for external signal
11	ADC1_SE11	The analog channel 11 for external signal
12	ADC1_SE12	The analog channel 12 for external signal
13	ADC1_SE13	The analog channel 13 for external signal
14	ADC1_SE14	The analog channel 14 for external signal
15	ADC1_SE15	The analog channel 15 for external signal
16	ADC1_SE16	The analog channel 16 for external signal
17	ADC1_SE17	The analog channel 17 for external signal
18	ADC1_SE18	The analog channel 18 for external signal
19	ADC1_SE19	The analog channel 19 for external signal
20	ADC1_SE20	The analog channel 20 for external signal

Yuntu Microelectronics

Table 34.2 continued from previous page

Channel selection	Signal Name	Description
21	ADC1_SE21	The analog channel 21 for external signal
22	ADC1_SE22	The analog channel 22 for external signal
23	ADC1_SE23	The analog channel 23 for external signal
24	ADC1_SE24	The analog channel 24 for external signal
25	ADC1_SE25	The analog channel 25 for external signal
26	ADC1_SE26	The analog channel 26 for external signal
27	ADC1_SE27	The analog channel 27 for external signal
28	ADC1_SE28	The analog channel 28 for external signal
29	ADC1_SE29	The analog channel 29 for external signal
30	ADC1_SE30	The analog channel 30 for external signal
31	ADC1_SE31	The analog channel 31 for external signal

34.1.3 Clock Description

Table 34.3: ADC Clock Table

Clock Name	Description
Bus Clock	Bus clock used for registers access
Function Clock	Function clock used for ADC sample and conversion

34.1.4 DMA Description

Table 34.4: ADC DMA Table

No.	Request Source	Description
42	The amount of data in FIFO exceeds the threshold CFG0[WM] of ADC0	Generate DMA request to read DATA and avoid DATA overflow when FIFO is full
43	The amount of data in FIFO exceeds the threshold CFG0[WM] of ADC1	Generate DMA request to read DATA and avoid DATA overflow when FIFO is full

34.1.5 Interrupt Description

Table 34.5: ADC Interrupts

Interrupt event	Event flag	Enable Control Bit
ADC ready	STS[ADRDY]	INTE[ADRDYIE]
End of conversion	STS[EOC]	INTE[EOCIE]
End of sequence of conversions	STS[EOSEQ]	INTE[EOSEQIE]
End of sampling phase	STS[EOSMP]	INTE[EOSMPIE]
Overrun	STS[OVR]	INTE[OVRIE]
Watchdog status bit is set	STS[WDx](x=0,1,2,3)	INTE[WDIE]

Yuntu Microelectronics

34.1.6 Power mode Description

Table 34.6: ADC Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Keeps working when function clock available
Deepsleep	Keeps working when function clock available
Standby	Not available
Powerdown	Not available

34.2 Introduction

34.2.1 Overview

The 12-bit ADC is a successive approximation analog-to-digital converter. A/D conversion of the various channels can be performed in single, continuous or discontinuous mode. The results of the ADC are stored in a 32bit x 16depth FIFO, and the data format is right-aligned or left aligned.

The watchdog feature allows the application to detect if the input voltage goes inside or outside the user-defined thresholds.

An efficient low power mode is implemented to allow very low consumption at low frequency.

34.2.2 Features

The features of the Analog-to-Digital Converter (ADC) module are listed below:

- Contain two ADC instances
  - ADC0 supports up to 32 external analog input channels, and 8 internal channels
  - ADC1 supports up to 32 external analog input channels
- Support 12-bit, 10-bit, 8-bit, and 6-bit single-ended configurable resolution
- Up to 2Msps for 12-bit resolution conversion performance
- Support DMA and conversion result FIFO with watermark
- Support multiple conversion modes
  - Single mode
  - Continuous mode
  - Discontinuous mode
- Support software/hardware trigger for ADC start conversion
- Support two power saving modes
  - Wait mode: prevent ADC overrun when FIFO is full
  - Auto off mode: automatic control ADC power off
- Support watchdog for conversion result monitoring
- Support automatic calibration
- Support interrupt generate
  - Ready for conversion
  - End of sampling
  - End of conversion
  - End of sequence conversion
  - Overrun event

Yuntu Microelectronics

  - Watchdog event
- Support work and wake up when the chip under low power mode
- Support self-test mode

34.2.3 Block diagram

The following figure shows the ADC block diagram.

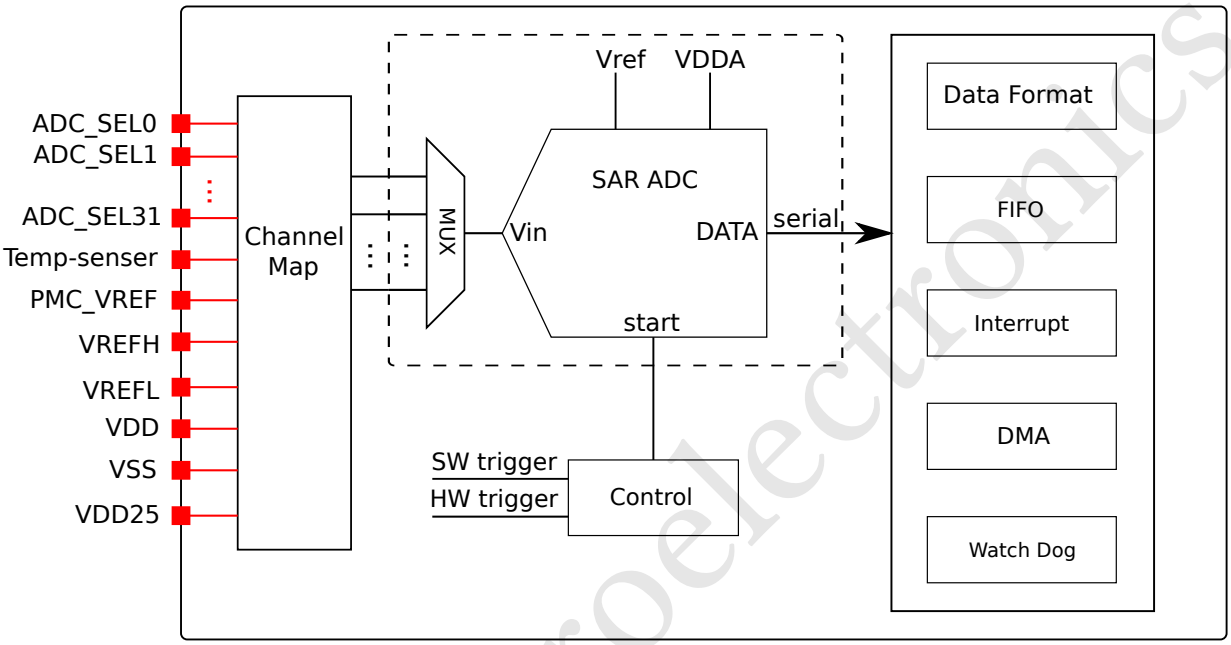


Figure 34.1: Block Diagram of ADC

34.3 Register Definition

The ADC registers are used to control ADC module and display the current working status. In addition, registers are used to configure functional modes, such as single, continuous, discontinuous, wait or auto off mode.

NOTE. Reading reserved registers is always 0, and the writing operation is not valid.

Register_address = Base_address + Address_offset.

34.3.1 ADC Memory Map

ADC0 base address 40100000h
ADC1 base address 40104000h

Table 34.7: ADC Memory Map

Offset	Register	Width	Access	Reset Value
0h	Status register (STS)	32	RW	00000040h
4h	Interrupt Enable Register (INTE)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
8h	Control Register (CTRL)	32	RW	00000000h
ch	Configuration 0 Register (CFG0)	32	RW	07400010h
10h	Configuration 1 Register (CFG1)	32	RW	000002C0h
14h	Sampling Register (SMP)	32	RW	00000002h
2ch	Channel Configure 0 Register (CHSEL0)	8	RW	00h
2dh	Channel Configure 0 Register (CHSEL1)	8	RW	00h
2eh	Channel Configure 0 Register (CHSEL2)	8	RW	00h
2fh	Channel Configure 0 Register (CHSEL3)	8	RW	00h
30h	Channel Configure 0 Register (CHSEL4)	8	RW	00h
31h	Channel Configure 0 Register (CHSEL5)	8	RW	00h
32h	Channel Configure 0 Register (CHSEL6)	8	RW	00h
33h	Channel Configure 0 Register (CHSEL7)	8	RW	00h
4ch	FIFO Access Register (FIFO)	32	RW	see description
54h	Temperature Sensor Coefficient (TSC)	32	RW	00000000h
60h	Calibration Register (CAL)	32	RW	00000000h
80h	Watchdog Control Register (WDCTRL)	32	RW	00000000h
90h	Watchdog Threshold Register (WDTH0)	32	RW	0FFF0000h
94h	Watchdog Threshold Register (WDTH1)	32	RW	0FFF0000h
98h	Watchdog Threshold Register (WDTH2)	32	RW	0FFF0000h
9ch	Watchdog Threshold Register (WDTH3)	32	RW	0FFF0000h

34.3.1.1 ADC STS Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R													WD3	WD2	WD1	WD0
W													W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R										EMPTY	FULL	OVR	EOSEQ	EOC	EOSMP	ADRDY
W												W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Yuntu Microelectronics

Table 34.8: ADC STS Register Description

Field	Function
19 WD3	**Watchdog 3 Flag** This bit is set by hardware when the converted voltage crosses the values defined in the WD3TH register. It is cleared by software writing 1 to it. 0b - No watchdog event occurred (or the flag event was already acknowledged and cleared by software). 1b - Watchdog event occurred.
18 WD2	**Watchdog 2 Flag** This bit is set by hardware when the converted voltage crosses the values defined in the WD2TH register. It is cleared by software writing 1 to it. 0b - No watchdog event occurred (or the flag event was already acknowledged and cleared by software). 1b - Watchdog event occurred.
17 WD1	**Watchdog 1 Flag** This bit is set by hardware when the converted voltage crosses the values defined in the WD1TH register. It is cleared by software writing 1 to it. 0b - No watchdog event occurred (or the flag event was already acknowledged and cleared by software). 1b - Watchdog event occurred.
16 WD0	**Watchdog 0 Flag** This bit is set by hardware when the converted voltage crosses the values defined in the WD0TH register. It is cleared by software writing 1 to it. 0b - No watchdog event occurred (or the flag event was already acknowledged and cleared by software). 1b - Watchdog event occurred.
6 EMPTY	**FIFO Empty Status** This bit is set by hardware when the FIFO is empty. It is cleared by hardware when FIFO is not empty. 0b - FIFO is not empty. 1b - FIFO is empty.
5 FULL	**FIFO Full Status** This bit is set by hardware when the FIFO is full. It is cleared by hardware when FIFO is not full. 0b - FIFO is not full. 1b - FIFO is full.
4	**ADC Overrun**

Yuntu Microelectronics

Field	Function
OVR	This bit is set by hardware when an overrun occurs, meaning that a new conversion has completed while the FULL flag was already set. It is cleared by software writing 1 to it. 0b - No overrun occurred (or the flag event was already acknowledged and cleared by software). 1b - Overrun has occurred.
3 EOSEQ	**End of Sequence Flag** This bit is set by hardware at the end of the conversion of a sequence. It is cleared by software writing 1 to it. 0b - Conversion sequence not complete (or the flag event was already acknowledged and cleared by software). 1b - Conversion sequence complete.
2 EOC	**End of Conversion Flag** This bit is set by hardware at the end of each conversion of a channel when a new data result is available in the FIFO register. It is cleared by software writing 1 to it or by reading the FIFO register. 0b - Channel conversion not complete (or the flag event was already acknowledged and cleared by software). 1b - Channel conversion complete.
1 EOSMP	**End of Sampling Flag** This bit is set by hardware during the conversion at the end of the sampling phase. 0b - Not at the end of the sampling phase (or the flag event was already acknowledged and cleared by software). 1b - End of sampling phase reached.
0 ADRDY	**ADC Ready Flag** This bit is set by hardware after the ADC has been enabled (ADEN=1) and when the ADC reaches a state where it is ready to accept conversion requests. It is cleared by software writing 1 to it. 0b - ADC not yet ready to start conversion (or the flag event was already acknowledged and cleared by software). 1b - ADC is ready to start conversion.

Yuntu Microelectronics

34.3.1.2 ADC INTE Register

Offset: 4h

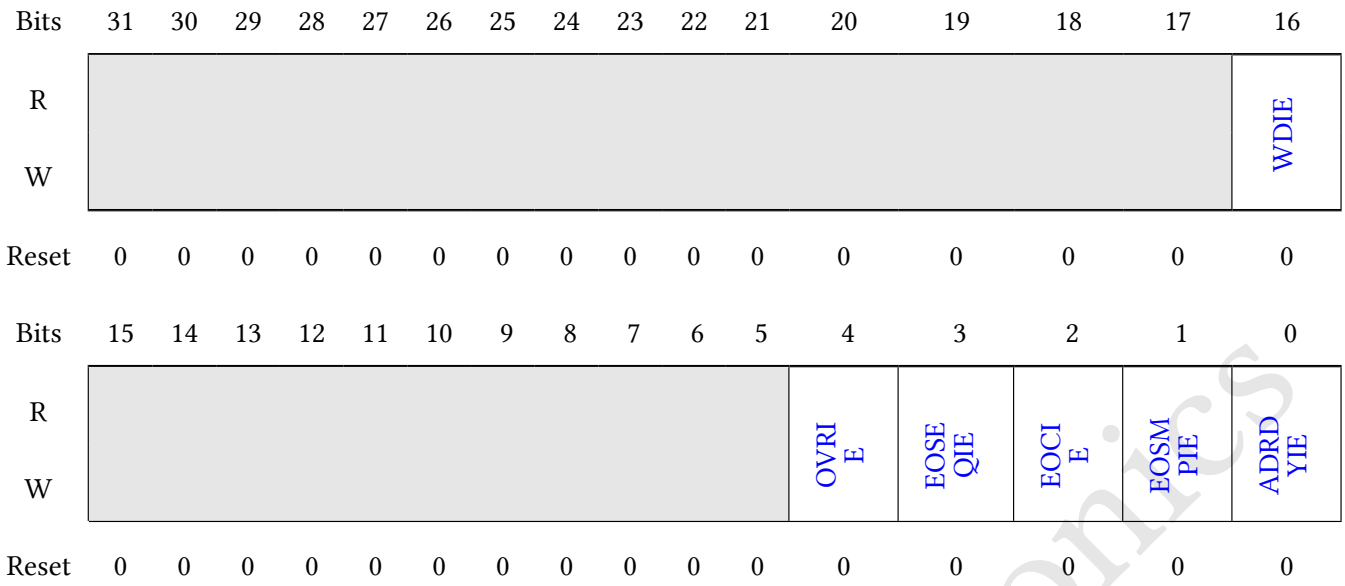


Table 34.9: ADC INTE Register Description

Field	Function
16 WDIE	**Watchdog Interrupt Enable** This bit is set and cleared by software to enable/disable the analog watchdog interrupt. 0b - Watchdog interrupt disabled. 1b - Watchdog interrupt enabled. An interrupt is generated when the WD bit is set.
4 OVRIE	**Overrun Interrupt Enable** This bit is set and cleared by software to enable/disable the overrun interrupt. 0b - Overrun interrupt disabled. 1b - Overrun interrupt enabled. An interrupt is generated when the OVR bit is set.
3 EOSEQIE	**End of Conversion Sequence Interrupt Enable** This bit is set and cleared by software to enable/disable the end of sequence of conversions interrupt. 0b - EOSEQ interrupt disabled. 1b - EOSEQ interrupt enabled. An interrupt is generated when the EOSEQ bit is set.
2 EOCIE	**End of Conversion Interrupt Enable** This bit is set and cleared by software to enable/disable the end of conversion interrupt. 0b - EOC interrupt disabled. 1b - EOC interrupt enabled. An interrupt is generated when the EOC bit is set.
1 EOSMPIE	**End of Sampling Flag Interrupt Enable** This bit is set and cleared by software to enable/disable the end of the sampling phase interrupt. 0b - EOSMP interrupt disabled. 1b - EOSMP interrupt enabled. An interrupt is generated when the EOSMP bit is set.

Yuntu Microelectronics

Field	Function
0 ADRDYIE	**ADC Ready Interrupt Enable** This bit is set and cleared by software to enable/disable the ADC Ready interrupt. 0b - ADRDY interrupt disabled. 1b - ADRDY interrupt enabled. An interrupt is generated when the ADRDY bit is set.

34.3.1.3 ADC CTRL Register

Offset: 8h

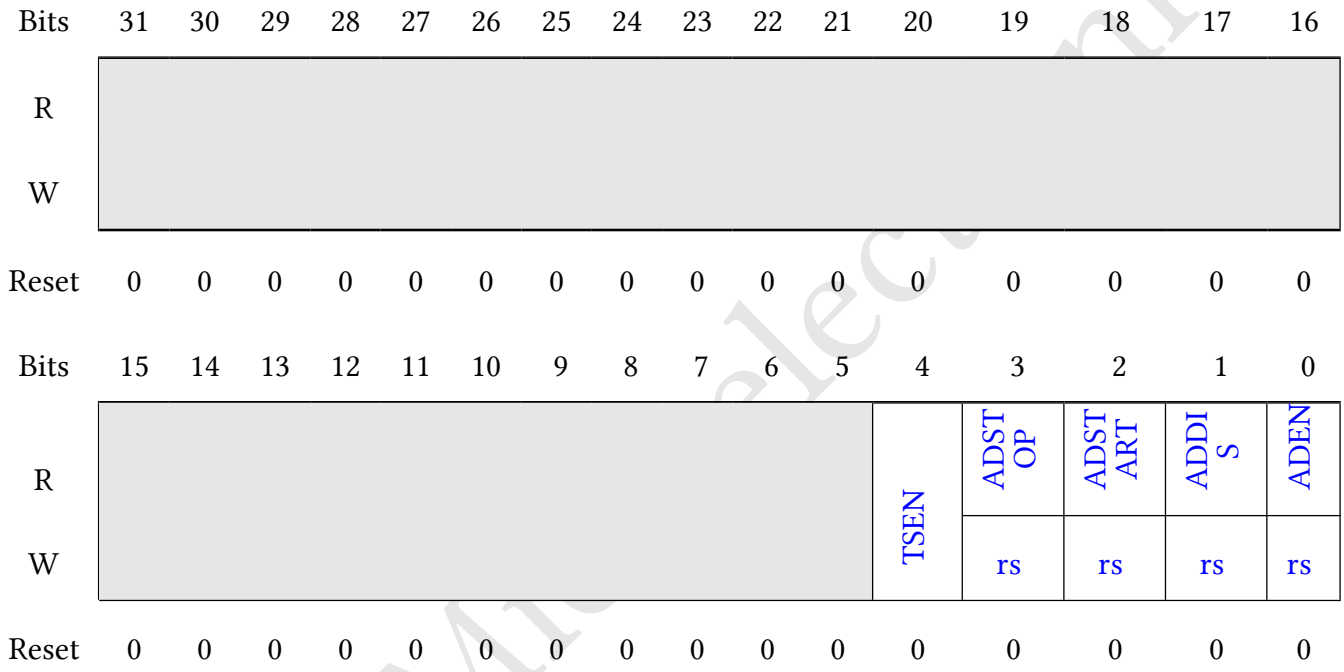


Table 34.10: ADC CTRL Register Description

Field	Function
4 TSEN	**Temperature Sensor Enable** This bit is to enable/disable internal temperature sensor. Only ADC0 support. 0b - Temperature sensor is disabled. 1b - Temperature sensor is enabled.
3 ADSTOP	**ADC Stop Conversion Command** This bit is set by software to stop and discard an ongoing conversion (ADSTOP Command). It is cleared by hardware when the conversion is effectively discarded and the ADC is ready to accept a new start conversion command. Software is allowed to set ADSTOP only when ADSTART=1 and ADDIS=0 (ADC is enabled and may be converting and there is no pending request to disable the ADC). 0b - No ADC stop conversion command ongoing. 1b - Write 1 to stop the ADC. Read 1 means that an ADSTOP command is in progress.

Yuntu Microelectronics

Field	Function
2 ADSTART	**ADC Start Conversion Command** This bit is set by software to start ADC conversion. Depending on the TRIGMD configuration bits, conversion either starts immediately (software trigger configuration) or once a hardware trigger event occurs (hardware trigger configuration). Software is allowed to set ADSTART only when ADEN=1 and ADDIS=0 (ADC is enabled and there is no pending request to disable the ADC). 0b - No ADC conversion is ongoing. 1b - Write 1 to start the ADC. Read 1 means that the ADC is operating and may be converting.
1 ADDIS	**ADC Disable Command** This bit is set by software to disable the ADC (ADDIS command) and put it into power-down state (OFF state). It is cleared by hardware once the ADC is effectively disabled (ADEN is also cleared by hardware at this time). 0b - No ADDIS command ongoing. 1b - Write 1 to disable the ADC. Read 1 means that an ADDIS command is in progress.
0 ADEN	**ADC Enable Command** This bit is set by software to enable the ADC. The ADC will be effectively ready to operate once the ADRDY flag has been set. It is cleared by hardware when the ADC is disabled, after the execution of the ADDIS command. 0b - ADC is disabled (OFF state). 1b - Write 1 to enable the ADC. Read 1 means that the ADC is enabled.

34.3.1.4 ADC CFG0 Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W						SEQLEN			OVRMD	LPEN	AUTOOFF	WAIT	ALIGN		MODE	
Reset	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CALMD	CALEN				NORTMD	RES		WM						DMAEN	
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Yuntu Microelectronics

Table 34.11: ADC CFG0 Register Description

Field	Function
26 - 24 SEQLEN	**Normal Sequence Length** These bits define the length of sequence. The sequence length is SEQLEN+1. For example, SEQLEN=7 means sequence length is 8; SEQLEN=0 means sequence length is 1.
23 OVRMD	**Overrun Management Mode** This bit is set and cleared by software and configure the way data overruns. 0b - FIFO register is preserved with the old data when an overrun is detected. 1b - The last data is overwritten with the conversion result when an overrun is detected.
22 LPEN	**Low power enable** Keep working enabled/disabled in low power mode. 0b - Keep working disabled in low power mode. 1b - Keep working enabled in low power mode.
21 AUTOOFF	**Auto-off Mode** This bit is set and cleared by software to enable/disable auto-off mode. 0b - Auto-off mode disabled. 1b - Auto-off mode enabled.
20 WAIT	**Wait Mode** This bit is set and cleared by software to enable/disable wait mode. 0b - Wait mode off. 1b - Wait mode on.
19 ALIGN	**Data Align** Data left/right aligned when read FIFO. 0b - Right aligned. 1b - Left aligned.
17 - 16 MODE	**Work Mode Selection** Work mode selection. 00b - Single mode. 01b - Continuous mode. 10b - Discontinuous mode. 11b - Reserved.
15 CALMD	**Calibration Mode** When select calibration mode, the conversion result will be calibrated automatically by CAL[OFFSET]. 0b - The conversion result will not be calibrated automatically. 1b - The conversion result will be calibrated automatically.
14	**Calibration Enable**

Yuntu Microelectronics

Field	Function
CALEN	Enable/disable calibration. The calibration values CAL[OFFSET] will be obtained automatically by 32 times conversion results. 0b - Calibration disabled. 1b - Calibration enabled.
10 NORTMD	**Normal Trigger Mode** The bit is used to select software trigger or external hardware trigger. 0b - Select software trigger as the trigger source. 1b - Select hardware trigger as the trigger source.
9 - 8 RES	**Data Resolution** These bits are written by software to select the resolution of the conversion. 00b - 12-bit resolution. 01b - 10-bit resolution. 10b - 8-bit resolution. 11b - 6-bit resolution.
7 - 4 WM	**Conversion Data Water Mark** When conversion data number is more than WM value and DMA enabled, trigger a DMA request.
0 DMAEN	**Direct Memory Access Enable** This bit is set and cleared by software to enable the generation of DMA requests. This allows to use the DMA controller to manage conversion data automatically. 0b - DMA disable. 1b - DMA enable.

Yuntu Microelectronics

34.3.1.5 ADC CFG1 Register

Offset: 10h

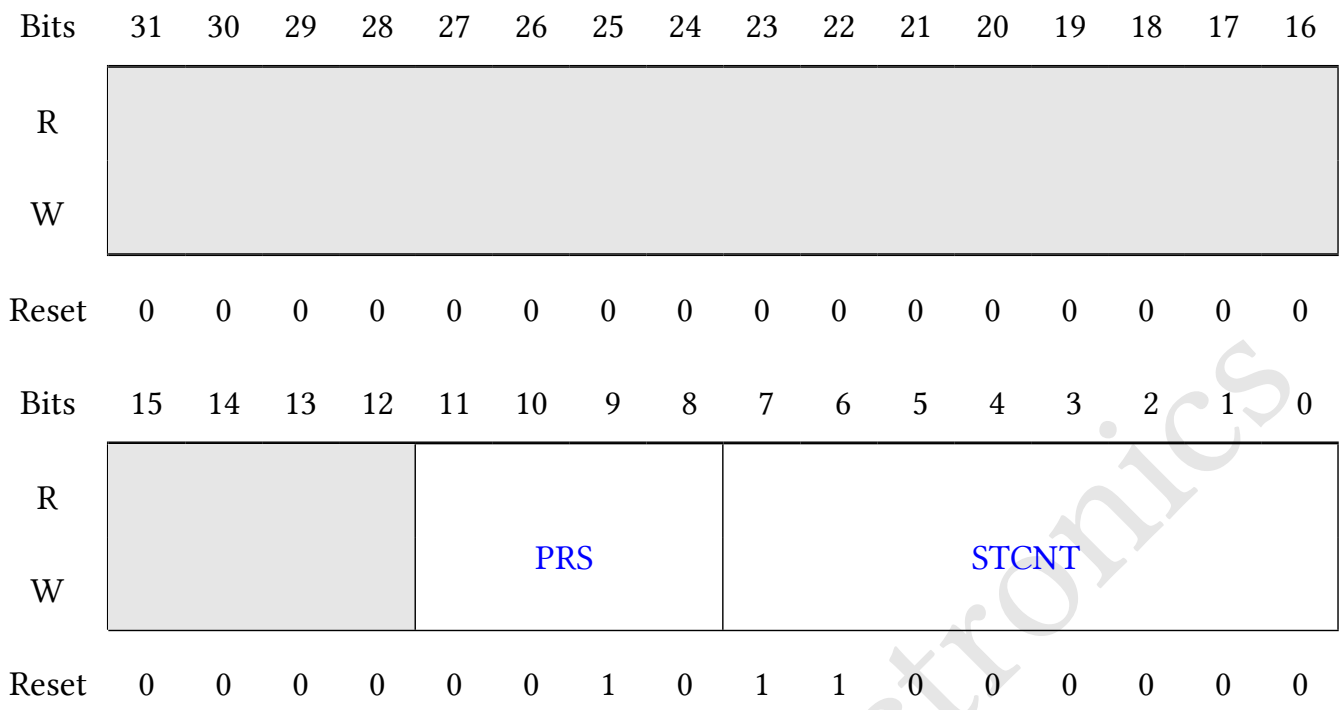


Table 34.12: ADC CFG1 Register Description

Field	Function
11 - 8 PRS	**Function Clock Prescaler** These bits are used to generate the function clock $F_{func_clk_div}$
7 - 0 STCNT	**ADC Startup Counter** The ADC needs a stabilization time of $t_{STAB}(2\mu s)$ before it starts converting accurately. This time is calculated by counting clock cycles(which is divided from function clock) until the internal counter reaches the STCNT. So user should set these bits before ADC enable. For example, if the $F_{func_clk_div} = 16MHz$, then should set the $STCNT_{min} = \frac{2\mu s}{\frac{1}{16MHz}} = 32$.

Yuntu Microelectronics

34.3.1.6 ADC SMP Register

Offset: 14h

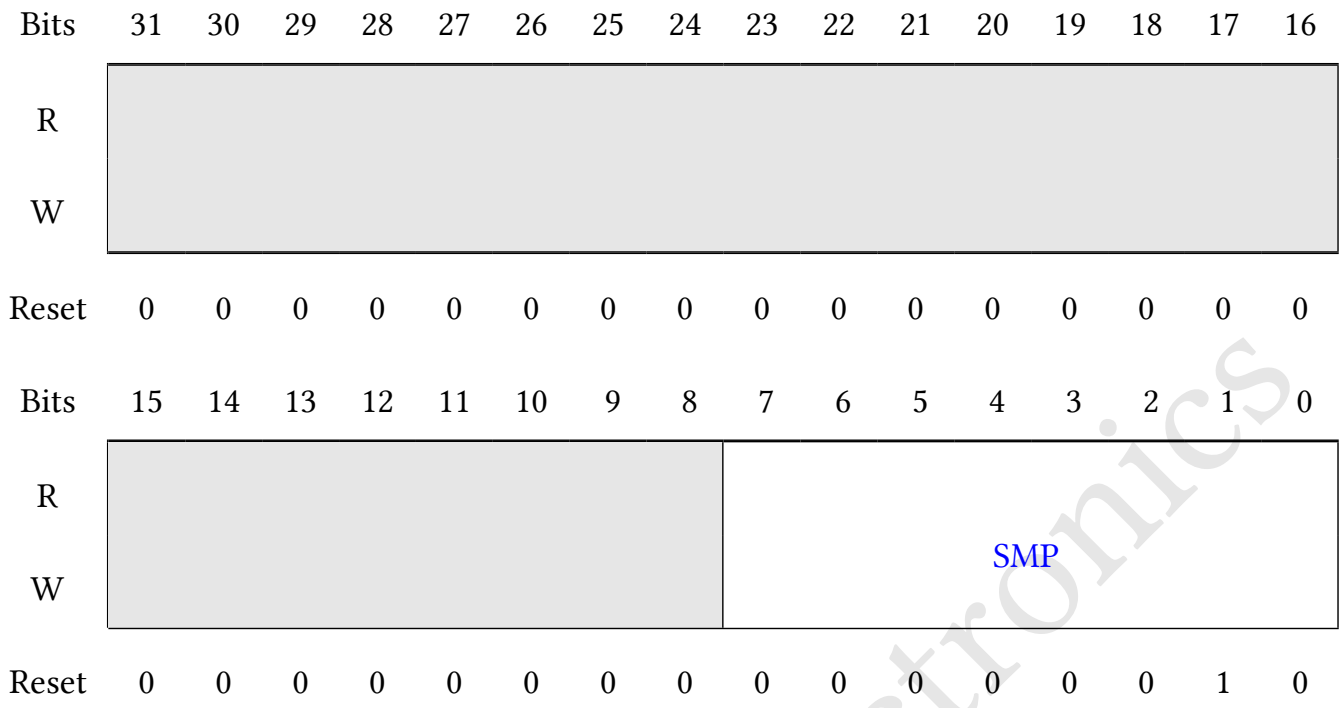


Table 34.13: ADC SMP Register Description

Field	Function
7 - 0 SMP	**Sampling Time Selection** These bits are written by software to select the sampling time that applies to all channels. The sample time is calculated as $T_{func_clk_div}$.

34.3.1.7 ADC CHSEL Register

Offset: 2ch + n * 1h

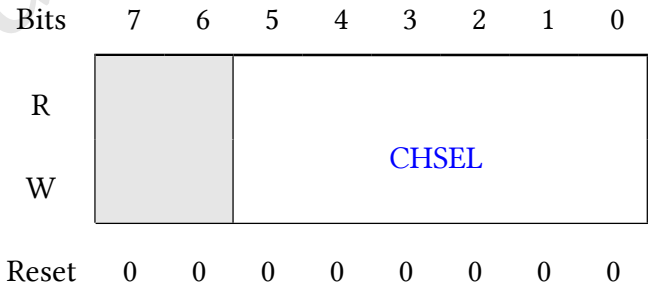


Table 34.14: ADC CHSEL Register Description

Field	Function
5 - 0 CHSEL	**Conversion Channel Select** These bits are configured by software to select the conversion channel.

Yuntu Microelectronics

34.3.1.8 ADC FIFO Register

Offset: 4ch

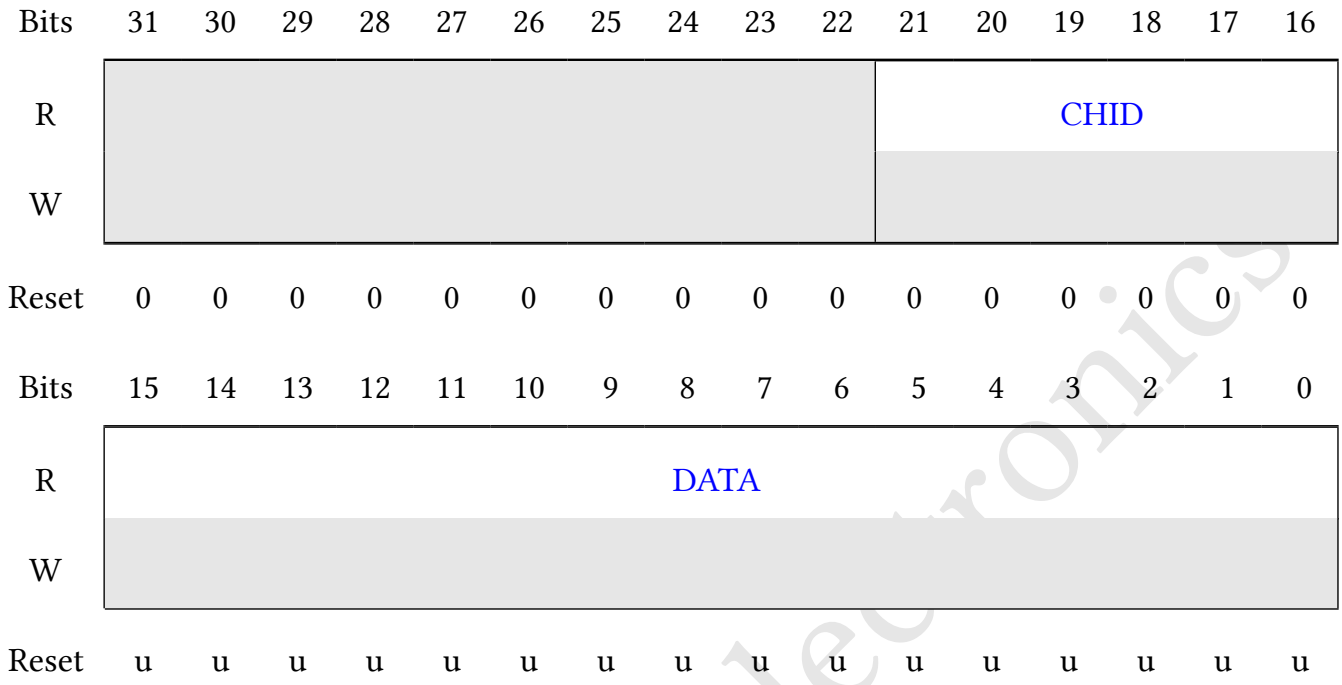


Table 34.15: ADC FIFO Register Description

Field	Function
21 - 16 CHID	**Channel ID** These bits indicate the channel identification that corresponds with the conversion data.
15 - 0 DATA	**Converted Data** Converted data

34.3.1.9 ADC TSC Register

Offset: 54h

Yuntu Microelectronics

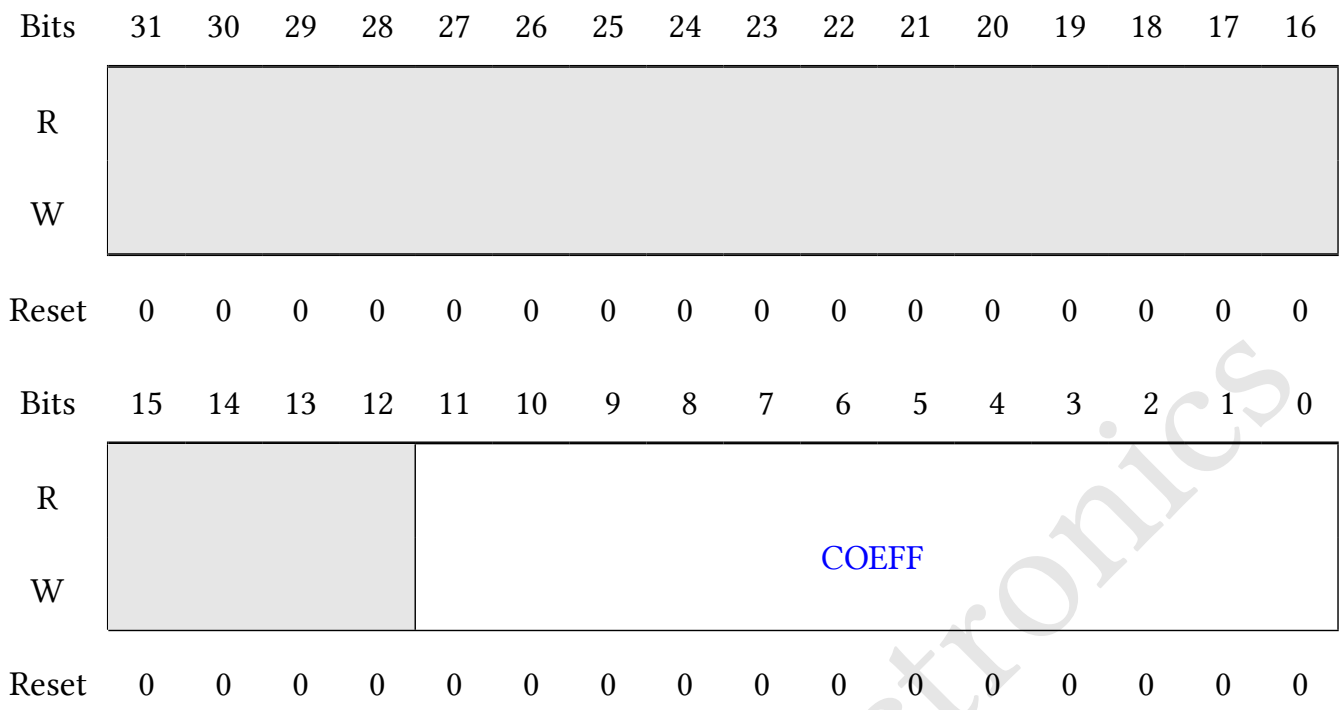


Table 34.16: ADC TSC Register Description

Field	Function
11 - 0 COEFF	**Temperature Sensor Coefficient** Temperature sensor coefficient value.

Yuntu Microelectronics

34.3.1.10 ADC CAL Register

Offset: 60h

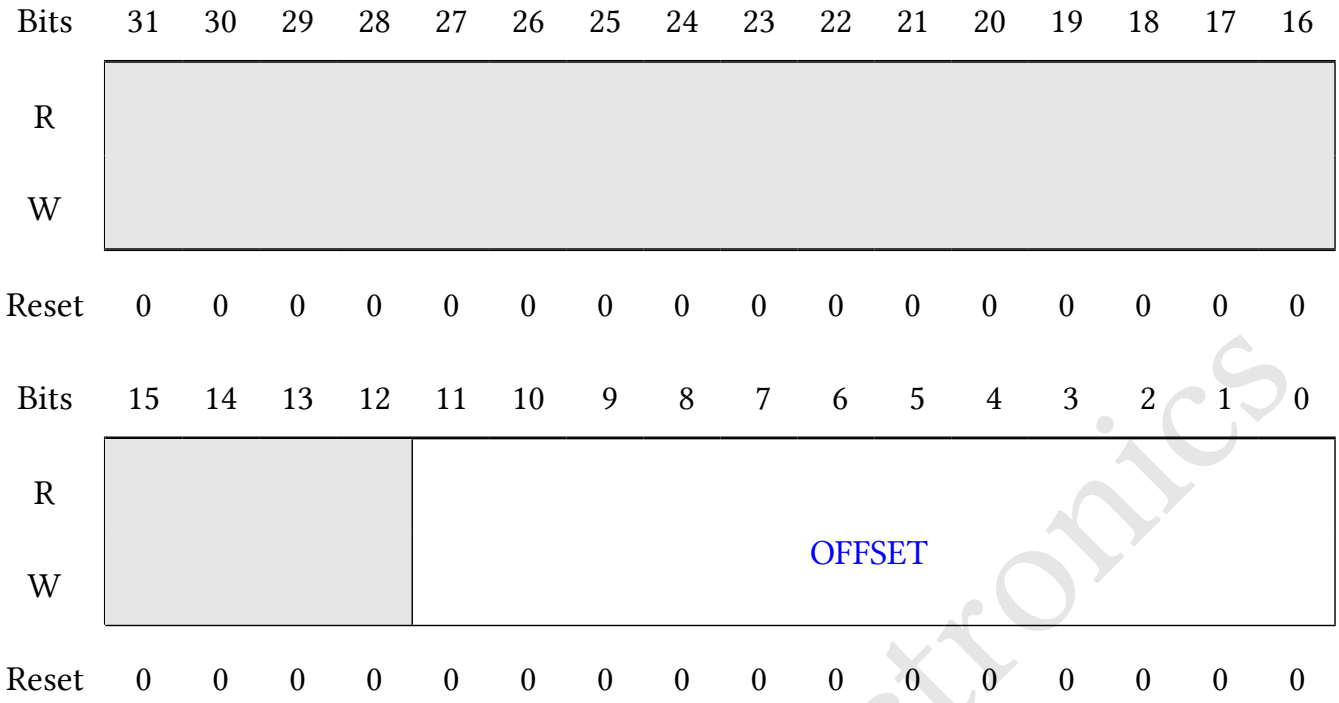
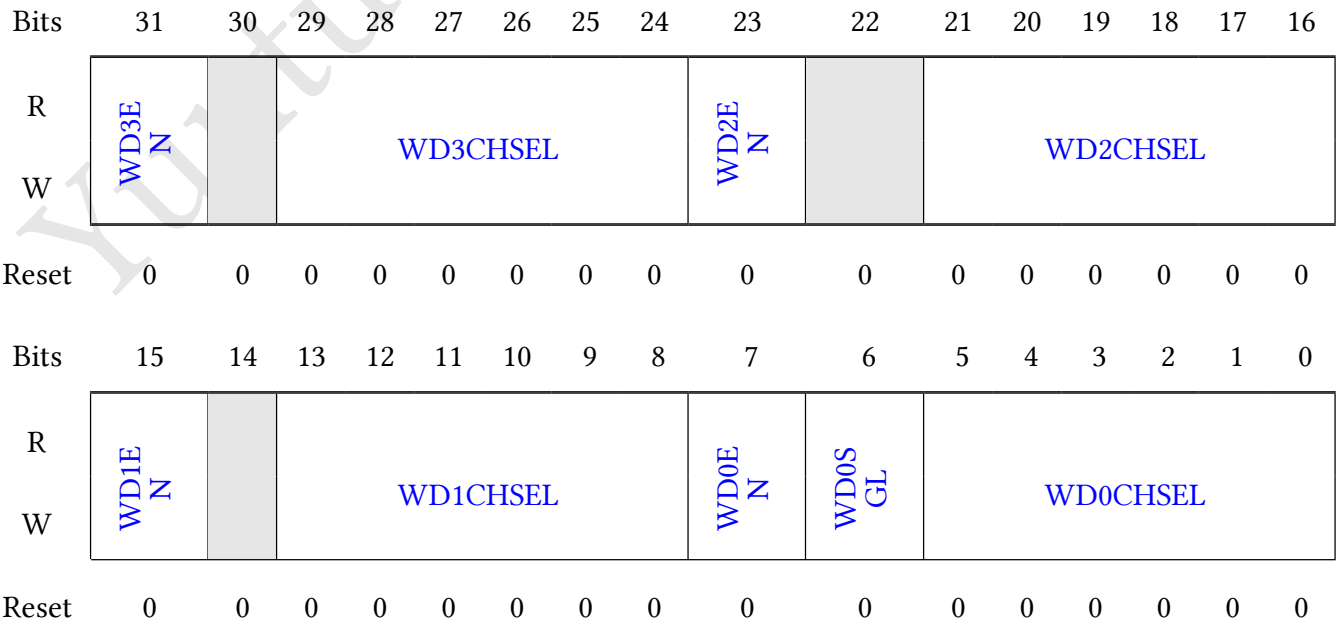


Table 34.17: ADC CAL Register Description

Field	Function
11 - 0 OFFSET	**Offset Value** Signed Offset value after calibration. Negative values are in complement format.

34.3.1.11 ADC WDCTRL Register

Offset: 80h



Yuntu Microelectronics

Table 34.18: ADC WDCTRL Register Description

Field	Function
31 WD3EN	**Watchdog 3 Enable** This bit is set and cleared by software to enable/disable watchdog. 0b - Watchdog 3 is disabled. 1b - Watchdog 3 is enabled.
29 - 24 WD3CHSEL	**Watchdog 3 Channel Select** These bits are set and cleared by software. They select the input channel to be guarded by the watchdog.
23 WD2EN	**Watchdog 2 Enable** This bit is set and cleared by software to enable/disable watchdog. 0b - Watchdog 2 is disabled. 1b - Watchdog 2 is enabled.
21 - 16 WD2CHSEL	**Watchdog 2 Channel Select** These bits are set and cleared by software. They select the input channel to be guarded by the watchdog.
15 WD1EN	**Watchdog 1 Enable** This bit is set and cleared by software to enable/disable watchdog. 0b - Watchdog 1 is disabled. 1b - Watchdog 1 is enabled.
13 - 8 WD1CHSEL	**Watchdog 1 Channel Select** These bits are set and cleared by software. They select the input channel to be guarded by the watchdog.
7 WD0EN	**Watchdog 0 Enable** This bit is set and cleared by software to enable/disable watchdog. 0b - Watchdog 0 is disabled. 1b - Watchdog 0 is enabled.
6 WD0SGL	**Apply the Watchdog 0 on a Single Channel** This bit is set and cleared by software to apply the watchdog on the channel identified by the WDCHSEL or all channels. 0b - Watchdog 0 enabled on all channels. 1b - Watchdog 0 enabled on a single channel.
5 - 0 WD0CHSEL	**Watchdog 0 Channel Select** These bits are set and cleared by software. They select the input channel to be guarded by the watchdog.

Yuntu Microelectronics

34.3.1.12 ADC WDTH Register

Offset: 90h + n * 4h

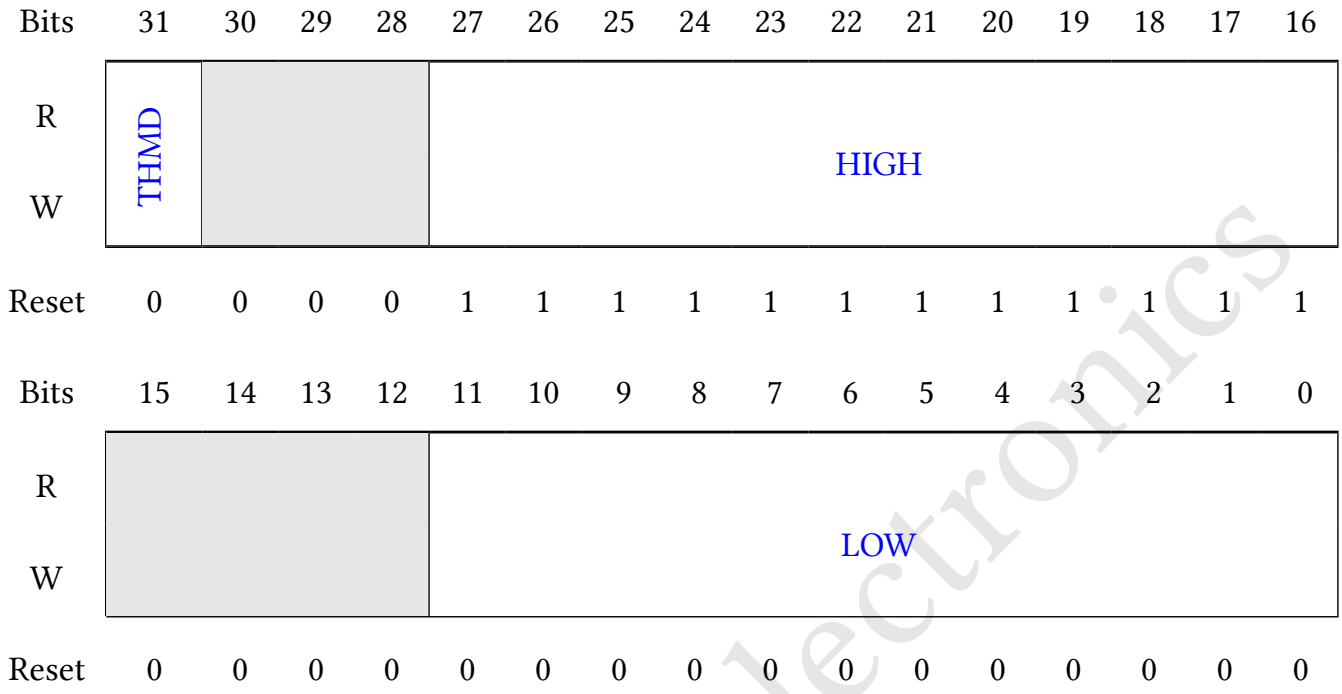


Table 34.19: ADC WDTH Register Description

Field	Function
31 THMD	**Watchdog Threshold Effective Mode** This bit is set and cleared by software to select the watchdog 0 threshold effective mode. 0b - Watchdog output validly when the converted voltage is outside the threshold defined by HIGH and LOW. 1b - Watchdog output validly when the converted voltage is inside the threshold defined by HIGH and LOW.
27 - 16 HIGH	**Watchdog Higher Threshold** These bits are written by software to define the higher threshold for the watchdog 0.
11 - 0 LOW	**Watchdog Lower Threshold** These bits are written by software to define the lower threshold for the watchdog 0.

34.4 Functional Description

ADC module Contains dual clock-domain architecture: bus clock and functional clock. It supports multiple conversion mode and asynchronous interrupt generation. The conversion results are stored in the FIFO. In addition, ADC module supports 4 analog window watchdog and up to 4 channels sequence

Yuntu Microelectronics

inject conversion.

34.4.1 Basic Configuration

34.4.1.1 ADC on-off Control

After MCU power-up, the ADC is disabled and under the 'Off' state (power-down mode). As shown in Figure ADC Convert Flow, the ADC needs a stabilization time $t_{startup}(\sim 2\mu s)$ before it starts converting accurately.

Two control bits CTRL[ADEN] and CTRL[ADDIS] are used to enable or disable the ADC:

- Set CTRL[ADEN]=1 to enable the ADC. ADC will enter the 'Start up' state from the 'Off' state, after a stabilization time $t_{startup}$ ADC will enter the 'Ready' state and wait for a conversion to start. The STS[ADRDY] flag is set as soon as the ADC is ready for operation.
- Set CTRL[ADDIS]=1 to disable the ADC. ADC will switch to the 'Off' state. The CTRL[ADEN] and CTRL[ADDIS] bits are then automatically cleared by hardware as soon as the ADC is fully disabled.

Two control bits CTRL[ADSTART] and CTRL[ADSTOP] are used to start or stop the conversion:

- When ADC is under the 'Ready' state, if software trigger is selected (CFG0[NORTMD]=0), set CTRL[ADSTART]=1 will start conversion immediately. If hardware trigger is selected (CFG0[NORTMD]=1), conversion doesn't start until getting an external trigger after setting CTRL[ADSTART]=1. The CTRL[ADSTART] bit will be cleared by hardware automatically after a sequence conversion finish or the ADC is fully disabled.
- When a conversion is ongoing, set CTRL[ADSTOP]=1 to stop the conversion, then ADC enters the 'Ready' state again. The CTRL[ADSTOP] bit is cleared by hardware automatically after conversion stops or the ADC is fully disabled.

Follow this procedure to enable the ADC:

- Set CTRL[ADEN]=1.
- Wait until STS[ADRDY]=1 (ADRDY is set after the ADC startup time $t_{startup}$). This can be handled by interrupt if the ADC ready interrupt is enabled by setting the INTE[ADRDYIE] bit.

Follow this procedure to disable the ADC:

- Check that CTRL[ADSTART]=0 to ensure that no conversion is ongoing. If required, stop any ongoing conversion by writing 1 to the CTRL[ADSTOP] bit and waiting until this bit reads 0.
- Set CTRL[ADDIS]=1.
- If required by the application, wait until CTRL[ADEN]=0, indicating that the ADC is fully disabled (CTRL[ADDIS] is automatically reset once ADEN=0).

Yuntu Microelectronics

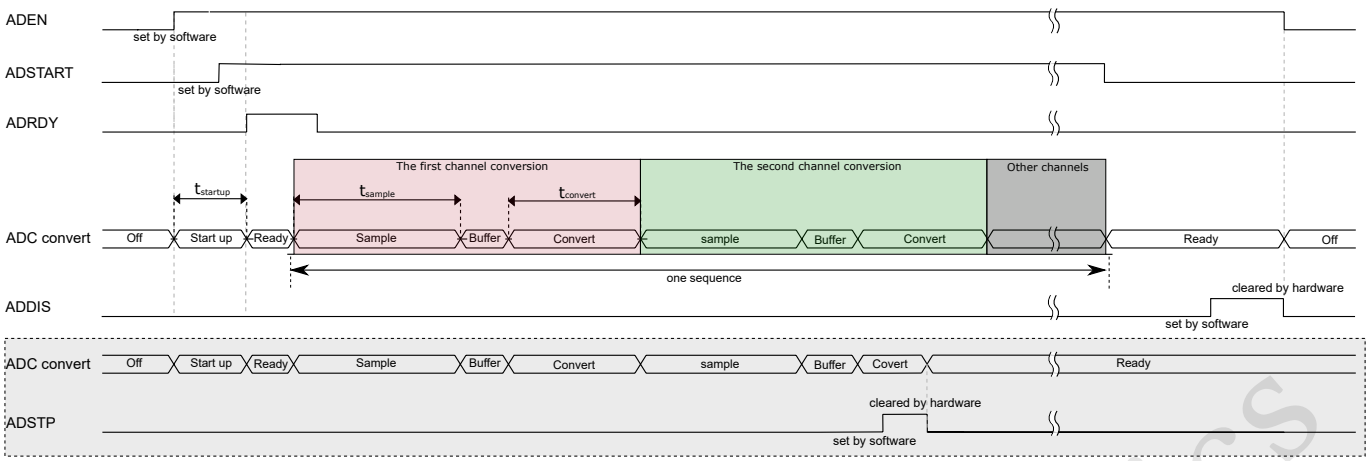


Figure 34.2: ADC Convert Flow

NOTE: In auto-off mode (CFG0[AUTOOFF]=1), the power-on/off phases are performed by hardware automatically, and the ADRDY flag won't be set.

34.4.1.2 ADC Clock

The ADC has a dual clock-domain architecture, as show in Figure ADC Clock Domain, it has the advantage that to reaching the maximum ADC clock frequency whatever the bus clock scheme selected.

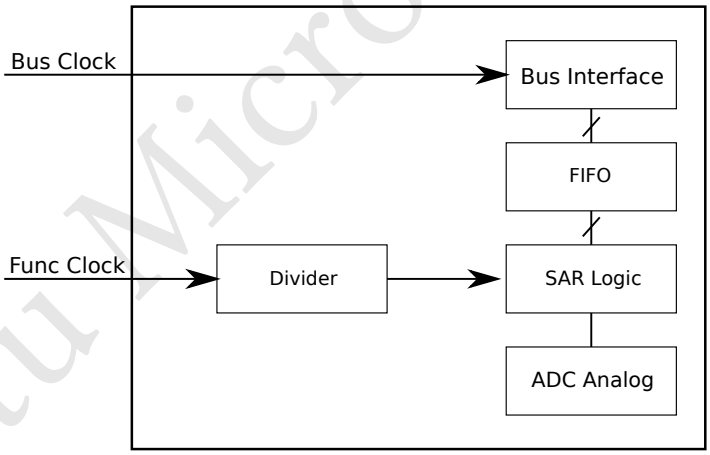


Figure 34.3: ADC Clock Domain

34.4.1.3 Channel Selection

There are up to 64 multiplexed channels from pins and 8 internal channels.

It is possible to convert a single channel or traverse several channels automatically.

If need scan several channels, they will be converted in sequence, organized as CHSEL0, then CHSEL1, ..., then CHSEL7. For example, if CFG0[SEQLEN] is configured as 2, the sequence is organized as CHSEL0, then CHSEL1, then CHSEL2.

Yuntu Microelectronics

34.4.1.4 ADC Configuration

Software must write to the CTRL[ADEN] bit if the ADC is disabled.

Software must only write to the CTRL[ADSTART] and CTRL[ADDIS] bits only if the ADC is enabled and there is no pending request to disable the ADC (CTRL[ADEN] = 1 and CTRL[ADDIS] = 0).

For all the other control bits in the INTE, CFG0/1, SMP, WDCTRL, WDTH, and CHSEL registers, software must only write to the configuration control bits if there is no conversion ongoing (CTRL[ADSTART] = 0).

Software must only write to the CTRL[ADSTOP] bit if the ADC is enabled (and possibly converting) and there is no pending request to disable the ADC (CTRL[ADSTART] = 1 and CTRL[ADDIS] = 0) .

NOTE: There is no hardware protection preventing software from making write operations forbidden by the above rules. If such a forbidden write access occurs, the ADC may enter an undefined state. To recover correct operation in this case, the ADC must be disabled(CTRL[ADDIS] = 1).

34.4.1.5 Starting conversions

Software starts ADC conversions by setting CTRL[ADSTART]=1.When CTRL[ADSTART] is set, the conversion:

- Starts immediately if CFG0[NORTMD] = 0 (software trigger)
- At the next active edge of the selected hardware trigger if CFG0[NORTMD] = 1

The CTRL[ADSTART] bit is also used to indicate whether an ADC operation is currently ongoing. It is possible to re-configure the ADC while CTRL[ADSTART]=0, indicating that the ADC is idle.

The CTRL[ADSTART] bit is cleared by hardware:

- In single mode with software trigger
  - At any end of conversion sequence (STS[EOSEQ]=1)
- In discontinuous mode with software trigger
  - At any end of conversion
- In all cases
  - After execution of the CTRL[ADSTOP] procedure invoked by software

NOTE:

- In continuous mode (CFG0[MODE]=0x1), the CTRL[ADSTART] bit is not cleared by hardware when the STS[EOSEQ] flag is set because the sequence is automatically relaunched.
- When hardware trigger is selected in single mode, CTRL[ADSTART] is not cleared by hardware when the STS[EOSEQ] flag is set. This avoids the need for software having to set the CTRL[ADSTART] bit again and ensures the next trigger event is not missed.

34.4.1.6 Stopping an ongoing conversion

The software can decide to stop any ongoing conversions by setting CTRL[ADSTOP]=1.

This will reset the ADC operation and the ADC will be idle, ready for a new operation.

When the CTRL[ADSTOP] bit is set by software, any ongoing conversion is aborted and the result is discarded (FIFO is not updated with the current conversion). The scan sequence is also aborted and reset (meaning that restarting the ADC would re-start a new sequence).

Yuntu Microelectronics

Once this procedure is complete, the CTRL[ADSTOP] and CTRL[ADSTART] bits are both cleared by hardware and the software must wait until CTRL[ADSTART]=0 before starting new conversions.

NOTE:

The flags in STS register are not cleared by STOP command, and the data in FIFO are not lost.

34.4.2 Timings

The ADC conversion logic uses function clock F_{func_clk} with a divider configured by CFG1[PRS] bits.

$$F_{func_clk_div} = \frac{F_{func_clk}}{CFG1[PRS] + 1}$$

The ADC needs a stabilization time $t_{startup}(\sim 2\mu s)$ before it starts sampling and converting accurately. The stabilization time configured by CFG1[STCNT] bits. As shown ADC Convert Time

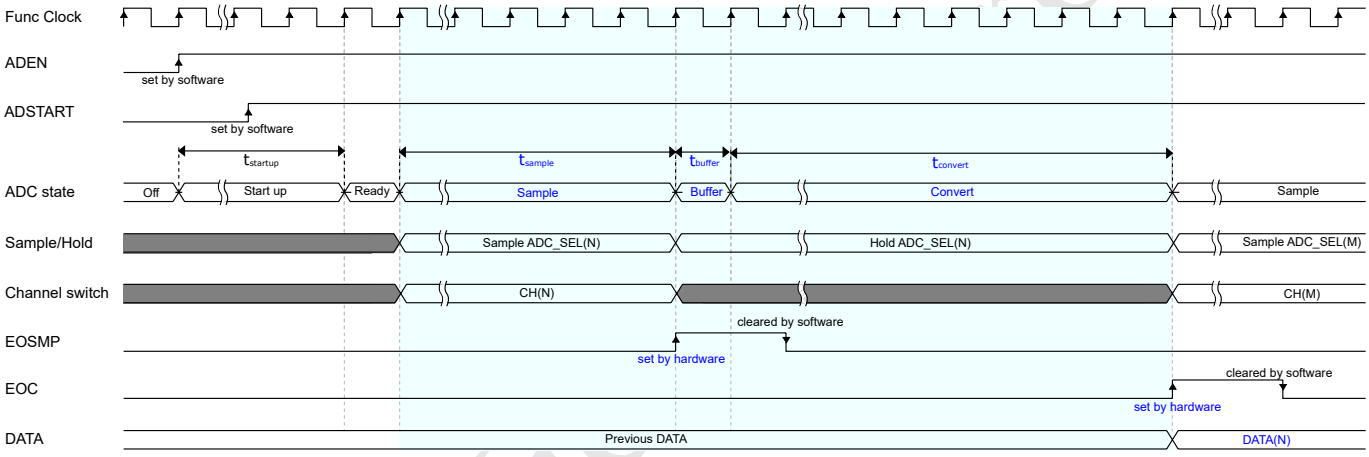


Figure 34.4: ADC Convert Time

$$t_{startup} = \frac{CFG1[STCNT]}{F_{func_clk_div}}$$

Before starting a conversion, the ADC needs to establish a direct connection between the voltage source to be measured and the embedded sampling capacitor of the ADC. This sampling time t_{sample} must be enough for the input voltage source to charge the sample and hold capacitor to the input voltage level. Having a programmable sampling time allows to trim the conversion speed according to the input resistance of the input voltage source. The ADC samples the input voltage for a number of ADC clock cycles that can be modified using the SMP[SMP] bits. This programmable sampling time is common to all channels. The ADC indicates the end of the sampling phase by setting the EOSMP flag. The 'Buffer' state take up a clock cycle and count as part of sample.

$$t_{sample} = \frac{SMP[SMP] + 2}{F_{func_clk_div}}$$

After sampling, the ADC starts the conversion and the conversion time varies depending on C_{res} (data resolution configured by CFG0[RES]). For 12-bit resolution, C_{res}=12.

$$t_{convert} = \frac{C_{res}}{F_{func_clk_div}}$$

The ADC conversion rate t_{adc}:

$$t_{adc} = t_{sample} + t_{convert}$$

For example, the ADC conversion rate is 1MHz, $F_{func_clk_div}$ =16MHz, SMP[SMP]=0x2, CFG0[RES]=0x0

$$t_{adc} = (\frac{(2+2)}{16} + \frac{12}{16})\mu s = 1\mu s$$

34.4.3 Functional Mode

34.4.3.1 Single conversion mode

In Single conversion mode, the ADC does a conversion of the selected sequence channel.

This mode is required to configure as follow:

- CFG0[MODE]=0

Conversion starts by either:

- Setting the CTRL[ADSTART] bit
- Hardware trigger event

Inside the sequence, after each conversion is complete:

- The converted data is stored into the FIFO
- The STS[EOC] (end of conversion) flag is set
- An interrupt is generated if the INTE[EOCIE] bit is set

After the sequence of conversions is complete:

- The STS[EOSEQ] (end of sequence) flag is set
- An interrupt is generated if the INTE[EOSEQIE] bit is set

Then the ADC stops until a new external trigger event occurs or the CTRL[ADSTART] bit is set again.

NOTE: To convert a single channel once, program the sequence length (CFG0[SEQLEN]) to 0.

34.4.3.2 Continuous conversion mode

In continuous conversion mode, when a software or hardware trigger event occurs, the ADC performs a sequence of conversions, after the sequence conversion finish, the ADC will re-start automatically and performs the same sequence of conversions continuously.

This mode is configured as follow:

- CFG0[MODE]=1.

Conversion is started by either:

- Setting the CTRL[ADSTART] bit

Yuntu Microelectronics

- Hardware trigger event

Inside the sequence, after each conversion is complete:

- The converted data is stored into the FIFO
- The STS[EOC] (end of conversion) flag is set
- An interrupt is generated if the INTE[EOCIE] bit is set

After the sequence of conversions is complete:

- The STS[EOSEQ] (end of sequence) flag is set
- An interrupt is generated if the INTE[EOSEQIE] bit is set

Then a new sequence restarts immediately and the ADC repeats the conversion continuously.

34.4.3.3 Discontinuous Mode

In discontinuous conversion mode, a hardware or software trigger event is required to start each conversion defined in the sequence.

This mode is configured as follow:

- CFG0[MODE]=2

Example:

- Channels to be converted = 0, 3, 7, 10
  - 1st trigger: channel 0 is converted and an STS[EOC] event is generated
  - 2nd trigger: channel 3 is converted and an STS[EOC] event is generated
  - 3rd trigger: channel 7 is converted and an STS[EOC] event is generated
  - 4th trigger: channel 10 is converted and both STS[EOC] and STS[EOSEQ] events are generated.
  - 5th trigger: channel 0 is converted an STS[EOC] event is generated
  - 6th trigger: channel 3 is converted and an STS[EOC] event is generated
  - ...

34.4.3.4 Conversion on external trigger

A conversion or a sequence of conversion can be triggered either by software or by an external event. If the CFG0[NORTMD] control bits are not equal to “0”, then external events are able to trigger a conversion. The trigger selection is effective once software set CTRL[ADSTART]=1.

Any hardware trigger which occurs while a conversion is ongoing will be ignored.

When CTRL[ADSTART]=0, any hardware trigger which occurs will be ignored.

The hardware trigger is from the TMU module or PTU module, and users can configure the CIM register to select the trigger type. There are three types of triggers to be chosen:

- Raw signal which from the PTU module (CIM->CTRL[ADCn_TRIG_SEL]=0)
- Signal from the TMU module and synchronized by bus clock (CIM->CTRL[ADCn_TRIG_SEL]=1)
- Signal from the TMU module and synchronized by function clock (CIM->CTRL[ADCn_TRIG_SEL]=2)

NOTE: Users should select the prepare trigger type according to the different applications.

Yuntu Microelectronics

34.4.3.5 Programmable resolution

It is possible to obtain shorter conversion time $t_{convert}$ by reducing the ADC resolution. The resolution can be configured to be either 12, 10, 8, or 6 bits by programming the CFG0[RES] bits. Lower resolution allows faster conversion times for applications where high data precision is not required.

NOTE:

- The CFG0[RES] bit must only be changed when the CTRL[ADEN] bit is 0.
- The result of the conversion is always 12 bits width and any unused LSB bits are read as zeroes.
- Lower resolution reduces the conversion time needed for the successive approximation steps.

34.4.3.6 End of sampling/conversion

The ADC indicates the end of sampling phase by setting the STS[EOSMP] flag. An interrupt can be generated if the INTE[EOSMPIE] bit is set. The STS[EOSMP] flag is cleared by software by writing 1 to it.

The ADC indicates each end of conversion by setting the STS[EOC] flag. The ADC sets the flag as soon as a new conversion data result is available. An interrupt can be generated if the INTE[EOCIE] bit is set. The STS[EOC] flag is cleared by software by either writing 1 to it or reading the FIFO.

34.4.3.7 End of conversion sequence

The ADC notifies the application of each end of sequence event by setting the STS[EOSEQ] flag.

The ADC sets the STS[EOSEQ] flag once the last data result of a conversion sequence is available in the FIFO. An interrupt can be generated if the INTE[EOSEQIE] bit is set. The STS[EOSEQ] flag is cleared by software by writing 1 to it.

34.4.4 Calibration

The ADC module supports calibration automatically when CFG0[CALEN]=1 and all channels are disabled. The ADC complete 32 conversions and take the averages. The calibration is 13-bits signed value. The conversion result in FIFO is automatically calibrated when CFG0[CALMD]=1. The calibration should be configured before normal conversion progress.

34.4.5 Data management

34.4.5.1 Data FIFO

At the end of each conversion (when an STS[EOC] event occurs), the result of the converted data is stored in the FIFO which is 32bit width x 16 depth, the FIFO data is consists of converted results (FIFO[DATA]) and conversion channel identifications (FIFO[CHID]).

The format of the read-out converted data depends on the configured data resolution and supports left/right aligned data by CFG0[ALIGN] configuration.

The FIFO supports byte, half-word, and word access, but the address offset should always be 0x4c.

For different data resolutions and use right aligned data configuration, see ADC DATA Resolution, users should note:

- If read by word, read once to get data[31:0], the next data is ready after reading data[31:0]
- If read by half word, read twice to get data[31:0], the next data is ready after reading data[15:0]
- If read by byte, read four times to get data[31:0], the next data is ready after reading data[7:0]

Yuntu Microelectronics

Table 34.20: ADC DATA Resolution

RES[1:0]	31	30	...	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0			...	chid[4:0]									data[11:0]											
1			...	chid[4:0]											data[9:0]									
2			...	chid[4:0]													data[7:0]							
3			...	chid[4:0]															data[5:0]					

34.4.5.2 ADC overrun

The overrun flag (STS[OVR]) indicates a data overrun event, when the converted data was not read in time by the CPU or the DMA before the FIFO is full.

The STS[OVR] flag is set if the STS[FULL] flag is still at '1' at the time when a new conversion completes. An interrupt can be generated if the INTE[OVRIE] bit is set.

When an overrun condition occurs, the ADC keeps operating and can continue to convert unless the software decides to stop and reset the sequence by setting the CTRL[ADSTOP] bit.

The STS[OVR] flag is cleared by software by writing 1 to it.

It is possible to configure if the data is preserved or overwritten when an overrun event occurs by programming the CFG0[OVRMD] bit:

- CFG0[OVRMD]=0

An overrun event preserves the data register from being overwritten: the old data is maintained and the new converted data is discarded. If STS[OVR] remains at 1, further conversions can be performed but the resulting data will be discarded. - CFG0[OVRMD]=1

The last data in the FIFO is overwritten with the conversion result. If STS[OVR] remains at 1, further conversions can be performed and the FIFO always contains the data from the latest conversion.

34.4.5.3 Managing a sequence of data converted without using the DMA

If the conversions are slow enough, the conversion sequence can be handled by software. In this case the software can use the interrupt that generated by STS[EOC] flag to handle each data result. Each time a conversion is complete, the STS[EOC] bit will be set and the FIFO register can be read.

Software can also use FIFO STS[EMPTY] flag to handle each data result. If STS[EMPTY] is not 0, it means that the FIFO has new data.

The CFG0[OVRMD] bit should be configured to 0 to manage overrun events as an error.

34.4.5.4 Managing converted data without using the DMA and overrun

It may be useful to let the ADC convert one or more channels without reading the data after each conversion. In this case, the CFG0[OVRMD] bit must be configured at 1 and the STS[OVR] flag should be ignored by the software. When CFG0[OVRMD]=1, an overrun event does not prevent the ADC from continuing to convert and the FIFO always contains the latest conversion data.

Yuntu Microelectronics

34.4.5.5 Managing converted data using the DMA

Once the data number in the FIFO is not empty and bit CFG0[DMAEN] is set, the ADC will send request to the DMA. This allows the transfer of the converted data from the FIFO to the destination location selected by the software.

Despite this, if an overrun event occurs (STS[OVR]=1) because the DMA could not serve the DMA transfer request in time, the ADC stops generating DMA requests and the data corresponding to the new conversion is not transferred by the DMA, which means that all the data transferred to the RAM can be considered as valid.

Depending on the configuration of CFG0[OVRMD] bit, the data is either preserved or overwritten.

The DMA transfer requests are blocked until the software clears the CFG0[OVR] bit.

34.4.6 Power Saving Modes

34.4.6.1 Wait Mode

Wait mode conversion can be used to simplify the software as well as optimizing the performance of applications clocked at low frequency where there might be a risk of ADC overrun occurring. When the CFG0[WAIT] bit is set to 1, a new conversion can start only if the FIFO is not full.

This is a way to automatically adapt the ADC speed with the system speed that reads the data.

NOTE: Any hardware trigger which occurs while a conversion is ongoing will be ignored.

34.4.6.2 Auto-off Mode

The ADC has an automatic power management feature which is called auto-off mode, and is enabled by setting CFG0[AUTOOFF]=1.

When CFG0[AUTOOFF]=1, the ADC is always powered off when not converting and automatically wakes-up when a conversion is started by a software or hardware trigger. A startup-time is automatically inserted between the trigger event which starts the sampling and the conversion time of the ADC. The ADC is then automatically disabled once the sequence of conversions is complete. The STS[ADRDY] flag won't set under auto-off mode.

The auto-off mode can cause a dramatic reduction in the power consumption of applications that need relatively few conversions or when conversion requests are timed far enough apart (for example with a low-frequency hardware trigger) to justify the extra power and extra time used for switching the ADC on and off.

The auto-off mode can be combined with the wait mode (CFG0[WAIT]=1) for applications clocked at low frequency. This combination can provide significant power savings if the ADC is automatically powered-off during the wait phase and restarted as soon as the FIFO is read by the application.

34.4.7 Analog Window Watchdog

The 4 analog window watchdog feature is enabled by setting the WDCTRL[WDxEN] bit. It is used to monitor that either one selected channel or all enabled channels remain within a configured voltage range (window) as shown Analog Watchdog Guarded Area.

Yuntu Microelectronics

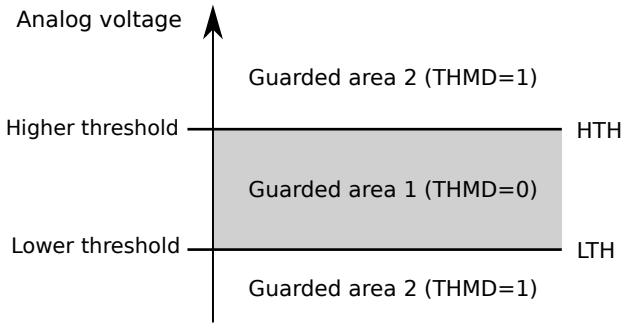


Figure 34.5: Analog Watchdog Guarded Area

It's able to select different threshold monitor modes by configuring WDTH[THMD]. When configure WDTH[THMD]=0, the analog watchdog status bit (STS[WDx]) is set if the analog voltage converted by the ADC is below a lower threshold or above a higher threshold. When configure WDxTH[THMD] = 1, the analog watchdog status bit (STS[WDx]) is set if the analog voltage converted by the ADC is above a lower threshold and below a higher threshold. These thresholds are programmed in the WDxTH register. An interrupt can be enabled by setting the WDIE bit in the INTE register.The WD flag is cleared by software by writing 1 to it.

When converting a data with a resolution that less than 12-bit (according to bits CFG0[RES]), the LSB of the programmed thresholds must be kept cleared because the internal comparison is always performed on the full 12-bit raw converted data.

Table Analog Watchdog Channel Selection shows how to configure the WDCTRL[WDSGL] and WDCTRL[WDEN] bits to enable the analog watchdog on one or more channels. When configuring WDCTRL[WDSGL]=1, the monitor channel number is selected by WDCTRL[WDCHSEL].

Table 34.21: Analog Watchdog Channel Selection

WDSGL bit	WDEN bit	Guarded Channels
x	0	None
0	1	All channels
1	1	Single channel

34.4.8 Temperature Sensor Description

The temperature sensor is internally connected to the ADC0 CHSEL = 32 internal channel which is used to convert the sensor's output voltage to a digital value. If select the internal temperature sensor channel, the ADC conversion rate should be equal to or less than 500Khz. See section Temperature Sensor (TEMPSENSOR) for specific temperature calculations.

34.4.9 Interrupts

An interrupt can be generated by any of the following events:

- ADC power-up, when the ADC is ready (STS[ADRDY] flag)
- End of any conversion (STS[EOC] flag)
- End of a sequence of conversions (STS[EOSEQ] flag)
- When a watchdog detection occurs (STS[WD] flag)

Yuntu Microelectronics

- When the end of sampling phase occurs (STS[EOSMP] flag)
- When a data overrun occurs (STS[OVR] flag)
- When the normal trigger missed error(STS[NORERR])
- When the inject trigger missed error(STS[INJERR])
- End of the inject sequence conversion(STS[INJFLAG])

Separate interrupt enable bits are available for flexibility.

Table 34.22: ADC Interrupts

Interrupt event	Event flag	Enable Control Bit
ADC ready	STS[ADRDY]	INTE[ADRDYIE]
End of conversion	STS[EOC]	INTE[EOCIE]
End of sequence of conversions	STS[EOSEQ]	INTE[EOSEQIE]
End of sampling phase	STS[EOSMP]	INTE[EOSMPIE]
Overrun	STS[OVR]	INTE[OVRIE]
Watchdog status bit is set	STS[WDx](x=0,1,2,3)	INTE[WDIE]

34.5 Glossary

ADC Analog-to-Digital Converter
FIFO First In First Out
SAR Successive Approximation Register
DMA Direct Memory Access

Yuntu Microelectronics

Chapter 35

Analog Comparator (ACMP)

35.1 Chip-specific ACMP information

35.1.1 Instantiation Information

YTM32B1HA0x contains one ACMP module. The input sources of the ACMP come from external analog signals or internal DAC output. ACMP module has two 8-to-1 analog MUXs to select external analog signals. The positive and negative port of the comparator share the common eight external analog input signals. ACMP module can compare two analog signals and raise an asynchronous interrupt, it can also operate on function clock which is independent from bus clock.

35.1.2 Signal Description

Table 35.1: ACMP Signal Table

Signal Name	Description	Direction
ACMPx_IN0	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN1	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN2	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN3	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN4	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN5	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN6	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_IN7	Input analog voltage, to the positive/negative port of the comparator	I
ACMPx_OUT	Output comparison result	O
ACMPx_ACTIVE	ACMPx current work state	O
VREFH	Reference voltage	I
DACx_OUT	DAC output to pin	O

Yuntu Microelectronics

35.1.3 Clock Description

Table 35.2: ACMP Clock Table

Clock Name	Description
Bus Clock	Bus clock used for registers access
Function Clock	Function clock used for function logic

35.1.4 DMA Description

Table 35.3: ACMP DMA Table

No.	Request Source	Description
48	Edge changes of ACMP out	Combine CTRL[EDGESEL] and STS[OUTNF/OUTPF] to generate DMA request

35.1.5 Interrupt Description

Table 35.4: ACMP Interrupt Table

Vector	IRQn	Request Source	Description	Wakeup
57	41	Result changes of ACMP in continuous mode or edge changes of ACMP out in other modes	Support asynchronous interrupt generation regardless of the existence of the bus clock	Y

35.1.6 Power Mode Description

Table 35.5: ACMP Power Mode Table

Power Mode	Description
Active	Full function
Sleep	Keeps working when function clock available
Deepsleep	Keeps working when function clock available
Standby	Keeps working when function clock available
Powerdown	Keeps working when function clock available. One of WKU wakeup sources.

35.2 Introduction

35.2.1 Overview

The Analog Comparator (ACMP) provides a circuit for comparing two analog input voltages and outputting comparison results. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation. The analog comparator module has two 8-to-1 analog MUXs. Each analog MUX provides a circuit for selecting an analog input signal from eight channels. The MUX circuit is designed to operate across the full range of the supply voltage.

Yuntu Microelectronics

35.2.2 Features

- Contain two ACMP instances
- Each ACMP supports 8 channels
- Operational over the entire supply range
- Inputs may range from rail to rail
- Programmable hysteresis control
- Selectable inversion on comparator output
- Function mode:
  - Common mode
  - Sample mode
  - Window mode
  - Continuous mode
    - one-shot mode
    - loop mode
- All channels can be used to execute automatic comparison
- Support digital filter, the filter can be bypassed
- Two software selectable performance levels
  - Shorter propagation delay at the expense of higher power
  - Low power with longer propagation delay
- Functional in all power modes
- Support independent 8-bit DAC output to the comparator
- Support DAC output to pin
- Support several interrupts
  - For common/sample/window mode
  - Generate interrupt on rising-edge, falling-edge or both edges of the comparator output
  - For continuous mode
  - Generate interrupt when the comparison results don't match with expectations
- Interrupt can be generated without any clock in common mode
- A comparison event can be selected to trigger DMA transfer

35.2.3 Block diagram

The following figure shows the ACMP block diagram.

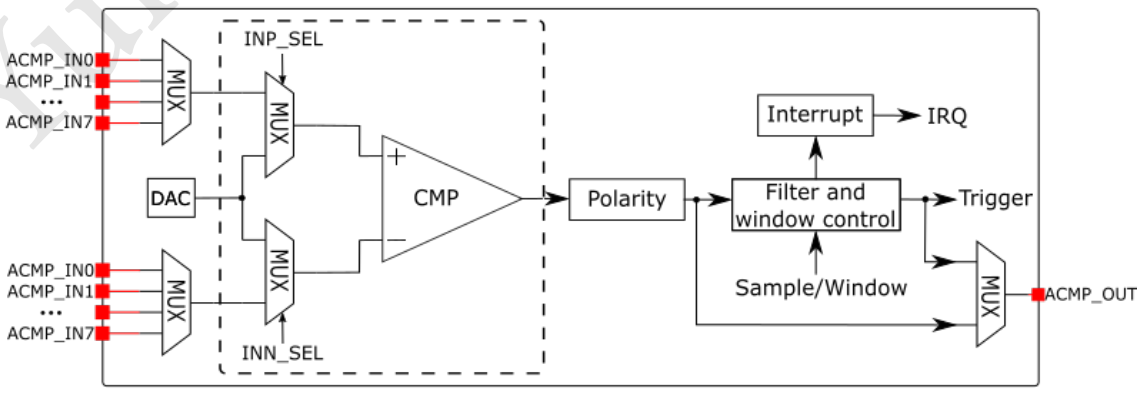


Figure 35.1: Block Diagram of ACMP

Yuntu Microelectronics

35.3 Register Definition

The ACMP registers are used to control ACMP module and display the current working status. In addition, registers are used to configure functional modes, such as continuous mode.

Reading reserved registers is always 0, the write operation is not valid.

Register_address = Base_address + Address_offset.

35.3.1 ACMP Memory Map

ACMP0 base address 4010c000h
ACMP1 base address 40110000h

Table 35.6: ACMP Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	DAC Register (DAC)	32	RW	00000000h
8h	Channels Multiplexer (MUX)	32	RW	00000000h
ch	Filter Register (FILT)	32	RW	00000000h
10h	DMA Control Register (DMACR)	32	RW	00000000h
18h	Status register (STS)	32	RW	00000000h
1ch	Interrupt Enable Register (INTE)	32	RW	00000000h
20h	Continuous Mode Register (CONT)	32	RW	18300000h
24h	Expect register (EXP)	32	RW	00000000h

35.3.1.1 ACMP CTRL Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W			EDGESEL								AUTODIS	OUTSEL	POL	OFFSET	HYSTVAL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W			INNSEL				INPSEL		CLKSEL		SMPMD				PWRMD	EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 35.7: ACMP CTRL Register Description

Field	Function
29 - 28 EDGESEL	**Edge Selection** Edge selection for interrupt/DMA 00b - No edge selection 01b - Select positive edge as interrupt/DMA condition, the positive edge flag is STS[OUTPF] 10b - Select negative edge as interrupt/DMA condition, the negative edge flag is STS[OUTNF] 11b - Select positive or negative edges as interrupt condition, not for DMA
21 AUTODIS	**Automatic Disable Hard Block** Automatic disable hard block when interrupt occurs or when waiting trigger in one-shot mode 0b - Not disable hard block automatically 1b - disable hard block automatically
20 OUTSEL	**Output Selection** Output selection bit 0b - Send ACMP filtered output to pin 1b - Send ACMP output to pin directly
19 POL	**Output Polarity** Output polarity selection bit 0b - Not invert comparator output 1b - Invert comparator output
18 OFFSET	**Comparator Offset Control Register** Comparator hard block offset control 0b - The hysteresis selected by HYSTVAL exists for INP crossing INN in the falling direction 1b - The hysteresis selected by HYSTVAL exists for both directions
17 - 16 HYSTVAL	**Comparator Hysteresis Value** Hysteresis level selection bit 00b - Comparator hard block output has no hysteresis internally 01b - Comparator hard block output has 20mV hysteresis internally 10b - Comparator hard block output has 40mV hysteresis internally 11b - Reserved
13 - 12 INNSEL	**Negative Input Selection** Negative input selection 00b - Select DAC output as comparator negative input 01b - Select one channel as comparator negative input 10b - Reserved 11b - Reserved
9 - 8	**Positive Input Selection**

Yuntu Microelectronics

Field	Function
INPSEL	Positive input selection 00b - Select DAC output as comparator positive input 01b - Select one channel as comparator positive input 10b - Reserved 11b - Reserved
7 - 6 CLKSEL	**Functional Clock Selection** Select the source of functional clock 00b - Functional clock is selected by IPC module (cannot be used in Powerdown mode) 01b - Functional clock is from SIRC 10b - Functional clock is from SXOSC 11b - Reserved
5 - 4 SMPMD	**Sample Mode Selection** Sample mode selection 00b - Common mode 01b - Trigger mode 10b - Window mode 11b - Reserved
1 PWRMD	**Power Mode** Power mode 0b - Low power mode 1b - High speed mode
0 EN	**Module Enable** Module enable 0b - ACMP module is disabled 1b - ACMP module is enabled

Yuntu Microelectronics

35.3.1.2 ACMP DAC Register

Offset: 4h

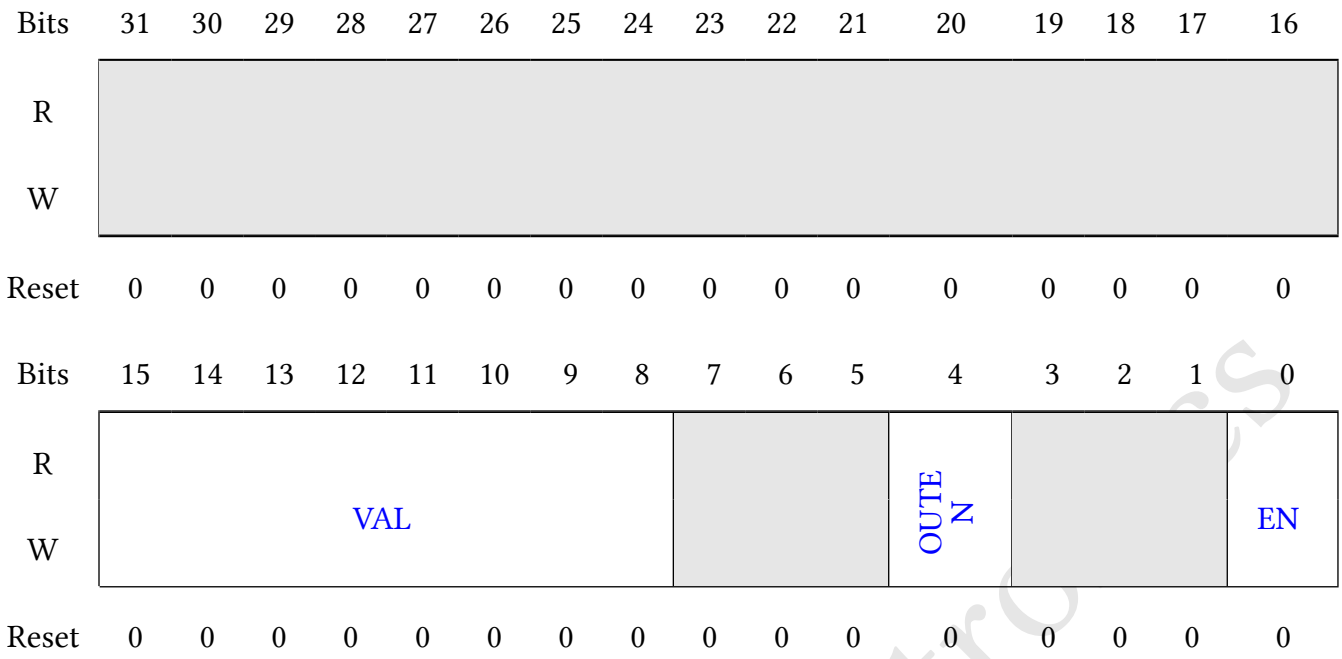


Table 35.8: ACMP DAC Register Description

Field	Function
15 - 8 VAL	**DAC Output Voltage Level** DAC output voltage level selection from value 0 to 255
4 OUTEN	**DAC Output Enable** DAC output to pin enable/disable 0b - DAC output to pin disabled 1b - DAC output to pin enabled
0 EN	**DAC Enable** DAC enable 0b - DAC is disabled 1b - DAC is enabled

35.3.1.3 ACMP MUX Register

Offset: 8h

Yuntu Microelectronics

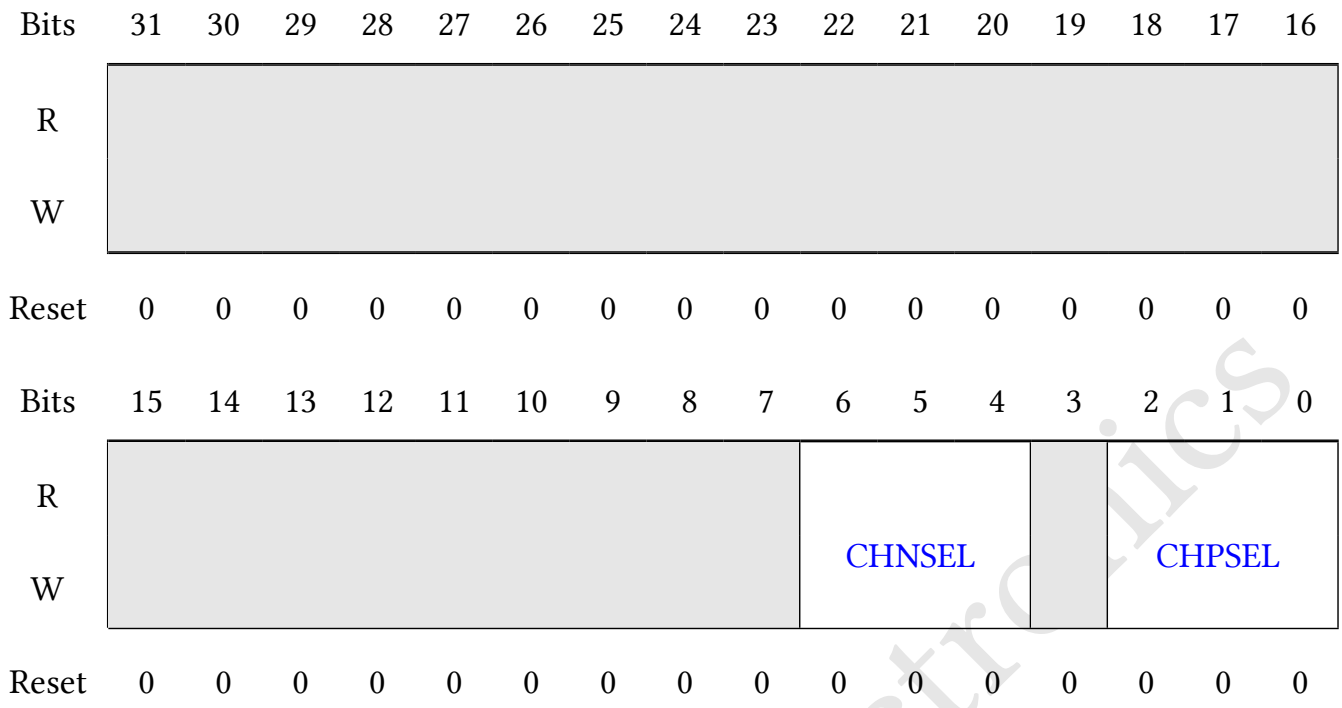


Table 35.9: ACMP MUX Register Description

Field	Function
6 - 4 CHNSEL	**Negative Channel Selection** Select one channel as negative Input of comparator 000b - Select channel 0 001b - Select channel 1 010b - Select channel 2 011b - Select channel 3 100b - Select channel 4 101b - Select channel 5 110b - Select channel 6 111b - Select channel 7
2 - 0 CHPSEL	**Positive Channel Selection** Select one channel as positive Input of comparator 000b - Select channel 0 001b - Select channel 1 010b - Select channel 2 011b - Select channel 3 100b - Select channel 4 101b - Select channel 5 110b - Select channel 6 111b - Select channel 7

35.3.1.4 ACMP FILT Register

Offset: ch

Yuntu Microelectronics

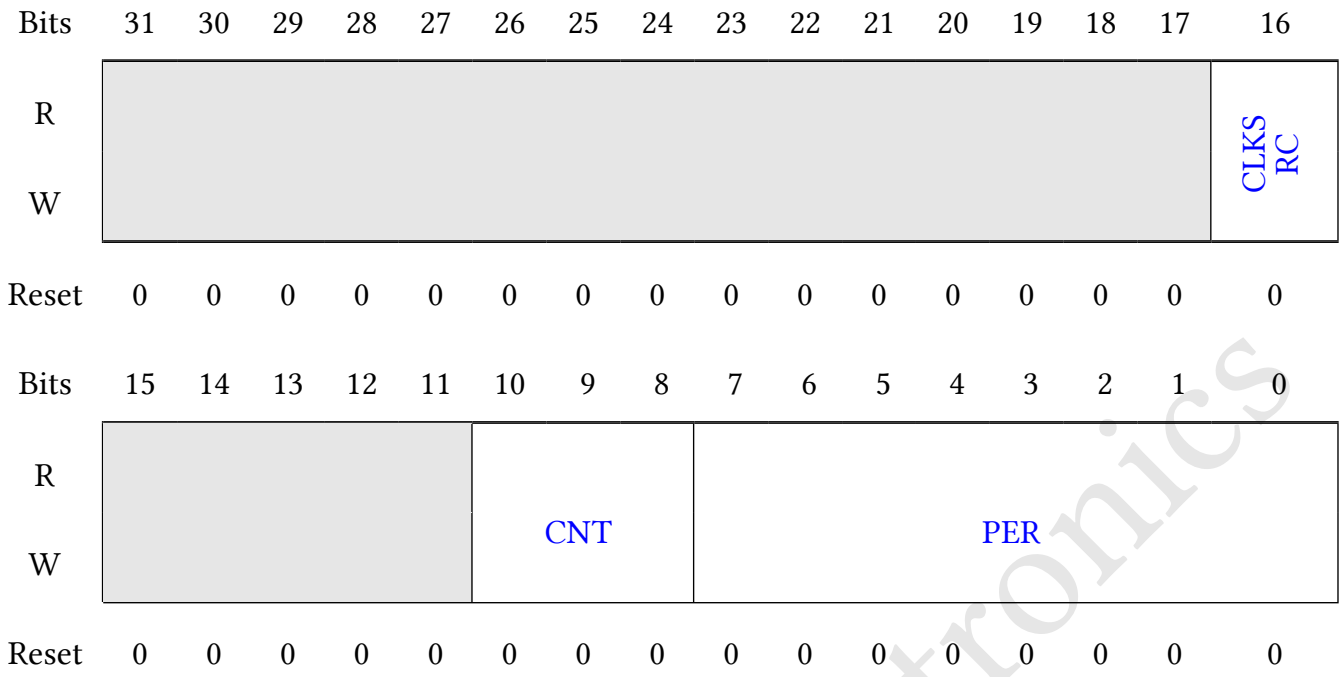


Table 35.10: ACMP FILT Register Description

Field	Function
16 CLKSRC	**Filter Clock Source** Filter clock source 0b - Select functional clock for filter 1b - Select bus clock for filter
10 - 8 CNT	**Digital Filter Counter** Digital filter counter
7 - 0 PER	**Digital Filter Period** Digital filter period

Yuntu Microelectronics

35.3.1.5 ACMP DMACR Register

Offset: 10h

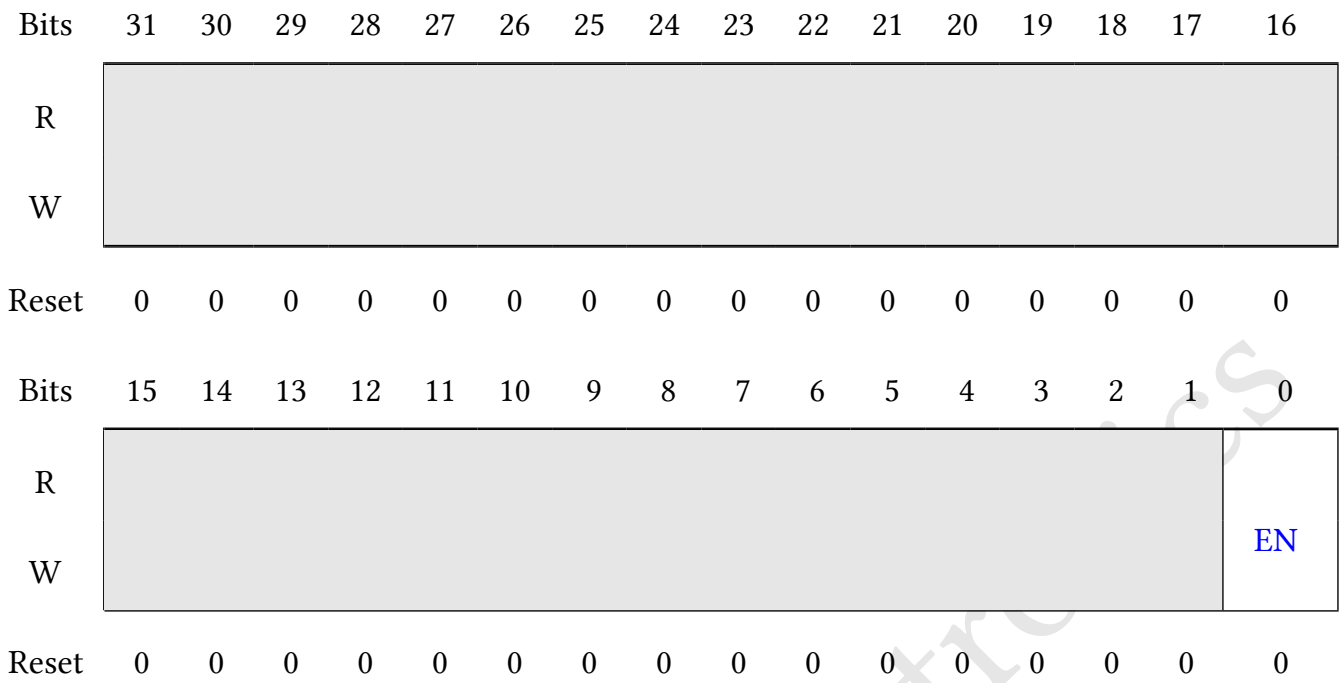
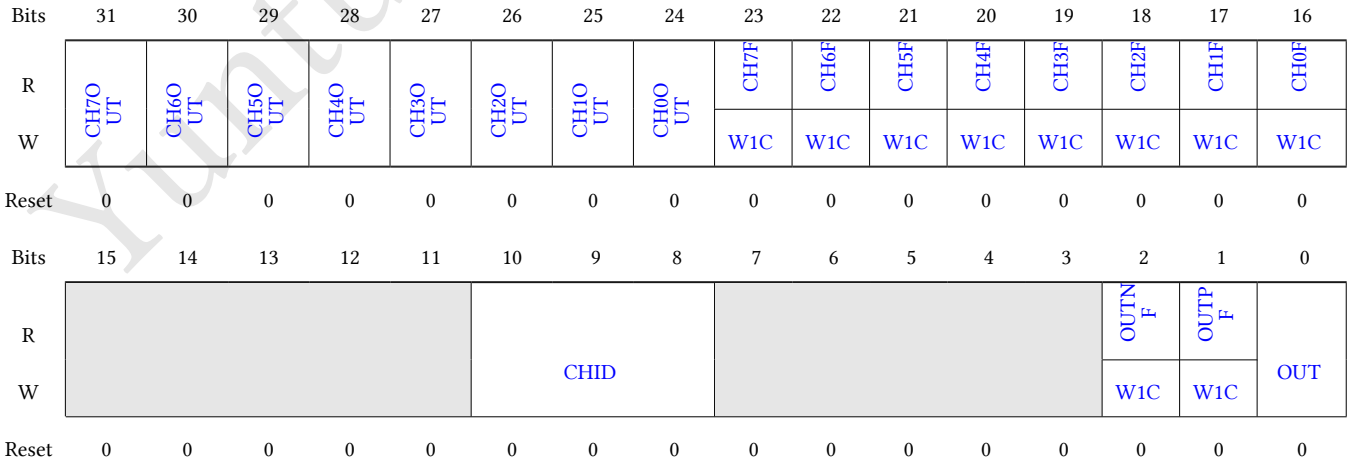


Table 35.11: ACMP DMACR Register Description

Field	Function
0 EN	**DMA Enable** DMA enable 0b - DMA request is disabled 1b - DMA request is enabled

35.3.1.6 ACMP STS Register

Offset: 18h



Yuntu Microelectronics

Table 35.12: ACMP STS Register Description

Field	Function
31 CH7OUT	**Channel 7 Output In Continuous Mode** Indicate the channel 7 result of comparator after continuous mode enabled 0b - Indicate the channel 7 result of comparator is 0 1b - Indicate the channel 7 result of comparator is 1
30 CH6OUT	**Channel 6 Output In Continuous Mode** Indicate the channel 6 result of comparator after continuous mode enabled 0b - Indicate the channel 6 result of comparator is 0 1b - Indicate the channel 6 result of comparator is 1
29 CH5OUT	**Channel 5 Output In Continuous Mode** Indicate the channel 5 result of comparator after continuous mode enabled 0b - Indicate the channel 5 result of comparator is 0 1b - Indicate the channel 5 result of comparator is 1
28 CH4OUT	**Channel 4 Output In Continuous Mode** Indicate the channel 4 result of comparator after continuous mode enabled 0b - Indicate the channel 4 result of comparator is 0 1b - Indicate the channel 4 result of comparator is 1
27 CH3OUT	**Channel 3 Output In Continuous Mode** Indicate the channel 3 result of comparator after continuous mode enabled 0b - Indicate the channel 3 result of comparator is 0 1b - Indicate the channel 3 result of comparator is 1
26 CH2OUT	**Channel 2 Output In Continuous Mode** Indicate the channel 2 result of comparator after continuous mode enabled 0b - Indicate the channel 2 result of comparator is 0 1b - Indicate the channel 2 result of comparator is 1
25 CH1OUT	**Channel 1 Output In Continuous Mode** Indicate the channel 1 result of comparator after continuous mode enabled 0b - Indicate the channel 1 result of comparator is 0 1b - Indicate the channel 1 result of comparator is 1
24 CH0OUT	**Channel 0 Output In Continuous Mode** Indicate the channel 0 result of comparator after continuous mode enabled 0b - Indicate the channel 0 result of comparator is 0 1b - Indicate the channel 0 result of comparator is 1
23	**Channel 7 Output Flag**

Yuntu Microelectronics

Field	Function
CH7F	In continuous mode, the flag set when channel 7 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 7 result of comparator match with its expected value 1b - Channel 7 result of comparator not match with its expected value
22 CH6F	**Channel 6 Output Flag** In continuous mode, the flag set when channel 6 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 6 result of comparator match with its expected value 1b - Channel 6 result of comparator not match with its expected value
21 CH5F	**Channel 5 Output Flag** In continuous mode, the flag set when channel 5 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 5 result of comparator match with its expected value 1b - Channel 5 result of comparator not match with its expected value
20 CH4F	**Channel 4 Output Flag** In continuous mode, the flag set when channel 4 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 4 result of comparator match with its expected value 1b - Channel 4 result of comparator not match with its expected value
19 CH3F	**Channel 3 Output Flag** In continuous mode, the flag set when channel 3 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 3 result of comparator match with its expected value 1b - Channel 3 result of comparator not match with its expected value
18 CH2F	**Channel 2 Output Flag** In continuous mode, the flag set when channel 2 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 2 result of comparator match with its expected value 1b - Channel 2 result of comparator not match with its expected value
17 CH1F	**Channel 1 Output Flag** In continuous mode, the flag set when channel 1 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 1 result of comparator match with its expected value 1b - Channel 1 result of comparator not match with its expected value
16 CH0F	**Channel 0 Output Flag** In continuous mode, the flag set when channel 0 result of comparator not match with its expected value. Write 1 to clear the flag 0b - Channel 0 result of comparator match with its expected value 1b - Channel 0 result of comparator not match with its expected value
10 - 8	**Current Channel identity**

Yuntu Microelectronics

Field	Function
CHID	Indicate current channel in continuous mode
2 OUTNF	**Comparator Output Negative Edge Flag** Comparator output negative edge flag
1 OUTPF	**Comparator Output Positive Edge Flag** Comparator output positive edge flag
0 OUT	**Current Comparator Output** Current comparator output

35.3.1.7 ACMP INTE Register

Offset: 1ch

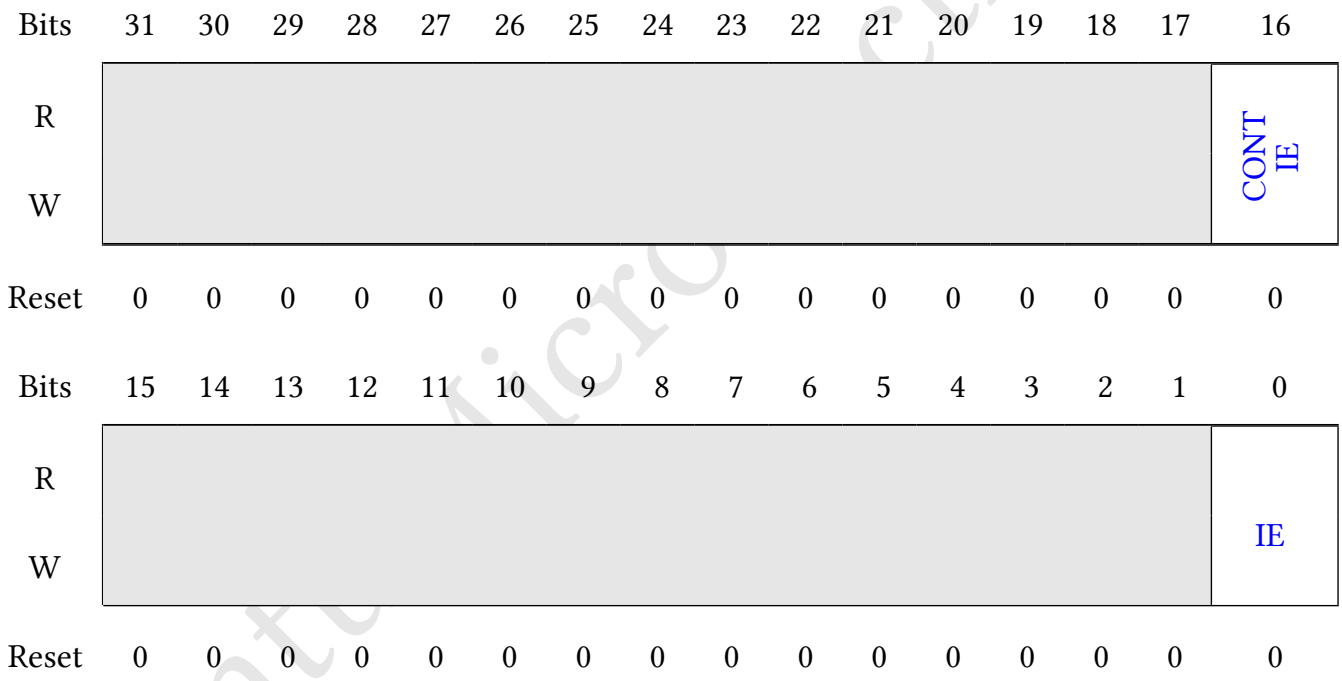


Table 35.13: ACMP INTE Register Description

Field	Function
16 CONTIE	**Continuous Mode Interrupt Enable** In continuous mode, interrupt occurs when INTE[CONTIE]=1 and STS[CHxF]=1 0b - Continuous mode interrupt is disabled 1b - Continuous mode interrupt is enabled
0	**Interrupt Enable**

Yuntu Microelectronics

Field	Function
IE	Interrupt occurs when INTE[IE]=1 and STS[OUTNF]=1/STS[OUTPF]=1(depending on CTRL[EDGESEL]) 0b - Interrupt is disabled 1b - Interrupt is enabled

35.3.1.8 ACMP CONT Register

Offset: 20h

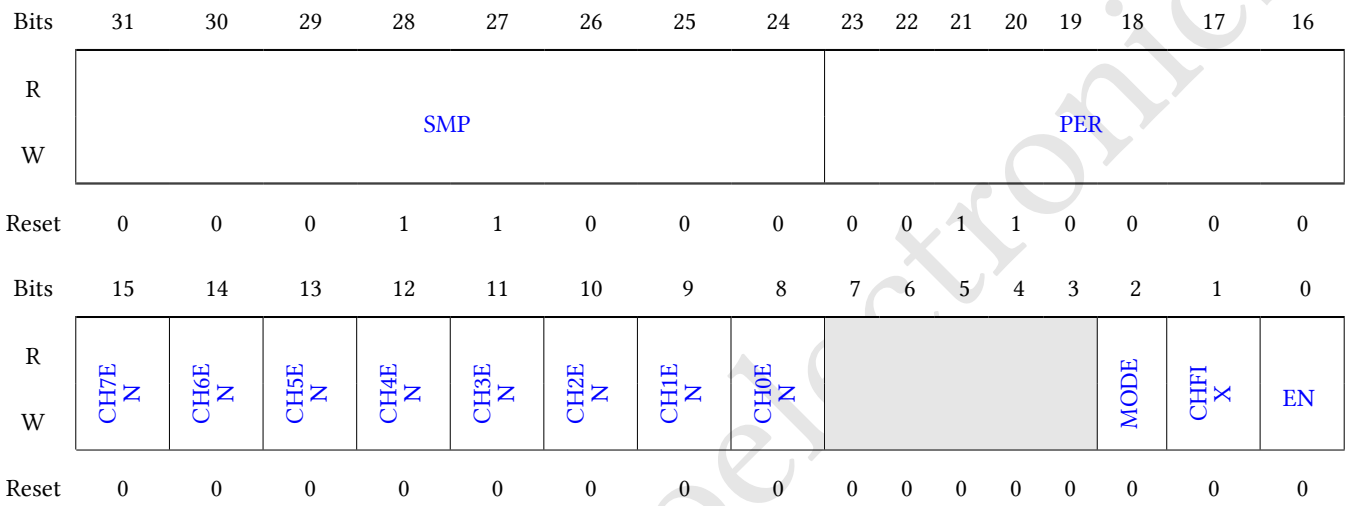


Table 35.14: ACMP CONT Register Description

Field	Function
31 - 24 SMP	**Sample Point in Continuous Mode** After channel switch stabilization time, sample the comparator result
23 - 16 PER	**Continuous Mode Period** Channel switch period
15 CH7EN	**Channel 7 Enable in Continuous Mode** Channel 7 participates in the continuous mode 0b - Channel 7 is disabled 1b - Channel 7 is enabled
14 CH6EN	**Channel 6 Enable in Continuous Mode** Channel 6 participates in the continuous mode 0b - Channel 6 is disabled 1b - Channel 6 is enabled
13	**Channel 5 Enable in Continuous Mode**

Yuntu Microelectronics

Field	Function
CH5EN	Channel 5 participates in the continuous mode 0b - Channel 5 is disabled 1b - Channel 5 is enabled
12 CH4EN	**Channel 4 Enable in Continuous Mode** Channel 4 participates in the continuous mode 0b - Channel 4 is disabled 1b - Channel 4 is enabled
11 CH3EN	**Channel 3 Enable in Continuous Mode** Channel 3 participates in the continuous mode 0b - Channel 3 is disabled 1b - Channel 3 is enabled
10 CH2EN	**Channel 2 Enable in Continuous Mode** Channel 2 participates in the continuous mode 0b - Channel 2 is disabled 1b - Channel 2 is enabled
9 CH1EN	**Channel 1 Enable in Continuous Mode** Channel 1 participates in the continuous mode 0b - Channel 1 is disabled 1b - Channel 1 is enabled
8 CH0EN	**Channel 0 Enable in Continuous Mode** Channel 0 participates in the continuous mode 0b - Channel 0 is disabled 1b - Channel 0 is enabled
2 MODE	**Continuous Mode** Continuous mode control 0b - Loop mode 1b - one-shot mode
1 CHFIX	**Fix Channel** Fix channel in continuous mode 0b - Fix negative channel, continuous happens on positive channels 1b - Fix positive channel, continuous happens on negative channels
0 EN	**Continuous Enable** Continuous Enable, ACMP module should be enabled 0b - Continuous mode is disabled 1b - Continuous mode is enabled

35.3.1.9 ACMP EXP Register

Offset: 24h

Yuntu Microelectronics

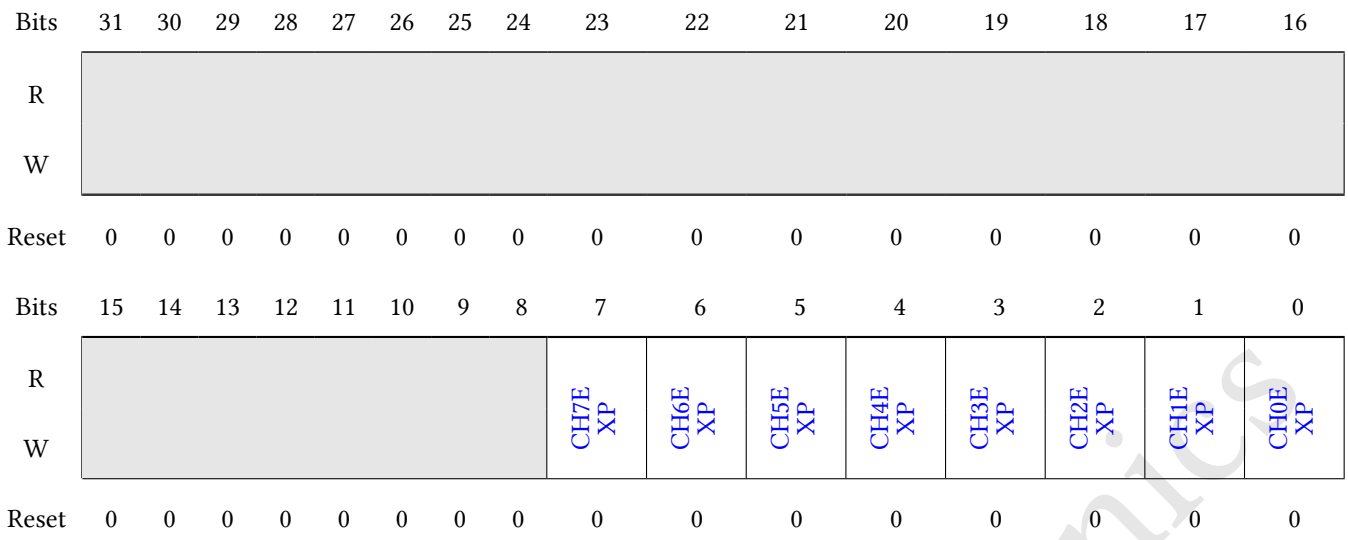


Table 35.15: ACMP EXP Register Description

Field	Function
7 CH7EXP	**Channel 7 Expect Value In Continuous Mode** Channel 7 expected value configuration before continuous mode enabled 0b - Indicate the channel 7 expected value is 0 1b - Indicate the channel 7 expected value is 1
6 CH6EXP	**Channel 6 Expect Value In Continuous Mode** Channel 6 expected value configuration before continuous mode enabled 0b - Indicate the channel 6 expected value is 0 1b - Indicate the channel 6 expected value is 1
5 CH5EXP	**Channel 5 Expect Value In Continuous Mode** Channel 5 expected value configuration before continuous mode enabled 0b - Indicate the channel 5 expected value is 0 1b - Indicate the channel 5 expected value is 1
4 CH4EXP	**Channel 4 Expect Value In Continuous Mode** Channel 4 expected value configuration before continuous mode enabled 0b - Indicate the channel 4 expected value is 0 1b - Indicate the channel 4 expected value is 1
3 CH3EXP	**Channel 3 Expect Value In Continuous Mode** Channel 3 expected value configuration before continuous mode enabled 0b - Indicate the channel 3 expected value is 0 1b - Indicate the channel 3 expected value is 1
2 CH2EXP	**Channel 2 Expect Value In Continuous Mode** Channel 2 expected value configuration before continuous mode enabled 0b - Indicate the channel 2 expected value is 0 1b - Indicate the channel 2 expected value is 1

Yuntu Microelectronics

Field	Function
1 CH1EXP	**Channel 1 Expect Value In Continuous Mode** Channel 1 expected value configuration before continuous mode enabled 0b - Indicate the channel 1 expected value is 0 1b - Indicate the channel 1 expected value is 1
0 CH0EXP	**Channel 0 Expect Value In Continuous Mode** Channel 0 expected value configuration before continuous mode enabled 0b - Indicate the channel 0 expected value is 0 1b - Indicate the channel 0 expected value is 1

35.4 Functional Description

The ACMP module has two 8-to-1 analog MUXs with eight analog input channels. The comparator output 1 when the positive port voltage is greater than the negative port voltage. Digital filter of comparator output is supported. ACMP module has four function modes: common mode, sample mode, window mode and continuous mode. The continuous mode includes two sub-modes: one-shot mode and loop mode. ACMP module supports to work in low power mode, and interrupt can wake up the chip from low power mode. In addition, the ACMP supports DMA.

35.4.1 Input Control

35.4.1.1 Input Selection

The input source of the ACMP can be selected by the CTRL and MUX register, there are two sources to choose from: the external signal from the pad and the internal signal from the DAC. The positive input source is selected by CTRL[INPSEL], and the negative input source is selected by CTRL[INNSEL]. ACMP module has two 8-to-1 analog MUXs with eight analog input channels which for external signal. The positive and negative input of the comparator can both select one of eight external signals by MUX[CHPSEL] and MUX[CHNSEL].

NOTE:

It is forbidden to select the same signal for both positive and negative input.

35.4.1.2 DAC (Digital to Analog Converter)

The ACMP has an internal 8-bit DAC module. The DAC reference voltage is selected between the external VREF and the internal VREF by configuring DAC[VREF]. The output voltage configured by DAC[VAL], the voltage value is (DAC[VAL]/255)*VREF. Setting DAC[EN]=1 to enable the DAC module.

35.4.2 Output Control

35.4.2.1 Polarity

The comparator output can be inverted or keep by configuring CTRL[POL] bit.

35.4.2.2 Filter

Comparator output result exists glitches. The digital filter is used to filter out glitches to avoid mistakes, it has a sampling counter with configurable sampling period. Filter is bypassed when FILT[CNT]=0 or

Yuntu Microelectronics

FILT[PER]=0.

As long as the comparator output changes, the sampling counter reset. If the comparator output remains unchanged, the sampling counter will accumulate to FILT[CNT]xFILT[PER] according to the sampling period and hold. Analog comparator output is regarded as a valid signal only when the sampling counter reaches the configured value (FILT[CNT]xFILT[PER]).

35.4.3 Function Mode

ACMP module has four function modes:

- common mode
- sample mode
- window mode
- continuous mode
  - one-shot mode
  - loop mode

35.4.3.1 Common Mode

ACMP module is in common mode when CTRL[SMPMD]=0. In common mode, the comparator output (STS[OUT]) reflects the comparison result of the selected two signals in real time. The interrupt and DMA request is generated according to STS[OUTNF]/STS[OUTPF].

35.4.3.2 Sample Mode

ACMP module is in sample mode when CTRL[SMPMD]=1. In sample mode, the comparator output (STS[OUT]) updates with the comparison result of the selected two signals when get sample signal from the TMU module (ACMP_SAMPLE_INPUT). The comparison result is sampled by hardware. The interrupt and DMA request is generated according to STS[OUTNF]/STS[OUTPF].

35.4.3.3 Window Mode

ACMP module is in window mode when CTRL[SMPMD]=2. In window mode, the comparator output (STS[OUT]) updates with the comparison result of the selected two signals during window signal keep high(ACMP_SAMPLE_INPUT=1). The comparison result works under window control. The interrupt and DMA request is generated according to STS[OUTNF]/STS[OUTPF].

35.4.3.4 Continuous Mode

ACMP module is in continuous mode when CTRL[SMPMD]=3. In continuous mode, the comparator output (STS[OUT]) reflects the comparison result of the current two signals in real time. Continuous mode includes two ways: one-shot mode and loop mode.

35.4.3.4.1 Channel Fix

In continuous mode, one of the two inputs is fixed and the other will scan the selected channels. The fixed input is chosen by CONT[CHFIX]. When CONT[CHFIX]=0, the negative input channel is fixed, and the positive input switches from the selected channels; When CONT[CHFIX]=1, the positive input channel is fixed, and the negative input port switches from the selected channels relatively. The input to be scanned is required to come from external analog signals, and the fixed input source could select an external analog signal or the internal Tempsenser output.

Yuntu Microelectronics

35.4.3.4.2 Channel Select

The scanned channels is enabled by configuring CONT[CHnEN] (n=0,1,...,7), the scan sequence is from channel 0 to channel 7. For example, when configure CONT[CHFIX]=0 and CONT[CHnEN]=1 (n=0,1,2,3), the negative port of the comparator fix and positive input channels (ACMP_IN0/ACMP_IN1/ACMP_IN2/ACMP_IN3) participate in automatic switching continuous one by one. If both two input sources select the external analog signal (MUX[INPSEL]=1 and MUX[INNSEL]=1), the fixed channel will not be scanned for the other input even if enabled by CONT[CHnEN].

35.4.3.4.3 Timing

The channel switching period is configurable. After channel switching stabilization time, the internal counter starts counting until CONT[PER] is reached, then the comparator switches to the next selected channel. When the internal counter reaches CONT[SMP], the comparator output result is sampled. Show as bellow "Continuous Mode of ACMP".

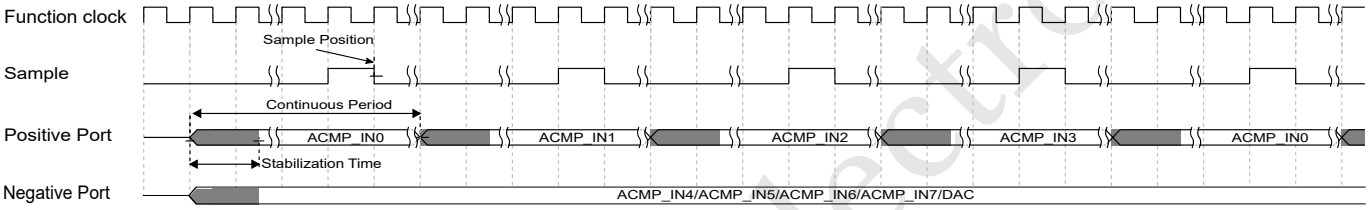


Figure 35.2: Continuous Mode of ACMP

NOTE:

The stabilization time is up to the hardware, it's less than 1 μs.

35.4.3.4.4 Scan Result

The scan result in continuous mode is reflected by STS[CHnF] (n=0,1,...,7). Before starting a continuous comparison, the initial value requests to be written into EXP[CHnEXP] (n=0,1,...,7). During the comparison, the initial value is regarded as the expected result and compared with the comparator output result. The corresponding flag set (STS[CHnF]=1) when the comparison result STS[CHnOUT] (n=0,1,...,7) of the current channel is not equal to the expected result EXP[CHnEXP] (n=0,1,...,7). The interrupt and DMA request is generated according to STS[CHnF].

35.4.3.4.5 Loop Mode

In continuous mode, loop mode is enabled by setting CONT[MODE]=0. In loop mode, the comparator restarts scanning the selected channels after finishing the comparison of the last channel. Configuring CTRL[EN]=0 to end the continuous comparison.

35.4.3.4.6 One-shot Mode

In continuous mode, one-shot mode is enabled by setting CONT[MODE]=1. In one-shot mode, the comparator starts with hardware trigger and stops the continuous comparison after finishing the comparison of the last channel.

The hardware trigger is from TMU and PTU modules, and user can configure the CIM register to select the trigger type. There are three types of triggers to be chosen:

Yuntu Microelectronics

- Raw signal which from the PTU module (CIM->CTRL0[ACMP0_TRIG_SEL]=0)
- Signal from the TMU module and synchronized by bus clock (CIM->CTRL0[ACMP0_TRIG_SEL]=1)
- Signal from the TMU module and synchronized by function clock (CIM->CTRL0[ACMP0_TRIG_SEL]=2)

NOTE:

Users should select the prepare trigger type according to the different applications.

35.4.4 Modes of Operation

ACMP module can work in all power modes including Active, Sleep, Deepsleep, Standby and Powerdown mode. it can generate interrupts to wake up MCU from Sleep, Deepsleep and Standby mode, it can also generate WKU event to wake up MCU from Powerdown mode. As show ACMP Power Mode Table.

35.4.5 Interrupts

ACMP module support asynchronous interrupt generation regardless of the existence of the bus clock. In continuous mode, the interrupt is generated when INTE[CONTIE]=1 and any channel flag set (STS[CHnF]=1). In other modes, select the expected interrupt edge by configuring CTRL[EDGESEL] first, then enable the interrupt by setting INTE[IE]=1, interrupt will be generated by STS[OUTNF]/STS[OUTPF] due to the comparator output changes. To clear the interrupt, write 1 to corresponding flag by software.

35.4.6 DMA Usage

The ACMP module supports DMA, this function is enabled by setting DMACR[EN]=1. DMA request generates when a valid edge defined in CTRL[EDGESEL] occurs. The flag STS[OUTNF] or STS[OUTPF] is set when a valid edge defined in CTRL[EDGESEL] occurs. DMA request is cleared by clearing the flag. The DMA priority is higher than the interrupt. As show ACMP DMA Table.

35.4.7 Reset

During a reset, the ACMP is in the default mode and disabled.

35.5 Glossary

ACMP	Analog Comparator
DAC	Digital-to-Analog Converter
MUX	Multiplexer
DMA	Direct Memory Access
VREF	Voltage Reference
WKU	Wake-up Unit

Yuntu Microelectronics

Chapter 36

Temperature Sensor (TEMPSENSOR)

36.1 Introduction

The temperature sensor provides a circuit to measure the temperature of the chip. It converts the temperature to the corresponding voltage, and ADC module can convert it to digital value through TEMP_SENSOR channel. According to the result of ADC conversion and the following formula, we can obtain the temperature of the chip.

$$T = C - K * V$$

K= 530

V is the voltage calculated from the ADC conversion result, formula is $V = (D/4096) * Vrefh$, Vrefh is the reference voltage of the ADC, D is the ADC conversion digital value.

C is an coefficient calibrated at 25°C.

For YTM32B1HA0x device, users can obtain C through reading the register TSC in the ADC.

Yuntu Microelectronics

Part IX

Security, Integrity and Safety

Yuntu Microelectronics

Chapter 37

Programmable Cyclic Redundancy Check (PCRC)

37.1 Introduction

The Programmable Cyclic Redundancy Check (PCRC)) module is used to compute CRC value for error detection. The CRC value is generated by programmable CRC polynomials. Bit-swap is optional for CRC input and output. Bit-inversion is only optional for CRC output.

37.1.1 Features

- Programmable CRC polynomials
- Programmable initial seed
- Optional bit-swap in one byte is available for input and output data
- Optional bit-swap in one word is available for input and output data
- Optional byte-swap in one word is available for input and output data
- Optional bit-inversion is available for output data
- 8/16/32-bit access for CRC input data

Yuntu Microelectronics

37.1.2 Block diagram

The following figure shows the CRC block diagram.

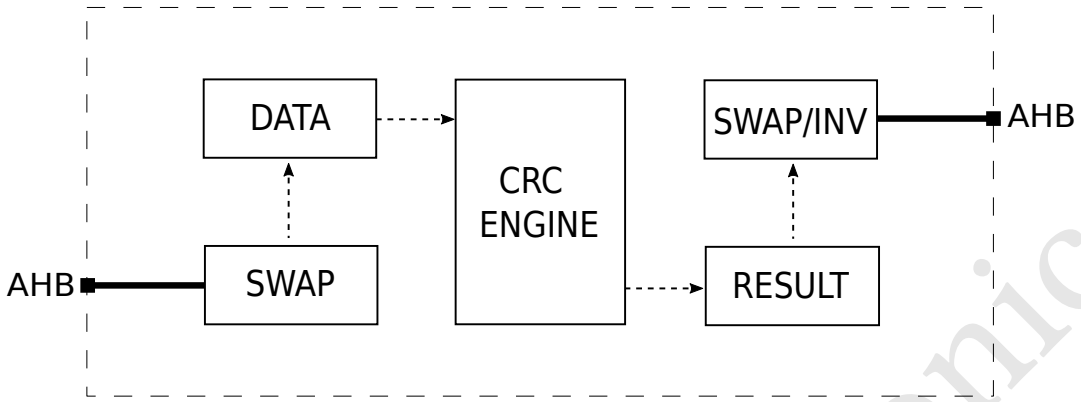


Figure 37.1: Block diagram of PCRC

37.2 Register Definition

37.2.1 PCRC Memory Map

PCRC base address 40194000h

Table 37.1: PCRC Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	00000000h
4h	Polynomial Register (POLY)	32	RW	00000000h
8h	Initial Seed Register (INIT)	32	RW	00000000h
ch	Data in 32 bits (DATA_IN32)	32	WO	00000000h
ch	Data in 16 bits (DATA_IN16)	16	WO	0000h
ch	Data in 8 bits (DATA_IN8)	8	WO	00h
10h	Result Register (RESULT)	32	RO	00000000h

37.2.1.1 PCRC CTRL Register

Offset: 0h

Yuntu Microelectronics

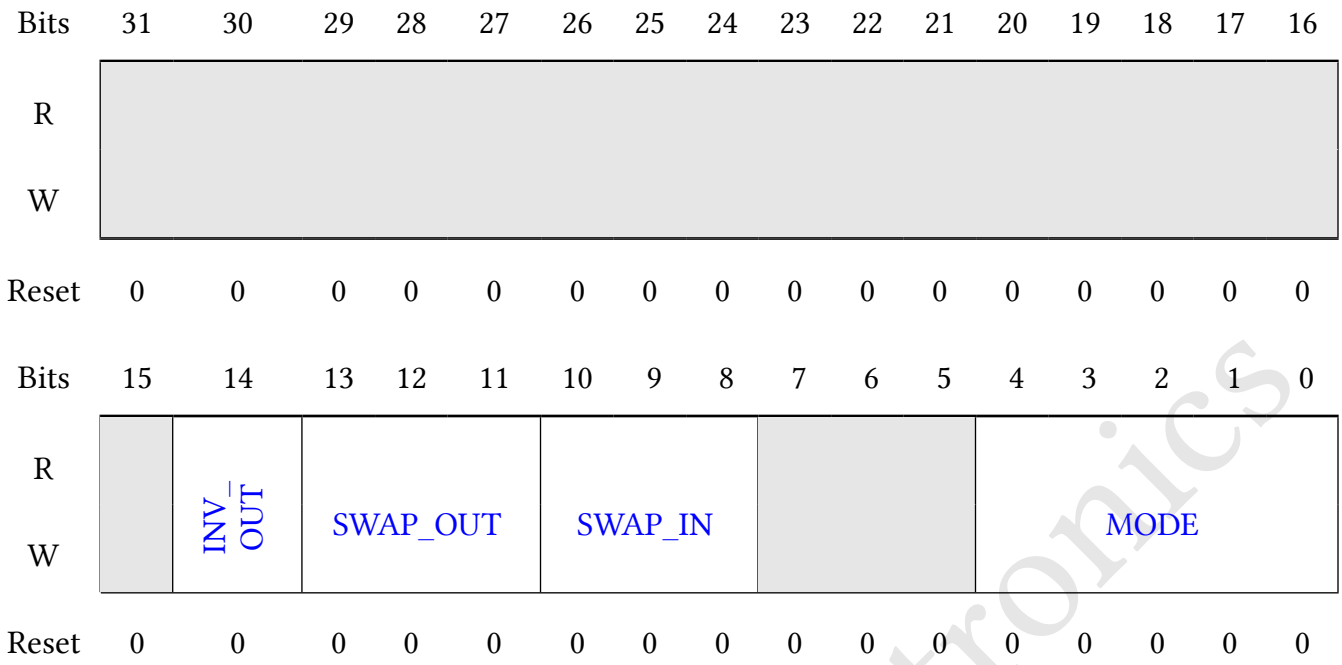


Table 37.2: PCRC CTRL Register Description

Field	Function
14 INV_OUT	**Invert Output** Bit-inversion for all CRC output bits. 0b - Disable the bit-inversion 1b - Enable the bit-inversion
13 - 11 SWAP_OUT	**Swap Output** Swap all the outputs bits for CRC output data. Please note that other register values will be no effect. 000b - Disable the output bit-swap 001b - Bit-swap in byte 010b - Bit-swap in word 100b - Byte-swap in word
10 - 8 SWAP_IN	**Swap Input** Swap all the Inputs bits for CRC input data. Please note that other register values will be no effect. 000b - Disable the input bit-swap 001b - Bit-swap in byte 010b - Bit-swap in word 100b - Byte-swap in word
4 - 0 MODE	**CRC Mode** Set CRC calculation mode. User should be aware of the correct crc mode of programmed polynomial.

Yuntu Microelectronics

37.2.1.2 PCRC POLY Register

Offset: 4h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	POLY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POLY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 37.3: PCRC POLY Register Description

Field	Function
31 - 0 POLY	**Polynomial** This register contains the programmable polynomial.

Yuntu Microelectronics

37.2.1.3 PCRC INIT Register

Offset: 8h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SEED															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SEED															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 37.4: PCRC INIT Register Description

Field	Function
31 - 0 SEED	**Initial Seed** This register contains the CRC initial seed value.

37.2.1.4 PCRC DATA_IN32 Register

Offset: ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 37.5: PCRC DATA_IN32 Register Description

Field	Function
31 - 0 DATA	**CRC Data Input** CRC data input in 32 bits

Yuntu Microelectronics

37.2.1.5 PCRC DATA_IN16 Register

Offset: ch

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 37.6: PCRC DATA_IN16 Register Description

Field	Function
15 - 0 DATA	**CRC Data Input** CRC data input in 16 bits

37.2.1.6 PCRC DATA_IN8 Register

Offset: ch

Bits	7	6	5	4	3	2	1	0
R								
W	DATA							
Reset	0	0	0	0	0	0	0	0

Table 37.7: PCRC DATA_IN8 Register Description

Field	Function
7 - 0 DATA	**CRC Data Input** CRC data input in 8 bits

Yuntu Microelectronics

37.2.1.7 PCRC RESULT Register

Offset: 10h

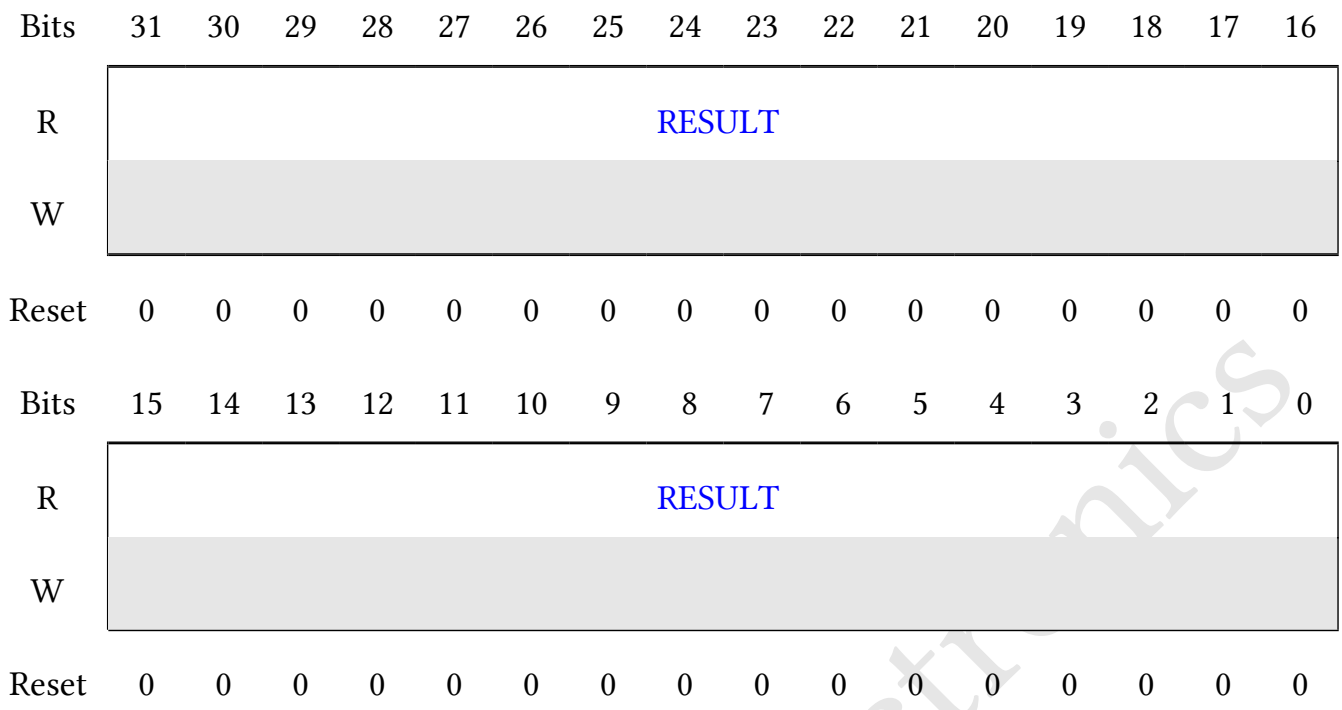


Table 37.8: PCRC RESULT Register Description

Field	Function
31 - 0 RESULT	**CRC Result** This register contains the result of CRC calculation. If bit-swap function is enabled, it may change the valid bits position, user should account for this situation.

37.3 Functional description

37.3.1 CRC initialization

Before the user starts a CRC calculation, it is necessary to set several configurations appropriately.

- Configure type of transpose and bit inversion if necessary.
- Program CRC polynomials
- Program initial CRC seed
- Program CRC input data to DATA register
- Get CRC result from RESULT register

37.3.2 CRC polynomials

The following polynomials for user's reference:

- CRC-4: $X^4 + X + 1$ = 0x3
- CRC-16: $X^{16} + X^{12} + X^5 + 1$ = 0x1021

Yuntu Microelectronics

- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ = 0x04C11DB7

37.3.3 Type of transpose

The CRC module provides the swap function for both writing the input data and reading the CRC result by using CTRL[SWAP_IN] or CTRL[SWAP_OUT] separately.

The following types of transpose function are available:

- If CTRL[SWAP_IN] or CTRL[SWAP_OUT] is 3'b001, the DATA[31:0] will become {DATA[24:31], DATA[16:23], DATA[8:15], DATA[0:7]}.

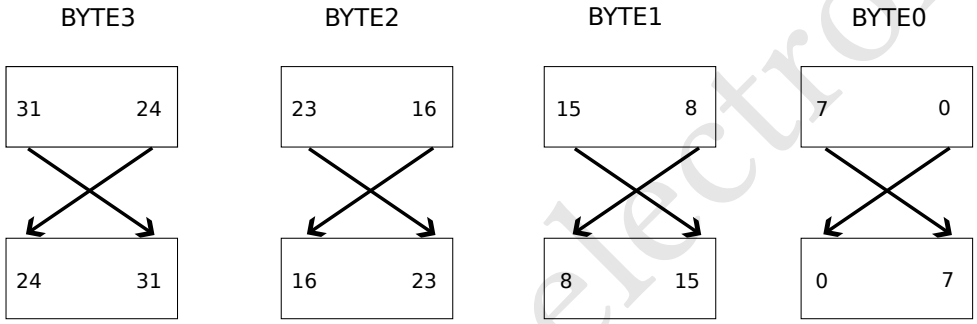


Figure 37.2: Bit swap in a Byte

- If CTRL[SWAP_IN] or CTRL[SWAP_OUT] is 3'b010, the DATA[31:0] will become {DATA[0], DATA[1], DATA[2], ..., DATA[30], DATA[31]}.

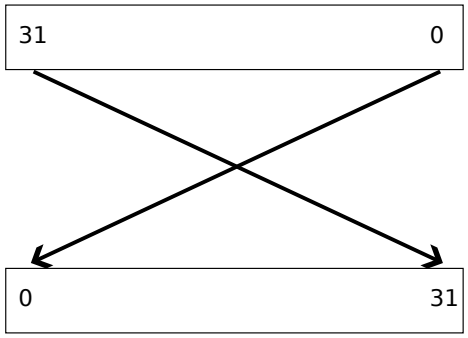


Figure 37.3: Bit swap in a Word

- If CTRL[SWAP_IN] or CTRL[SWAP_OUT] is 3'b100, the DATA[31:0] will become {DATA[7:0], DATA[15:8], DATA[23:16], DATA[31:24]}.

Yuntu Microelectronics

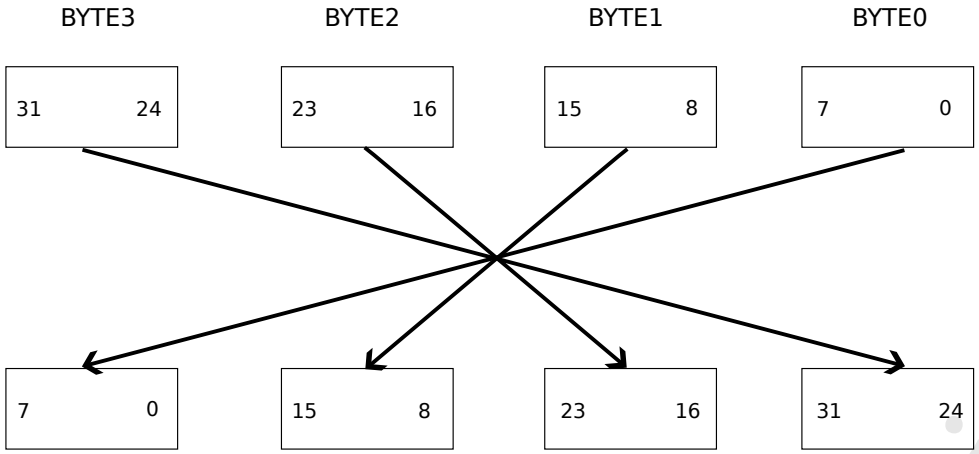


Figure 37.4: Byte swap in a Word

Please note for 16-bit and 8-bit write access, CRC will calculate on the valid bits. For 16-bit and 8-bit read access with CTRL[SWAP_OUT] is 3'b010 and 3'b100, user should be aware of the valid bits will be moved to high address position.

37.3.4 Bit inversion

This module also provides the invert function for reading the CRC result.

- If CTRL[INV_OUT] is 1, each bit of RESULT will be inverted (including valid and invalid bits).

Yuntu Microelectronics

Chapter 38

Hardware Cryptography Unit (HCU)

38.1 Chip-specific HCU information

38.1.1 Instantiation Information

YTM32B1HA0x contains one HCU module that can be used for data encryption/decryption. This module can perform AES, SM4, SHA algorithms based on the internal algorithm engines.

38.2 Introduction

38.2.1 Block diagram

The following figure shows the HCU block diagram.

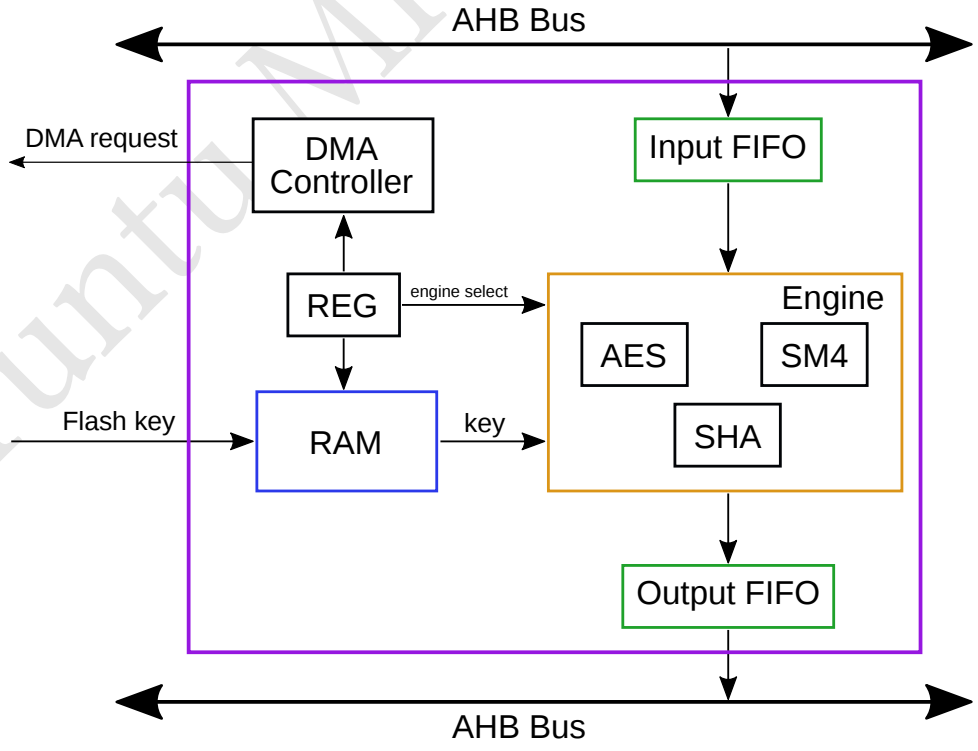


Figure 38.1: Block Diagram of HCU

Yuntu Microelectronics

38.2.2 Features

- Support 128-, 192- and 256-bit key length
- Support both encryption and decryption
- Support secure hardware key and flexible software key
- Support several algorithm engines
  - AES (ECB, CBC, CTR, CCM, CMAC)
  - SM4 (ECB)
  - SHA (SHA-256, SHA-384)
- All data input/output are in little-endian format
- Support bit, byte and half-word data swapping
- Size of input/output FIFO is up to 32*32 bits
- Support DMA transport between chip memory and input/output FIFO
- Support several interrupts
  - Operation done interrupt
  - Input FIFO empty interrupt
  - Output FIFO full interrupt
  - Input FIFO overflow interrupt
  - Output FIFO underflow interrupt
  - Input/output FIFO watermark interrupt
- Clock gating strategy is applied for engine core when input/output FIFO is not ready

38.2.3 Signal Description

The HCU module has no external pins.

38.3 Register Definition

The HCU registers are used to select the algorithm engine, configure the algorithm mode, control the module work and stop, prepare related data for the encryption/decryption, etc.

The HCU only allows 32-bit accesses.

Some registers are unwritten when GO[GO] is 1, they are CR (except OFDMAE & IFDMAE), MSGINFO, MSGADL.

NOTE. Accessing a reserved address will result in a bus error.
Register_address = Base_address + Address_offset
For example: as KEYn
Address_offset = 0x20 + n * 0x04 (n=0/1/2/3/4/5/6/7)

38.3.1 HCU Memory Map

HCU base address 4019c000h

Table 38.1: HCU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Go Register (GO)	32	RW	00000000h
4h	Configuration Register (CR)	32	RW	00000000h
8h	Message Information Register (MSGINFO)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
ch	Message and Additional Data Length Register (MSGADL)	32	RW	00000000h
14h	FIFO watermark Register (FIFOWM)	32	RW	00040004h
18h	Interrupt Enable Register (INTE)	32	RW	00000000h
1ch	Status Register (SR)	32	RW	00008500h
20h	Key (0~7) Register (KEY0)	32	WO	00000000h
24h	Key (0~7) Register (KEY1)	32	WO	00000000h
28h	Key (0~7) Register (KEY2)	32	WO	00000000h
2ch	Key (0~7) Register (KEY3)	32	WO	00000000h
30h	Key (0~7) Register (KEY4)	32	WO	00000000h
34h	Key (0~7) Register (KEY5)	32	WO	00000000h
38h	Key (0~7) Register (KEY6)	32	WO	00000000h
3ch	Key (0~7) Register (KEY7)	32	WO	00000000h
50h	AES Counter Value (0~3) Register (AESCV0)	32	RW	00000000h
54h	AES Counter Value (0~3) Register (AESCV1)	32	RW	00000000h
58h	AES Counter Value (0~3) Register (AESCV2)	32	RW	00000000h
5ch	AES Counter Value (0~3) Register (AESCV3)	32	RW	00000000h
60h	AES Initial Vector (0~3) Register (AESIV0)	32	RW	00000000h
64h	AES Initial Vector (0~3) Register (AESIV1)	32	RW	00000000h
68h	AES Initial Vector (0~3) Register (AESIV2)	32	RW	00000000h
6ch	AES Initial Vector (0~3) Register (AESIV3)	32	RW	00000000h
70h	AES MAC (0~3) Register (AESMAC0)	32	RW	00000000h
74h	AES MAC (0~3) Register (AESMAC1)	32	RW	00000000h
78h	AES MAC (0~3) Register (AESMAC2)	32	RW	00000000h
7ch	AES MAC (0~3) Register (AESMAC3)	32	RW	00000000h
80h	SHA ICV (0~11) Register (SHAICV0)	32	RW	00000000h
84h	SHA ICV (0~11) Register (SHAICV1)	32	RW	00000000h
88h	SHA ICV (0~11) Register (SHAICV2)	32	RW	00000000h
8ch	SHA ICV (0~11) Register (SHAICV3)	32	RW	00000000h
90h	SHA ICV (0~11) Register (SHAICV4)	32	RW	00000000h
94h	SHA ICV (0~11) Register (SHAICV5)	32	RW	00000000h
98h	SHA ICV (0~11) Register (SHAICV6)	32	RW	00000000h
9ch	SHA ICV (0~11) Register (SHAICV7)	32	RW	00000000h
a0h	SHA ICV (0~11) Register (SHAICV8)	32	RW	00000000h
a4h	SHA ICV (0~11) Register (SHAICV9)	32	RW	00000000h
a8h	SHA ICV (0~11) Register (SHAICV10)	32	RW	00000000h
ach	SHA ICV (0~11) Register (SHAICV11)	32	RW	00000000h
b0h	Input FIFO Data Register (IFDAT)	32	WO	00000000h
b4h	Output FIFO Data Register (OFDAT)	32	RO	00000000h

38.3.1.1 HCU GO Register

Offset: 0h

Yuntu Microelectronics

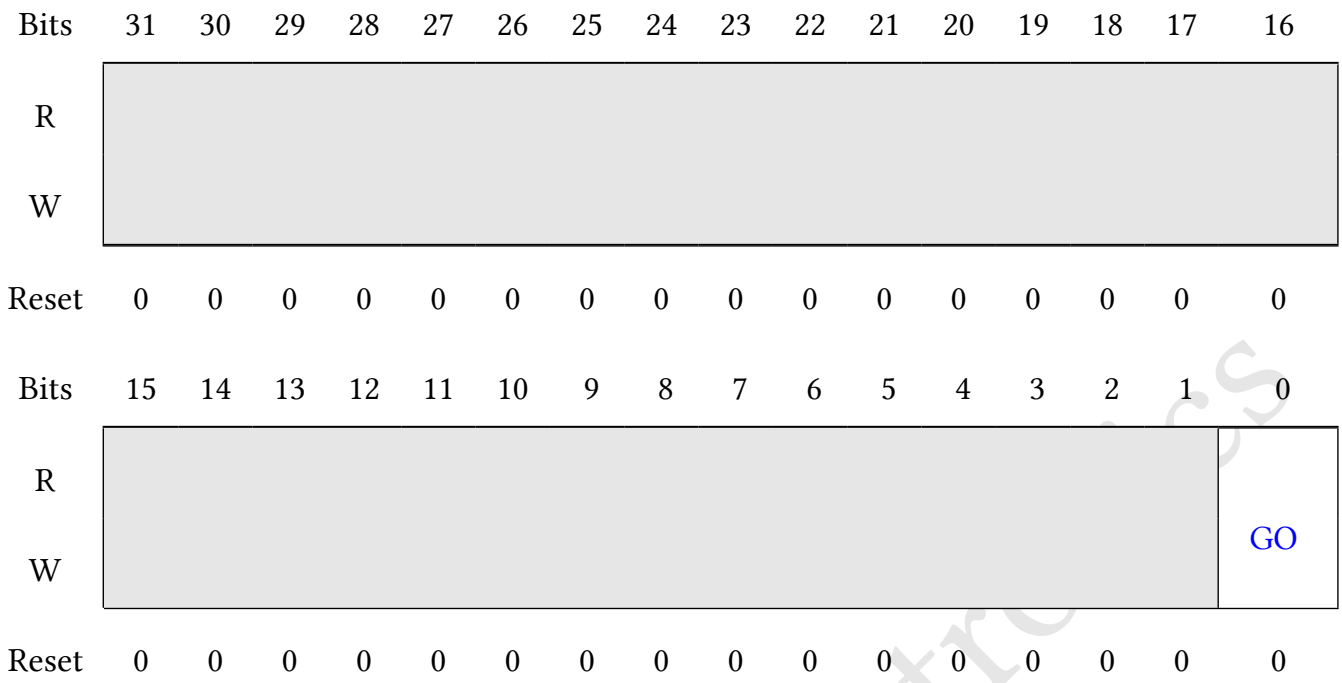
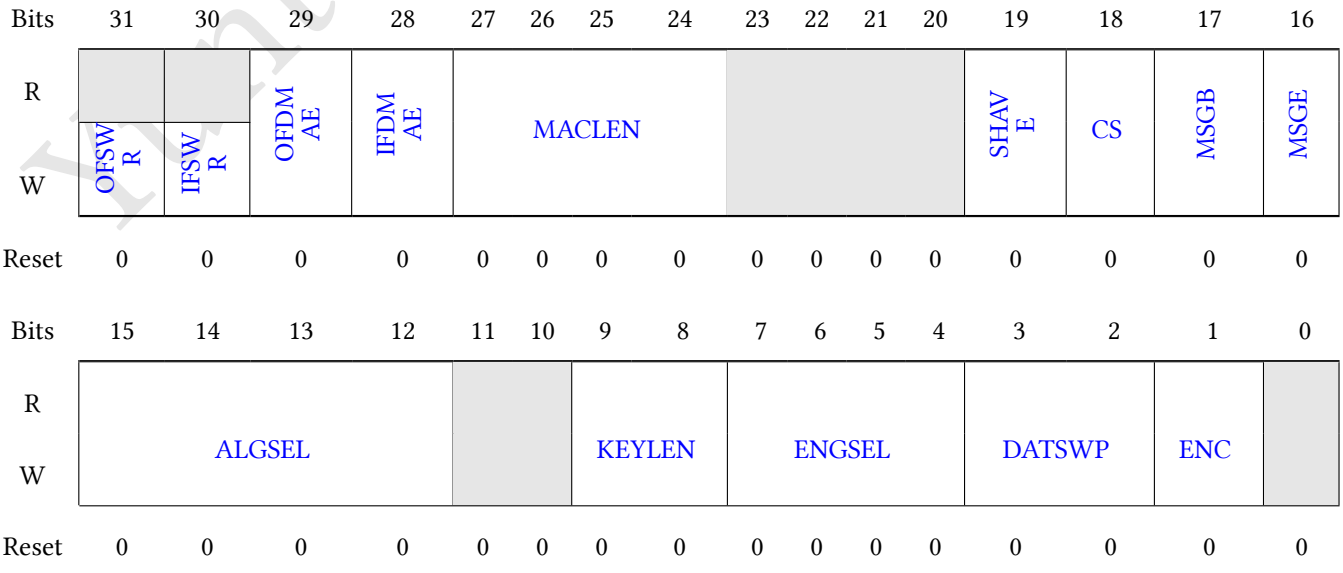


Table 38.2: HCU GO Register Description

Field	Function
0 GO	**Engine Run** Start Encryption/Decryption and indicate the module work state, write 0 to this bit is invalid. 0b - Encryption/Decryption idle. 1b - Encryption/Decryption running.

38.3.1.2 HCU CR Register

Offset: 4h



Yuntu Microelectronics

Table 38.3: HCU CR Register Description

Field	Function
31 OFSWR	**Output FIFO Software Reset** Reset output FIFO, output FIFO data will be cleared, this bit is cleared automatic. 0b - Don't reset the output FIFO. 1b - Reset the output FIFO.
30 IFSWR	**Input FIFO Software Reset** Reset input FIFO, input FIFO data will be cleared, this bit is cleared automatic. 0b - Don't reset the input FIFO. 1b - Reset the input FIFO.
29 OFDMAE	**Output FIFO DMA Enable** Output FIFO DMA enable/disable. 0b - Output FIFO DMA is disabled. 1b - Output FIFO DMA is enabled.
28 IFDMAE	**Input FIFO DMA Enable** Input FIFO DMA enable/disable. 0b - Input FIFO dma is disabled. 1b - Input FIFO dma is enabled.
27 - 24 MACLEN	**MAC Length** MAC length for generation or validation. 0000b - 16 bytes 0001b - 1 byte 0010b - 2 bytes 0011b - 3 bytes 0100b - 4 bytes 0101b - 5 bytes 0110b - 6 bytes 0111b - 7 bytes 1000b - 8 bytes 1001b - 9 bytes 1010b - 10 bytes 1011b - 11 bytes 1100b - 12 bytes 1101b - 13 bytes 1110b - 14 bytes 1111b - 15 bytes
19 SHAVE	**SHA Verification Enable** SHA hash result verification enable/disable. 0b - SHA hash result verification is disable. 1b - SHA hash result verification is enable.
18	**Context data Store**

Yuntu Microelectronics

Field	Function
CS	Store the context data, for multiple message blocks process. 0b - Don't store context data. 1b - Store context data.
17 MSGB	**Message Beginning** Indicate the first message block, for multiple message blocks process. 0b - Current data block isn't the first block of the whole data. 1b - Current data block is the first block of the whole data.
16 MSGE	**Message Ending** Indicate the last message block, for multiple message blocks process. 0b - Current data block isn't the last block of the whole data. 1b - Current data block is the last block of the whole data.
15 - 12 ALGSEL	**Algorithm Select** Select algorithm for selected engine. 0000b - AES-ECB/SM4-ECB/SHA-Reserved 0001b - AES-CBC/SM4-Reserved/SHA-256 0010b - AES-CTR/SM4-Reserved/SHA-384 0011b - AES-CCM/SM4-Reserved/SHA-Reserved 0100b - AES-CMAC/SM4-Reserved/SHA-Reserved
9 - 8 KEYLEN	**Key Length** Select key length for engine. 00b - 128-bit 01b - 192-bit 10b - 256-bit
7 - 4 ENGSEL	**Engine Selection** Select engine for encryption/decryption. 0000b - disable 0001b - AES 0010b - SM4 0011b - SHA
3 - 2 DATSWP	**Data Swapping** Select the type of data swapping. 00b - No swap 01b - Half-word swapping 10b - Byte swapping 11b - Bit swapping
1 ENC	**Encryption/Decryption Select** Select encryption or decryption mode. 0b - Decryption. 1b - Encryption.

Yuntu Microelectronics

38.3.1.3 HCU MSGINFO Register

Offset: 8h

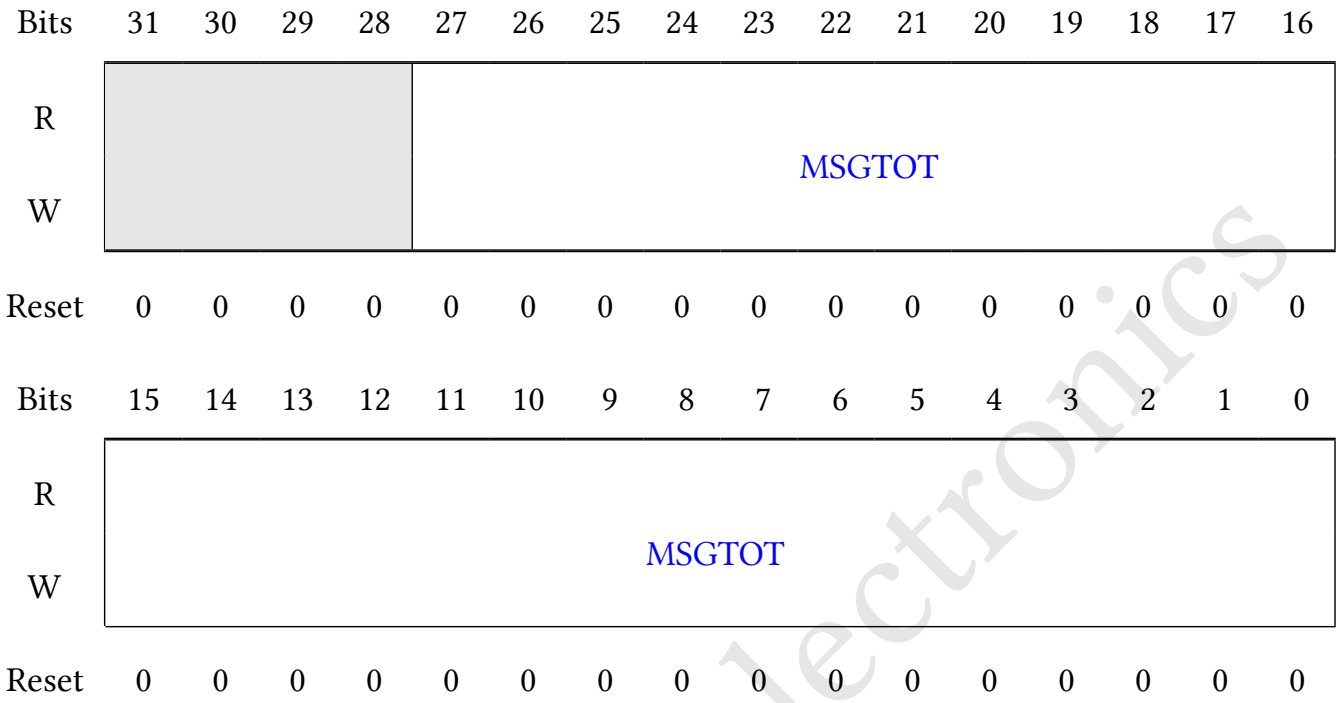


Table 38.4: HCU MSGINFO Register Description

Field	Function
27 - 0 MSGTOT	**Message Total Byte** Indicate byte number of message in total.

Yuntu Microelectronics

38.3.1.4 HCU MSGADL Register

Offset: ch

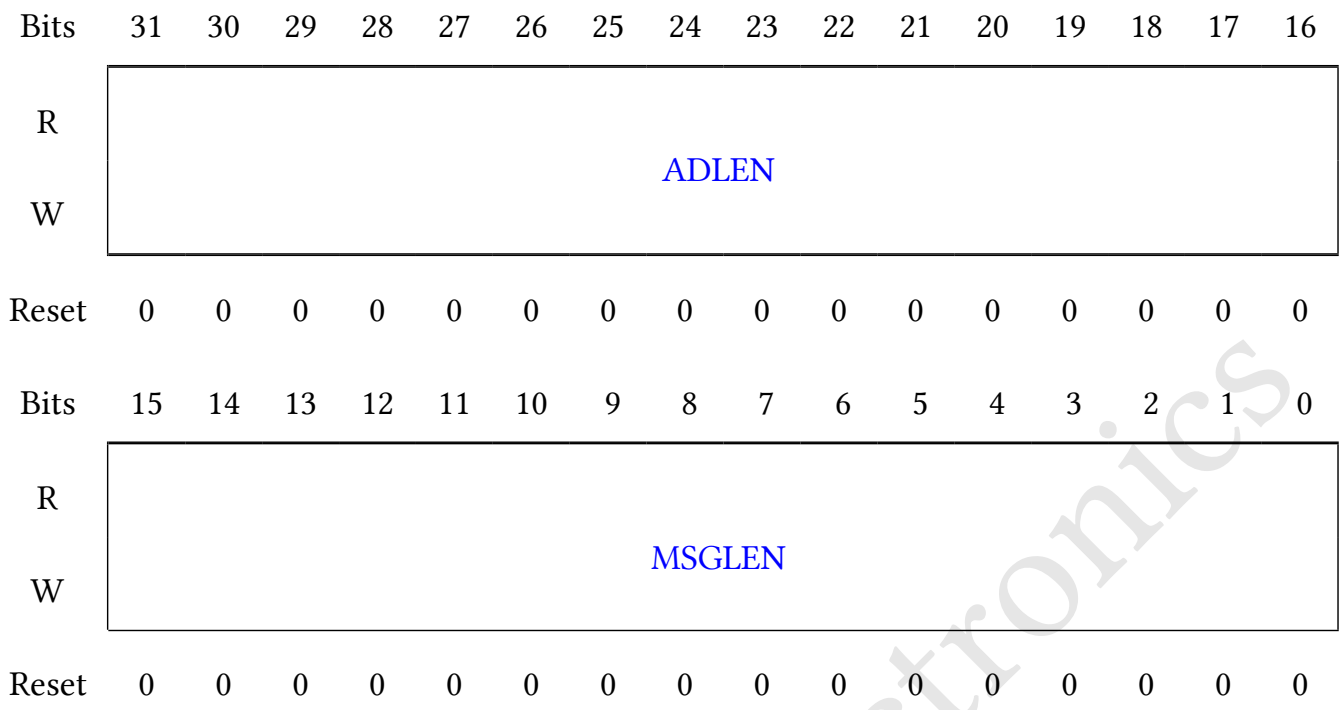


Table 38.5: HCU MSGADL Register Description

Field	Function
31 - 16 ADLEN	**Additional Data Length** Indicate byte number of additional data in current block. Only [30:16] field is valid for AES engine.
15 - 0 MSGLEN	**Message Length** Indicate byte number of message in current block. Only [14:0] field is valid for AES engine.

Yuntu Microelectronics

38.3.1.5 HCU FIFOWM Register

Offset: 14h

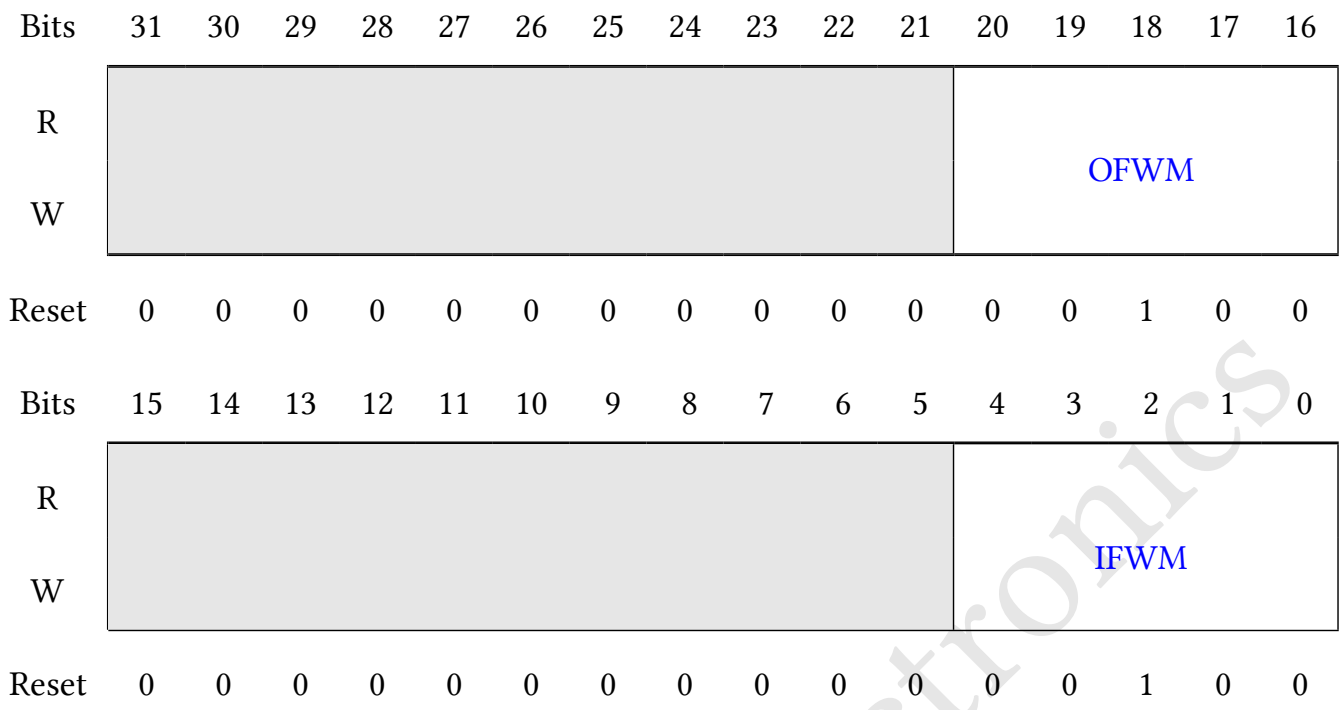


Table 38.6: HCU FIFOWM Register Description

Field	Function
20 - 16 OFWM	**Output FIFO Watermark** When the number of data in output FIFO is not less than the OFWM, interrupt or DMA request can be generated.
4 - 0 IFWM	**Input FIFO Watermark** When the number of data in input FIFO is less than the IFWM, interrupt or DMA request can be generated.

Yuntu Microelectronics

38.3.1.6 HCU INTE Register

Offset: 18h

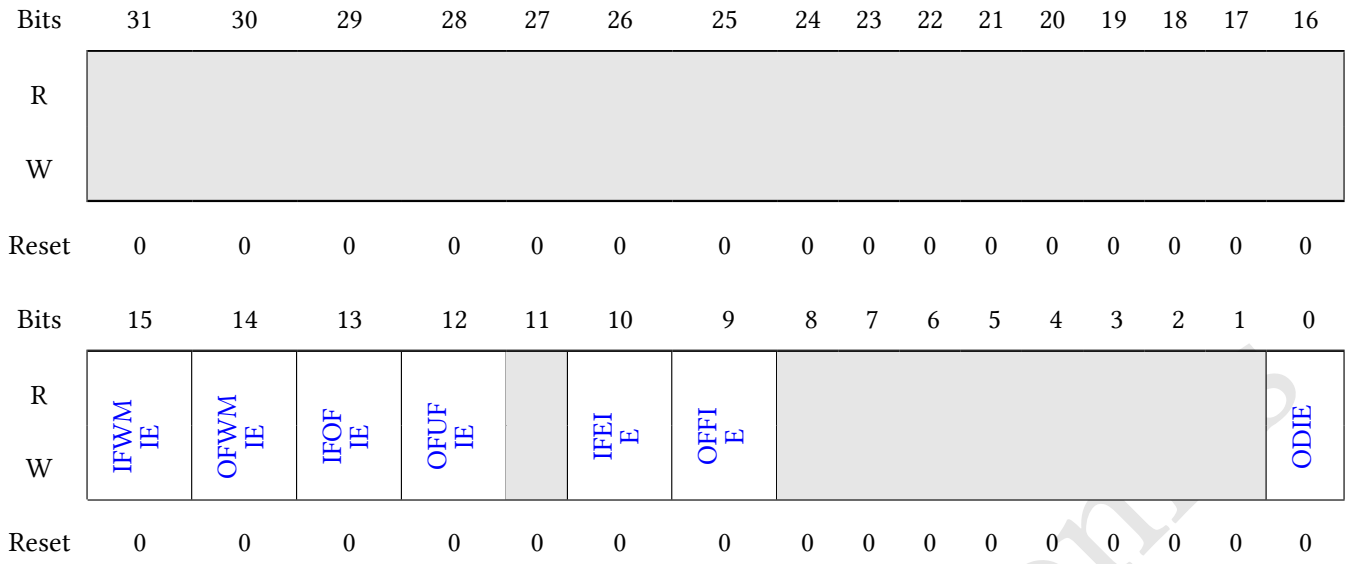


Table 38.7: HCU INTE Register Description

Field	Function
15 IFWMIE	**Input FIFO Watermark Interrupt Enable** Input FIFO watermark interrupt enable/disable. 0b - Input FIFO watermark interrupt disable. 1b - Input FIFO watermark interrupt enable.
14 OFWMIE	**Output FIFO Watermark Interrupt Enable** Output FIFO watermark interrupt enable/disable. 0b - Output FIFO watermark interrupt disable. 1b - Output FIFO watermark interrupt enable.
13 IFOFIE	**Input FIFO Overflow Interrupt Enable** Input FIFO overflow interrupt enable/disable. 0b - Input FIFO overflow interrupt disable. 1b - Input FIFO overflow interrupt enable.
12 OFUFIE	**Output FIFO Underflow Interrupt Enable** Output FIFO underflow interrupt enable/disable. 0b - Output FIFO underflow interrupt disable. 1b - Output FIFO underflow interrupt enable.
10 IFEIE	**Input FIFO Empty Interrupt Enable** Input FIFO empty interrupt enable/disable. 0b - Input FIFO Empty interrupt disable. 1b - Input FIFO empty interrupt enable.
9	**Output FIFO Full Interrupt Enable**

Yuntu Microelectronics

Field	Function
OFFIE	Output FIFO full interrupt enable/disable. 0b - Output FIFO full interrupt disable. 1b - Output FIFO full interrupt enable.
0 ODIE	**Operation Done Interrupt Enable** Operation done interrupt enable/disable. 0b - Operation done interrupt disable. 1b - Operation done interrupt enable.

38.3.1.7 HCU SR Register

Offset: 1ch

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IFWMF	OFWMF	IFOF F	OFUF F	IFFS	IFES	OFFS	OFES						SHAI VAL	AESV AL	OD
W			W1C	W1C												W1C
Reset	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Table 38.8: HCU SR Register Description

Field	Function
15 IFWMF	**Input FIFO Watermark Flag** Input FIFO watermark flag. 0b - Input FIFO watermark is not valid. 1b - Input FIFO watermark is valid.
14 OFWMF	**Output FIFO Watermark Flag** Output FIFO watermark flag. 0b - Output FIFO watermark is not valid. 1b - Output FIFO watermark is valid.
13	**Input FIFO Overflow Flag**

Yuntu Microelectronics

Field	Function
IFOFF	Input FIFO overflow indicate. Write 1 to clear this bit. 0b - No input FIFO overflow detected. 1b - Input FIFO overflow detected.
12 OFUFF	**Output FIFO Underflow Flag** Output FIFO underflow indicate. Write 1 to clear this bit. 0b - No output FIFO underflow detected. 1b - Output FIFO underflow detected.
11 IFFS	**Input FIFO Full Status** Indicate the input FIFO full state. 0b - Input FIFO is not full. 1b - Input FIFO is full.
10 IFES	**Input FIFO Empty Status** Indicate the input FIFO empty state. 0b - Input FIFO is not empty. 1b - Input FIFO is empty.
9 OFFS	**Output FIFO Full Status** Indicate the output FIFO full state. 0b - Output FIFO is not full. 1b - Output FIFO is full.
8 OFES	**Output FIFO Empty Status** Indicate the output FIFO empty state. 0b - Output FIFO is not empty. 1b - Output FIFO is empty.
2 SHAIVAL	**SHA hash Invalid** Indicate the verify result of input hash value. 0b - Input hash value is equal to the calculate result. 1b - Input hash value is not equal to the calculate result.
1 AESVAL	**AES MAC Valid** Indicate AES MAC validation result. 0b - The input MAC is not equal to the calculate result. 1b - The input MAC is equal to the calculate result.
0 OD	**Operation Done** Indicate Encryption/Decryption operation state. Write 1 to clear this bit. 0b - Operation doesn't begin or under running. 1b - Operation done.

38.3.1.8 HCU KEY Register

Offset: 20h + n * 4h

Yuntu Microelectronics

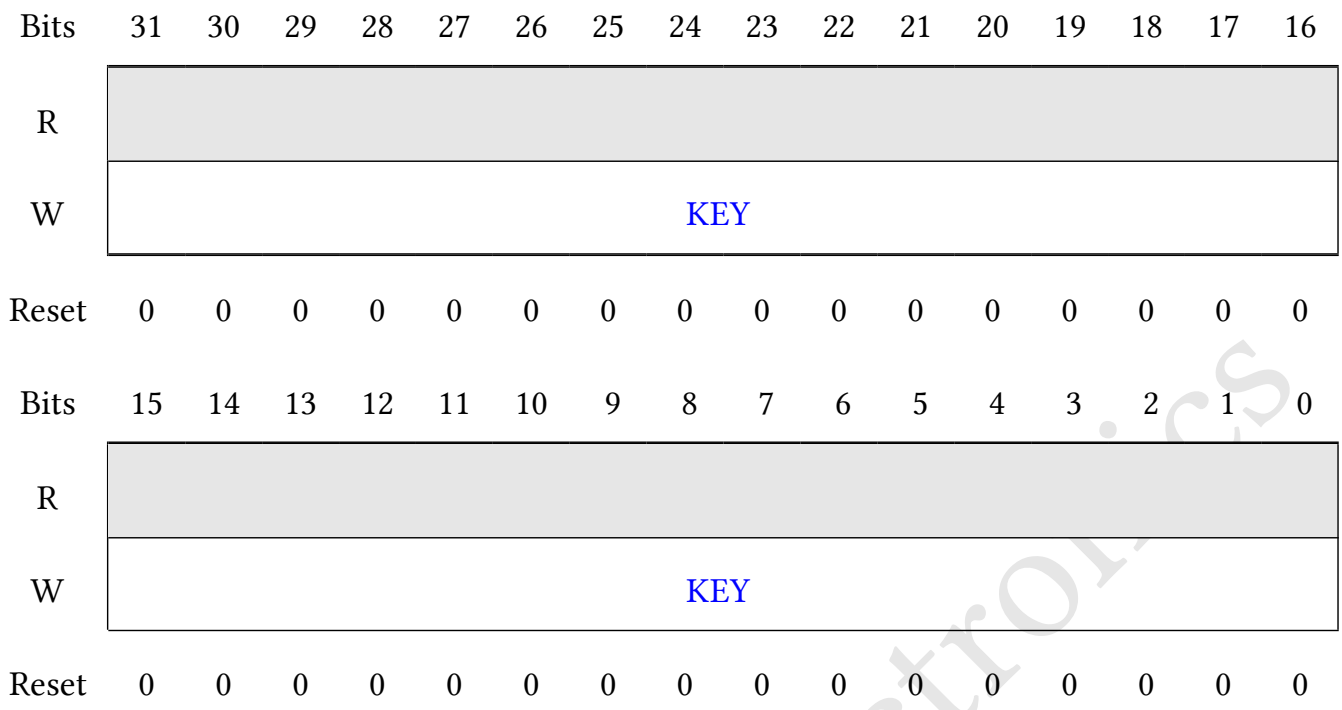


Table 38.9: HCU KEY Register Description

Field	Function
31 - 0 KEY	**Key value** Key value for Encryption/Decryption.

Yuntu Microelectronics

38.3.1.9 HCU AESCV Register

Offset: 50h + n * 4h

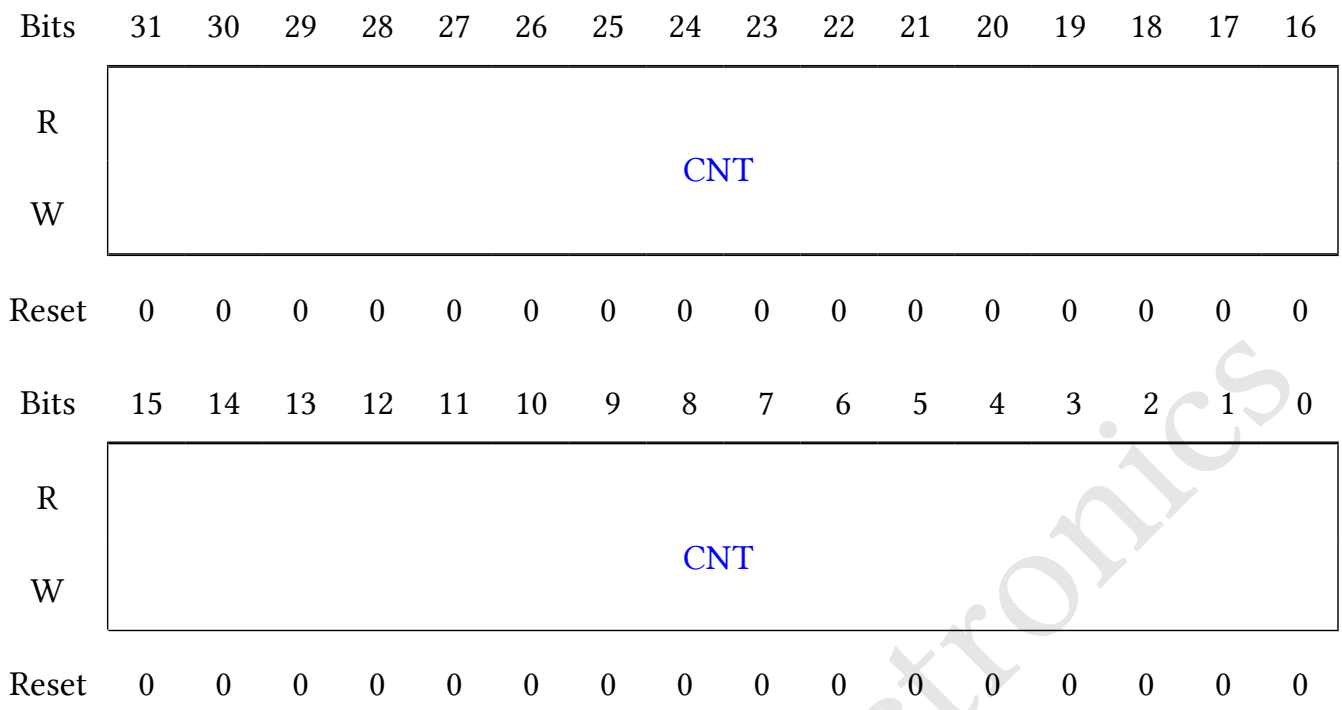


Table 38.10: HCU AESCV Register Description

Field	Function
31 - 0 CNT	**Counter Value** Counter value for Encryption/Decryption.

Yuntu Microelectronics

38.3.1.10 HCU AESIV Register

Offset: 60h + n * 4h

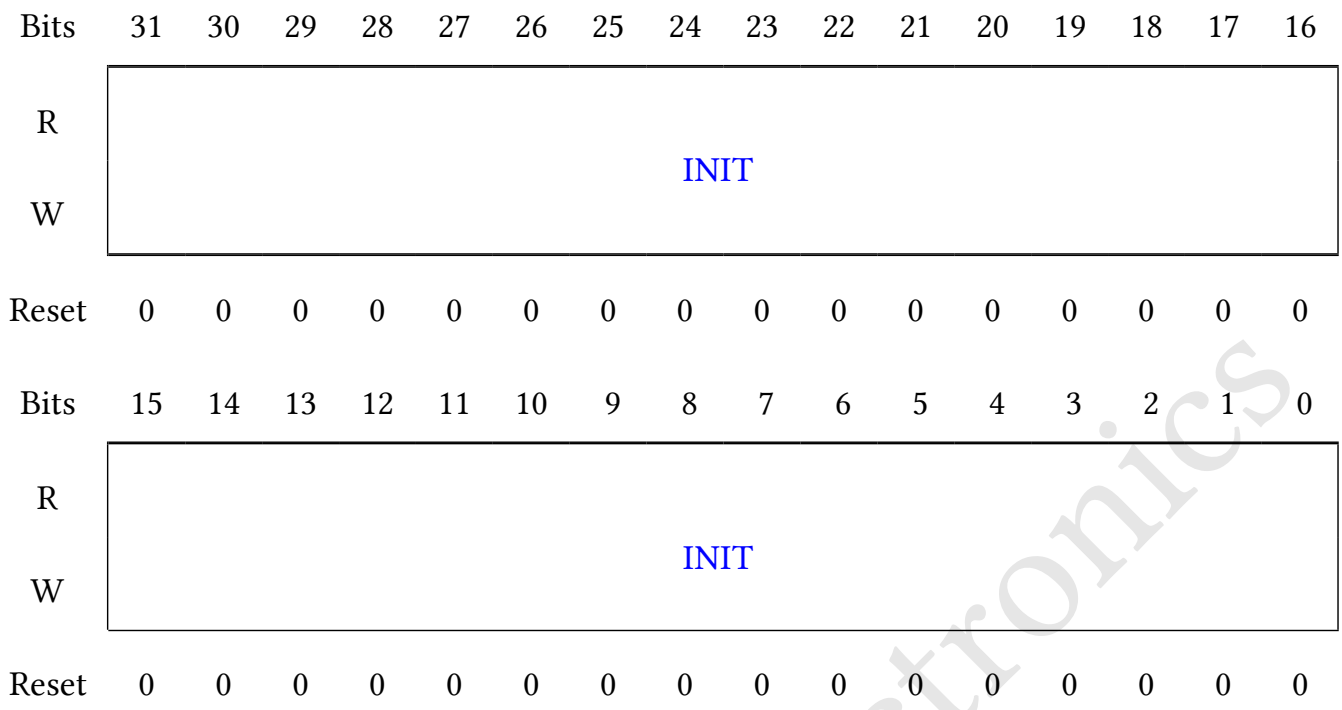


Table 38.11: HCU AESIV Register Description

Field	Function
31 - 0 INIT	**Initial Vector value** Initial vector value for Encryption/Decryption.

Yuntu Microelectronics

38.3.1.11 HCU AESMAC Register

Offset: 70h + n * 4h

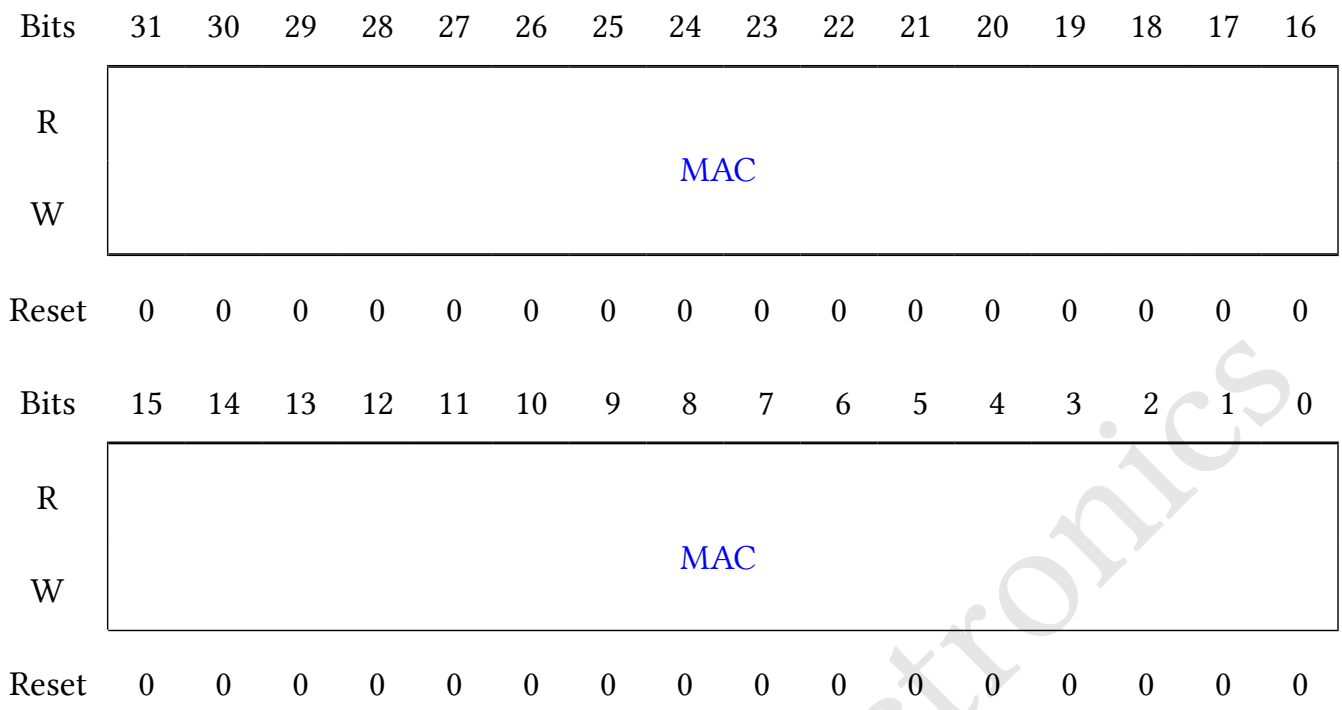


Table 38.12: HCU AESMAC Register Description

Field	Function
31 - 0 MAC	**MAC value** MAC value generated in Encryption or write for Decryption.

Yuntu Microelectronics

38.3.1.12 HCU SHAICV Register

Offset: 80h + n * 4h

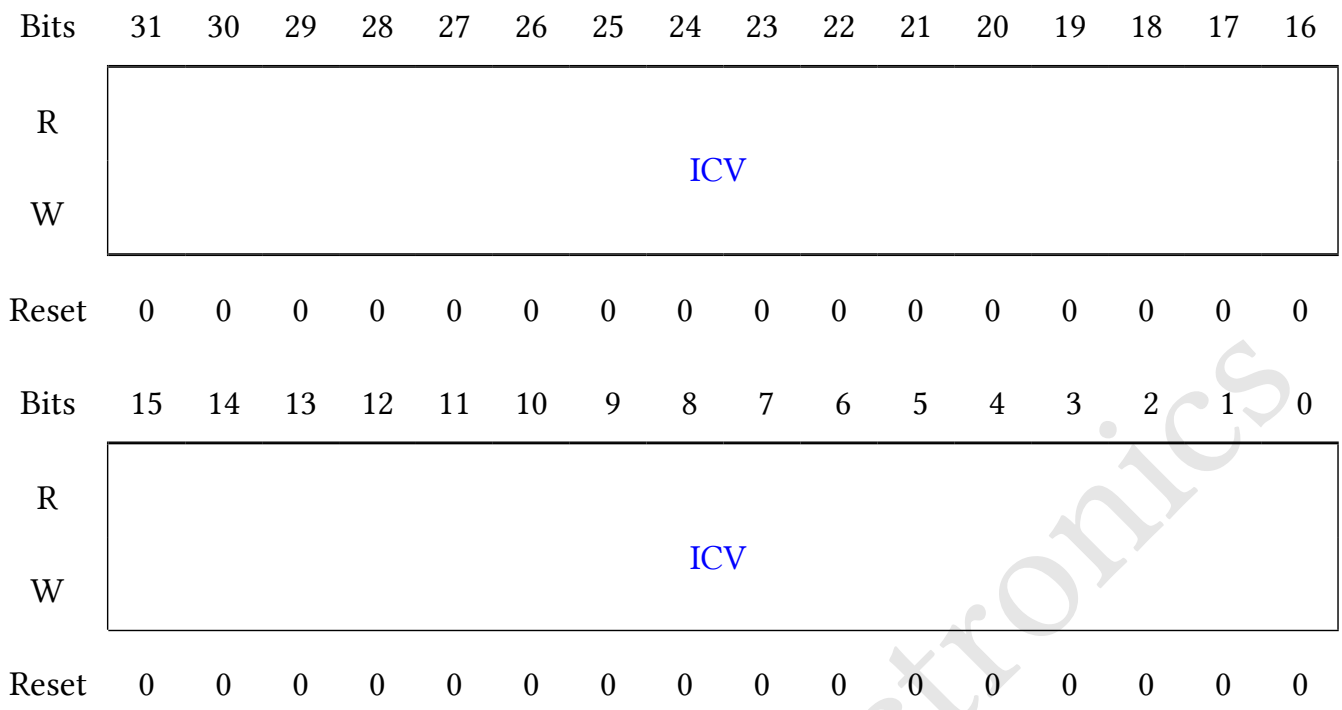


Table 38.13: HCU SHAICV Register Description

Field	Function
31 - 0 ICV	**ICV value** SHA input hash value for verification. Register 0~7 is used for SHA-256, register 0~11 is used for SHA-384.

Yuntu Microelectronics

38.3.1.13 HCU IFDAT Register

Offset: b0h

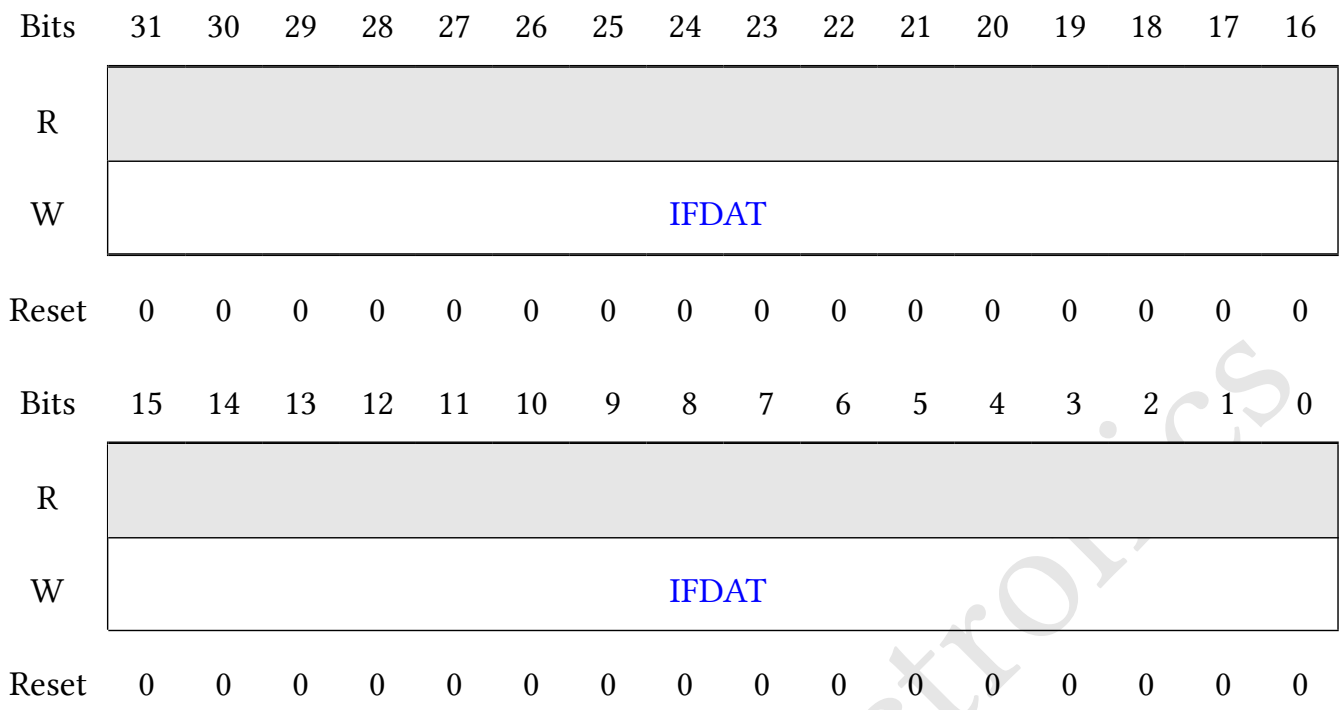


Table 38.14: HCU IFDAT Register Description

Field	Function
31 - 0 IFDAT	**Input FIFO Data** Input FIFO Data, write data that to be encrypted/decrypted. The data should be written in 128 bits.

Yuntu Microelectronics

38.3.1.14 HCU OFDAT Register

Offset: b4h

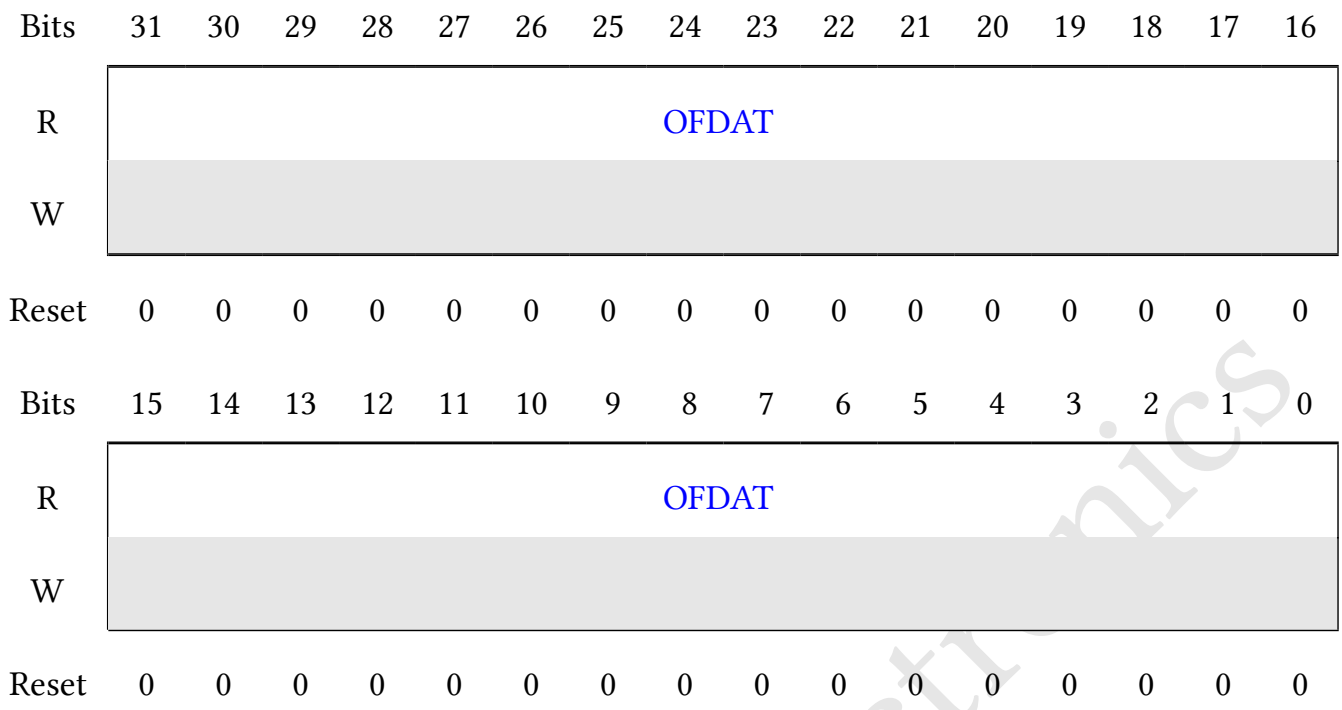


Table 38.15: HCU OFDAT Register Description

Field	Function
31 - 0 OFDAT	**Output FIFO Data** Output FIFO Data, read data that after encryption/decryption.

38.4 Functional description

38.4.1 General

This section gives detailed information on the internal operation of the module. The HCU module contains several algorithm engines: AES, SM4, and SHA. The AES engine supports 128-bit, 192-bit and 256-bit key size, and it supports multiple modes for data encryption/decryption, they are AES-ECB, AES-CBC, AES-CTR, AES-CCM, and AES-CMAC. The SM4 engine only supports 128-bit key size, and it only supports SM4-ECB mode. The SHA engine supports SHA-256 and SHA-384 mode.

38.4.2 Use Flow

To use this module correctly, please follow the procedure below.

38.4.2.1 AES Engine

The AES engine supports ECB, CBC, CTR, CCM and CMAC mode.

38.4.2.1.1 ECB

1. Write key value if you need, configure KEY registers.

Yuntu Microelectronics

2. Select AES engine, configure CR[ENGSEL].
3. Select ECB algorithm, configure CR[ALGSEL].
4. Select key length, configure CR[KEYLEN].
5. Write data into input FIFO, write IFDAT register.
6. Indicate message data size, configure MSGADL[MSGLEN].
7. Select encryption or decryption, configure CR[ENC].
8. Start encryption or decryption, set GO[GO].
9. Wait operation done, wait SR[OD] = 1.
10. Read data from output FIFO, read OFDAT register.

38.4.2.1.2 CBC

1. Write key value if you need, configure KEY registers.
2. Select AES engine, configure CR[ENGSEL].
3. Select CBC algorithm, configure CR[ALGSEL].
4. Select key length, configure CR[KEYLEN].
5. Write IV value, configure AESIV registers.
6. Write data into input FIFO, write IFDAT register.
7. Indicate message data size, configure MSGADL[MSGLEN].
8. Select encryption or decryption, configure CR[ENC].
9. Start encryption or decryption, set GO[GO].
10. Wait operation done, wait SR[OD] = 1.
11. Read data from output FIFO, read OFDAT register.

38.4.2.1.3 CTR

1. Write key value if you need, configure KEY registers.
2. Select AES engine, configure CR[ENGSEL].
3. Select CTR algorithm, configure CR[ALGSEL].
4. Select key length, configure CR[KEYLEN].
5. Write CTR value, configure AESCV registers.
6. Write data into input FIFO, write IFDAT register.
7. Indicate message data size, configure MSGADL[MSGLEN].
8. Select encryption or decryption, configure CR[ENC].
9. Start encryption or decryption, set GO[GO].
10. Wait operation done, wait SR[OD] = 1.
11. Read data from output FIFO, read OFDAT register.

38.4.2.1.4 CCM

1. Write key value if you need, configure KEY registers.
2. Select AES engine, configure CR[ENGSEL].
3. Select CCM algorithm, configure CR[ALGSEL].
4. Select key length, configure CR[KEYLEN].
5. Write CTR value, configure AESCV registers.
6. Write data into input FIFO, write IFDAT register, write additional data first, followed with message data.
7. Indicate additional data size, configure MSGADL[ADLEN].

Yuntu Microelectronics

8. Indicate message data size, contain additional data size, configure MSGADL[MSGLEN].
9. Indicate MAC length to generate or validate, configure CR[MACLEN].
10. Indicate only one message block, set CR[MSGE], set CR[MSGB].
11. Select encryption or decryption, configure CR[ENC].
12. Write MAC value if do decryption, configure AESMAC registers.
13. Start encryption or decryption, set GO[GO].
14. Wait operation done, wait SR[OD] = 1.
15. Read data from output FIFO, read OFDAT register.
16. Read MAC value if do encryption, read AESMAC registers.
17. Check validation result if do decryption, read SR[AESVAL].

38.4.2.1.5 CMAC

1. Write key value if you need, configure KEY registers.
2. Select AES engine, configure CR[ENGSEL].
3. Select CMAC algorithm, configure CR[ALGSEL].
4. Select key length, configure CR[KEYLEN].
5. Write data into input FIFO, write IFDAT register.
6. Indicate message data size, configure MSGADL[MSGLEN].
7. Indicate MAC length to generate or validate, configure CR[MACLEN].
8. Indicate only one message block, set CR[MSGB], set CR[MSGE].
9. Select encryption for MAC generation or decryption for MAC validation, configure CR[ENC].
10. Write MAC value if do decryption, configure AESMAC registers.
11. Start encryption or decryption, set GO[GO].
12. Wait operation done, wait SR[OD] = 1.
13. Read MAC value if do encryption, read AESMAC registers.
14. Check validation result if do decryption, read SR[AESVAL].

38.4.2.2 SM4 Engine

The SM4 engine only supports ECB mode.

38.4.2.2.1 ECB

1. Write key value if you need, configure KEY registers.
2. Select SM4 engine, configure CR[ENGSEL].
3. Select ECB algorithm, configure CR[ALGSEL].
4. Write data into input FIFO, write IFDAT register.
5. Indicate message data size, configure MSGADL[MSGLEN].
6. Select encryption or decryption, configure CR[ENC].
7. Start encryption or decryption, set GO[GO].
8. Wait operation done, wait SR[OD] = 1.
9. Read data from output FIFO, read OFDAT register.

38.4.2.3 SHA Engine

The SHA engine supports SHA-256 and SHA-384 mode.

Yuntu Microelectronics

38.4.2.3.1 SHA-256/SHA-384 generate hash value

1. Select SHA engine, configure CR[ENGSEL].
2. Select SHA-256 or SHA-384 algorithm, configure CR[ALGSEL].
3. Write data into input FIFO, write IFDAT register.
4. Indicate message data size, configure MSGADL[MSGLEN].
5. Indicate total message data size, configure MSGINFO[MSGTOT]=MSGADL[MSGLEN].
6. Indicate only one data block, set both CR[MSGB] and CR[MSGE].
7. Start hash value generation, set GO[GO].
8. Wait operation done, wait SR[OD] = 1.
9. Read generated hash value from output FIFO, read OFDAT register.

38.4.2.3.2 SHA-256/SHA-384 verify hash value

1. Select SHA engine, configure CR[ENGSEL].
2. Select SHA-256 or SHA-384 algorithm, configure CR[ALGSEL].
3. Write data into input FIFO, write IFDAT register.
4. Indicate message data size, configure MSGADL[MSGLEN].
5. Indicate total message data size, configure MSGINFO[MSGTOT]=MSGADL[MSGLEN].
6. Indicate only one data block, set both CR[MSGB] and CR[MSGE].
7. Write hash value into SHAICV registers.
8. Enable hash value verification, configure CR[SHAVE]=1.
9. Start hash value verification, set GO[GO].
10. Wait operation done, wait SR[OD] = 1.
11. Check hash value verification state, read SR[SHAIVAL].
12. Read generated hash value from output FIFO, read OFDAT register.

38.4.3 Multiple Message Blocks For AES/SHA

Usually the AES/SHA engine processes one message block which less than 65536 bytes, just indicate the message byte number by configuring MSGADL[MSGLEN]. If you want to encrypt or decrypt a message which contains more than 65536 bytes, multiple message blocks is needed, you need to indicate the first message block, the middle message block and the last message block by configuring CR[MSGB] and CR[MSGE], and also you should to store some context data into internal memory by setting CR[CS].

The general operation method is as follows

- Configure MSGINFO register to indicate total message data size, not include additional data size in AES-CCM
- Set CR[CS] to enable the context data storing
- ……
- Set CR[MSGB] and clear CR[MSGE] to indicate the first message block
- Configure MSGADL[MSGLEN] to indicate message block data size
- Start encryption or decryption, set GO[GO].
- Wait operation done, wait SR[OD] = 1.
- ……
- Clear CR[MSGB] and clear CR[MSGE] to indicate the middle message block
- Configure MSGADL[MSGLEN] to indicate message block data size
- Start encryption or decryption, set GO[GO].

Yuntu Microelectronics

- Wait operation done, wait SR[OD] = 1.
- ……
- Clear CR[MSGB] and set CR[MSGE] to indicate the last message block
- Configure MSGADL[MSGLEN] to indicate message block data size
- Start encryption or decryption, set GO[GO].
- Wait operation done, wait SR[OD] = 1.
- ……

NOTE:
1. This function is only available in AES(CCM/CMAC) and SHA mode.
2. Block size should be divided into times of 64 bytes in SHA-256 and 128 bytes in SHA-384.

38.4.4 Context Store

Usually, when we start new encryption or decryption in AES-CBC or AES-CTR mode, the IV and CTR values will load from the register. If CR[CS] set, after the data block process is done, the new IV and CTR will be stored in the register. The next data block process will use the new IV and CTR. It's useful for long data processes in AES-CBC and AES-CTR mode.

38.4.5 Secure key

The HCU module supports hardware key and software key, up to 128 groups 256-bit keys are stored in flash as the hardware key, to use them for encryption or decryption, you should configure EFM register to load the key. Also you can configure the KEY registers to use the software key. For security reasons, both hardware key and software key are unreadable.

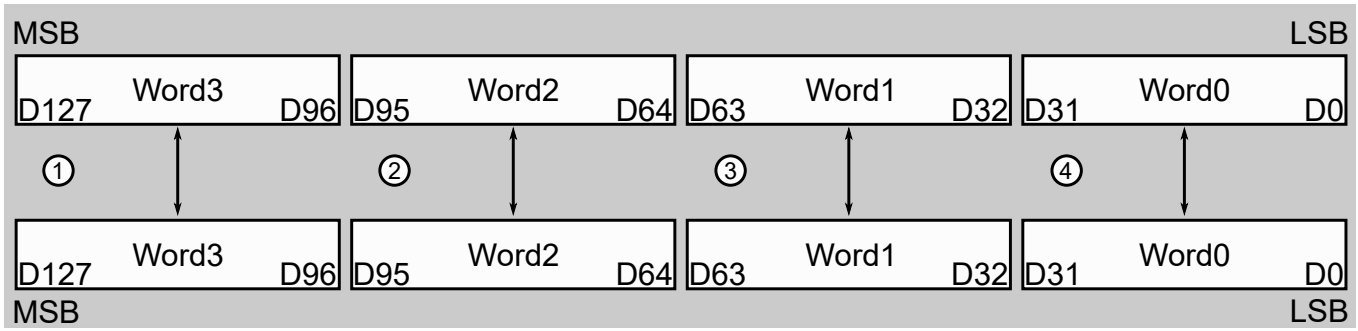
38.4.6 Data swapping

The HCU module can be configured to perform bit swapping, byte swapping, half-word swapping or no swapping on the input data and output data. The data swap type is selected through CR[DATSWP], the selection is applied for both input data and output data.

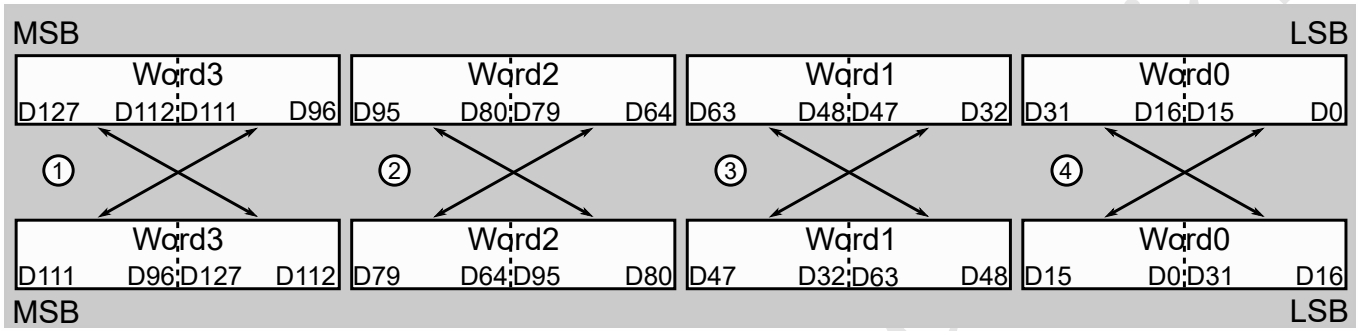
The following figure shows the different data swap selection.

Yuntu Microelectronics

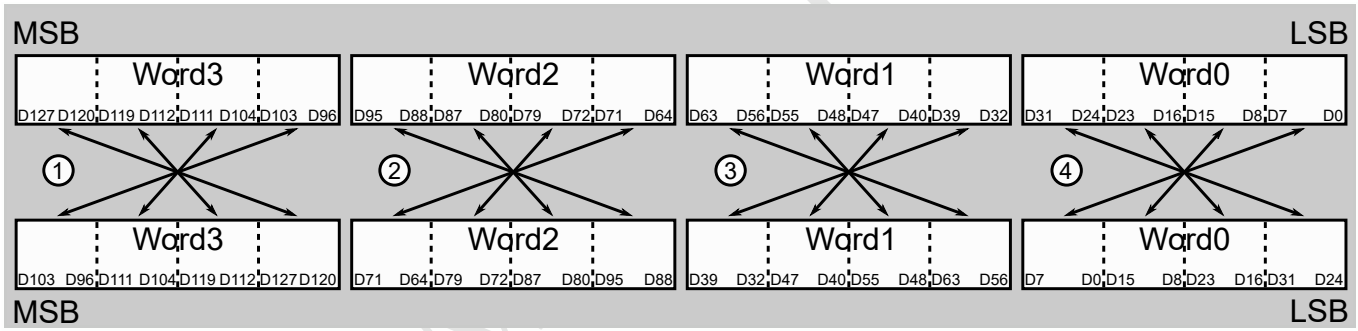
CR[DATSWP] = 00 : no swapping



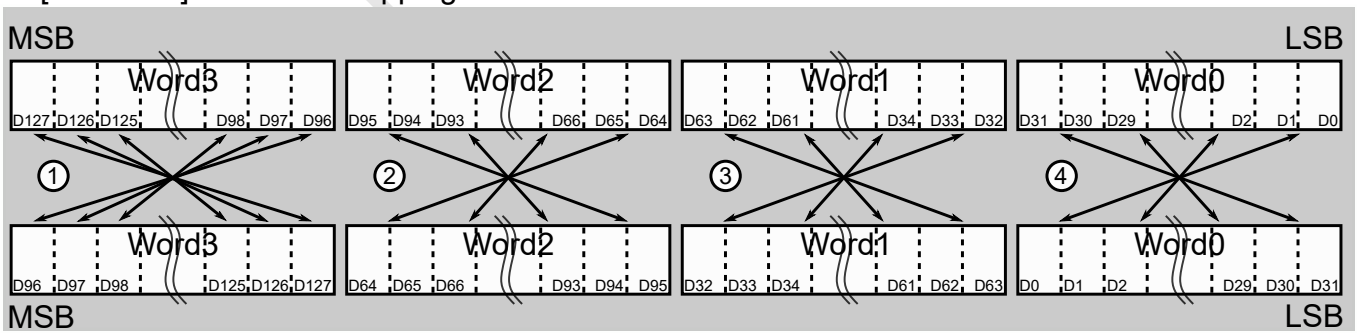
CR[DATSWP] = 01 : half-word swapping



CR[DATSWP] = 10 : byte swapping



CR[DATSWP] = 11 : bit swapping



NOTE: ① ~ ④ indicate the order of writing to IFDAT or reading from OFDAT.

Figure 38.2: Data swap type of 128-bit data block

38.4.7 Interrupts

HCU module just support one interrupt, please refer to the table of interrupt vectors. There are seven sources to generate the HCU interrupt, the detail information please refer to the interrupt enable register

(INTE). For operation done interrupt, input FIFO overflow interrupt and output FIFO underflow interrupt, their interrupt flags can be cleared by SR register. Other interrupts does't need to clear the interrupt flag, they are just relate to the FIFO status.

38.4.8 DMA

HCU module supports DMA to transport between chip memory and input/output FIFO, configure CR[IFDMAE] and CR[OFDMAE] to enable or disable DMA function. Whenever the input FIFO DMA is enabled and input FIFO data number is less than the watermark, input FIFO DMA request will be generated. Whenever the output FIFO DMA is enabled and output FIFO data number is not less than the watermark, output FIFO DMA request will be generated.

38.4.9 Clock Gating Strategy

If the input/output FIFO is not ready, the clock of engine will be locked, and the encryption or decryption will not end. Read input FIFO when it's empty will result in input FIFO not ready, write output FIFO when it's full will result in output FIFO not ready.

38.5 Glossary

HCU	Hardware Cryptography Unit
AES	Advanced Encryption Standard
ECB	Electronic Codebook
CBC	Cipher Block Chaining
CTR	Counter
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CMAC	Cipher-based Message Authentication Code
SM4	Shang Mi 4
SHA	Secure Hash Algorithm

Yuntu Microelectronics

Chapter 39

True Random Number Generator (TRNG)

39.1 Introduction

The standalone true random number generator is a hardware accelerator module that generates a 256-bit entropy as needed by an entropy-consuming module or by other post-processing functions.

39.1.1 Features

- Generate a 256-bit entropy
- Monobit test to limit the number of value 0 and 1
- Long run test to avoid continuous constant value
- 1 ring OSC with clock checker
- 3 interrupt sources
  - Entropy valid interrupt
  - Frequency error interrupt
  - Hardware error interrupt

Yuntu Microelectronics

39.1.2 Block diagram

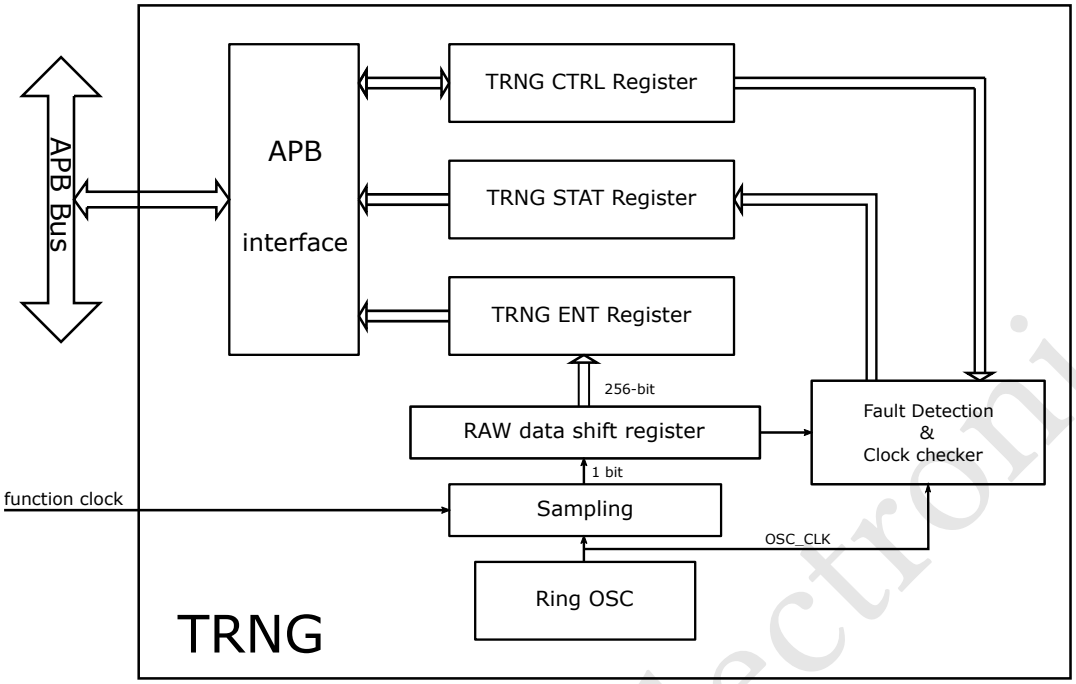


Figure 39.1: TRNG Block Diagram

39.2 Register Definition

This subsection provides a description of the memory map and registers.

39.2.1 TRNG Memory Map

TRNG base address 40198000h

Table 39.1: TRNG Memory Map

Offset	Register	Width	Access	Reset Value
0h	Control Register (CTRL)	32	RW	01220020h
4h	Status Register (STAT)	32	RW	00000000h
8h	Seed Control Register (SDCTL)	32	RW	0C8009C4h
ch	Frequency Minimum Limit Register (FRQMIN)	32	RW	00000640h
10h	Frequency Maximum Limit Register (FRQMAX)	32	RW	00006400h
14h	Frequency Count Register (FRQCNT)	32	RW	00000000h
18h	Statistical Check Monobit Limit Register (SCML)	32	RW	010C0568h
1ch	Statistical Check Monobit Count Register (SCMC)	32	RW	00000000h
40h	Entropy Read Register (ENT0)	32	RW	00000000h
44h	Entropy Read Register (ENT1)	32	RW	00000000h
48h	Entropy Read Register (ENT2)	32	RW	00000000h
4ch	Entropy Read Register (ENT3)	32	RW	00000000h
50h	Entropy Read Register (ENT4)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
54h	Entropy Read Register (ENT5)	32	RW	00000000h
58h	Entropy Read Register (ENT6)	32	RW	00000000h
5ch	Entropy Read Register (ENT7)	32	RW	00000000h

39.2.1.1 TRNG CTRL Register

Offset: 0h

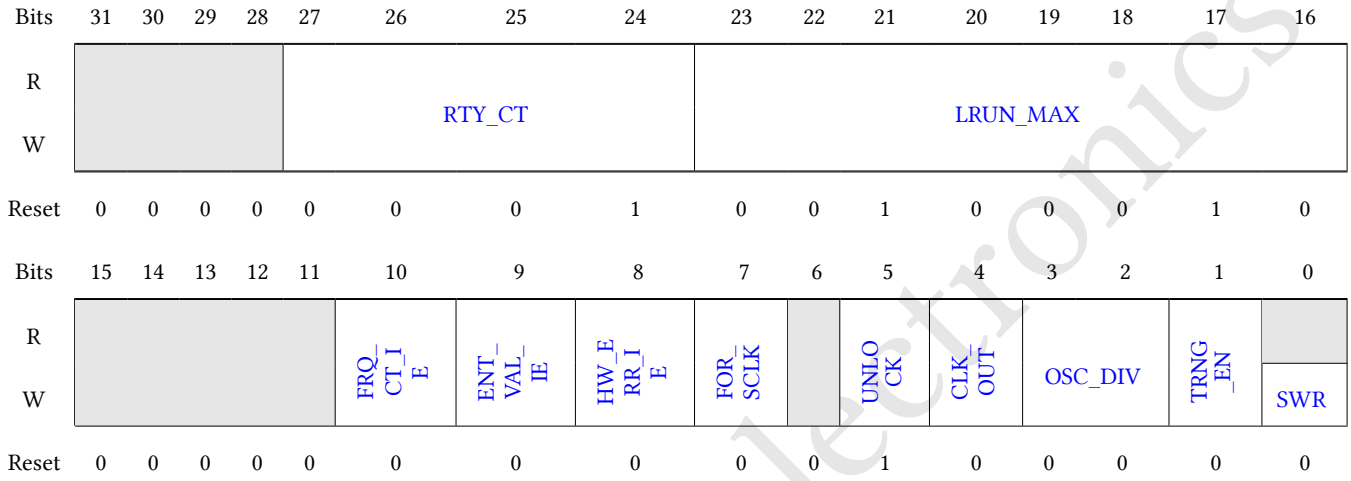


Table 39.2: TRNG CTRL Register Description

Field	Function
27 - 24 RTY_CT	**Retry Count.** It indicates the number of times a retry should occur before generating an error. Retry occurs when the statistical check fails during the TRNG entropy generation. This bit is writable only if UNLOCK bit is 1, or the UNLOCK bit is being written to 1 simultaneously.
23 - 16 LRUN_MAX	**Long Run Max Limit** It is the largest allowable number of consecutive samples of all 1 or all 0 during the entropy generation. This bit is writable only if UNLOCK bit is 1, or the UNLOCK bit is being written to 1 simultaneously.
10 FRQ_CT_IE	**Frequency Count Fail Interrupt Enable** 0b - Interrupt disable. 1b - Interrupt enable.
9 ENT_VAL_IE	**Entropy Valid Interrupt Enable** 0b - Interrupt disable. 1b - Interrupt enable.
8	**Hardware Error Interrupt Enable**

Yuntu Microelectronics

Field	Function
HW_ERR_IE	0b - Interrupt disable. 1b - Interrupt enable.
7 FOR_SCLK	**Force System Clock.** If the bit is set, the system clock is used to operate the TRNG instead of the ring OSC. This bit is writable only if UNLOCK bit is 1, or the UNLOCK bit is being written to 1 simultaneously. 0b - No effect. 1b - Force the system clock to operate the TRNG instead of the ring OSC.
5 UNLOCK	**Configuration Unlock Bit** This bit can prevent the value of various control bits from changing due to unexpected operation. 0b - Lock. Write to the control bits has no effect. 1b - Unlock. The control bits are able to be modified.
4 CLK_OUT	**TRNG OSC Clock Output.** After divided by 10, trng_osc_clk is output for test. 0b - No effect. 1b - trng_osc_clk is output for test.
3 - 2 OSC_DIV	**OSC Divider** This field is writable only if UNLOCK bit is 1, or the UNLOCK bit is being written to 1 simultaneously. 00b - No devider. 01b - Ring OSC divided by 2. 10b - Ring OSC divided by 4. 11b - Ring OSC divided by 8.
1 TRNG_EN	**TRNG Enable** Enable the TRNG to generate and entropy value that can be read via ENT0-ENT7 registers. 0b - TRNG disable. 1b - TRNG enable.
0 SWR	**TRNG Software Reset** Writing 1 to this bit clears various TRNG registers(RTY_CT, LRUN_MAX, FOR_SCLK, CLK_OUT, OSC_DIV, SAMP_SIZE and SCML) to their default state. This bit is writable only if UNLOCK bit is 1, or the UNLOCK bit is being written to 1 simultaneously. 0b - No effect. 1b - Software reset.

39.2.1.2 TRNG STAT Register

Offset: 4h

Yuntu Microelectronics

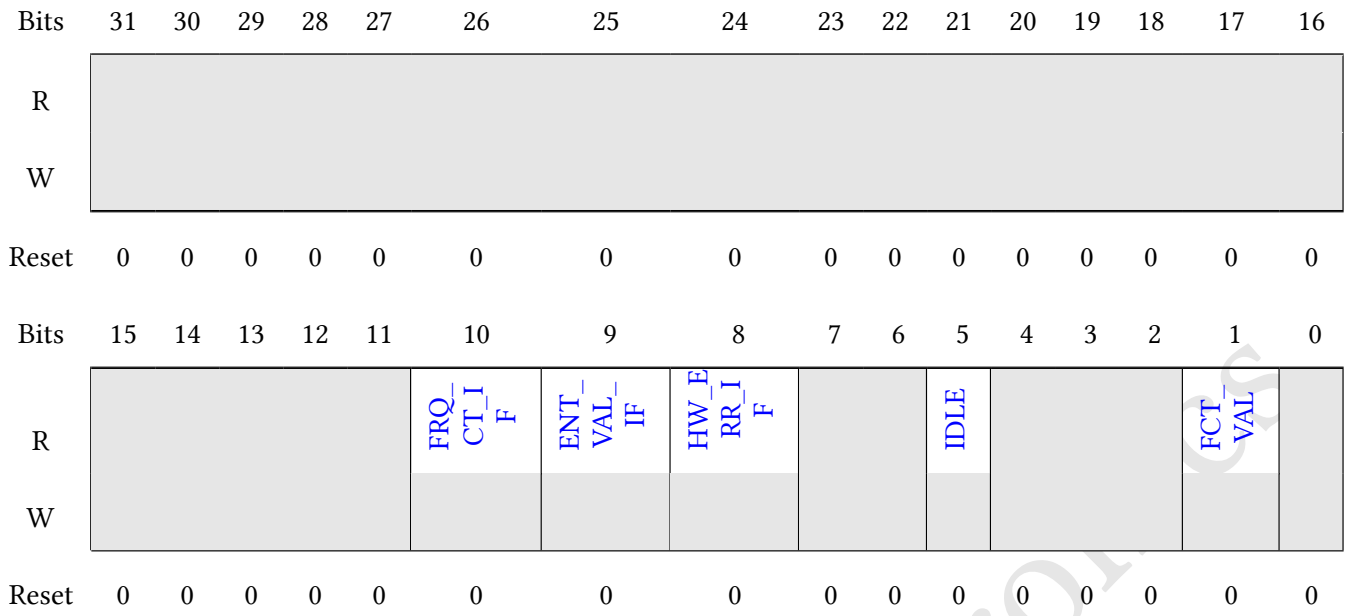


Table 39.3: TRNG STAT Register Description

Field	Function
10 FRQ_CT_IF	**Frequency Count Fail Interrupt Flag** It indicate the frequency counter has detected a failure. The FRQMAX and FRQMIN registers has effect on it. 0b - Frequency count success. 1b - Frequency count fail.
9 ENT_VAL_IF	**Entropy Valid Interrupt Flag.** When entropy generation succeed, the TRNG will generate an entropy value that can be read via ENT0-ENT7 registers and assert this bit. 0b - No valid entropy value 1b - Valid entropy value can be read.
8 HW_ERR_IF	**Hardware Error Interrupt Flag** 0b - No hardware error. 1b - Hardware error happens.
5 IDLE	**IDLE** The status of the entropy generation FSM. 0b - State idle. 1b - State busy.
1 FCT_VAL	**Frequency Count Valid.** The frequency count for trng_osc_clk is valid to read. 0b - The frequency count for trng_osc_clk is not ready for read. 1b - The frequency count for trng_osc_clk is ready to read.

Yuntu Microelectronics

39.2.1.3 TRNG SDCTL Register

Offset: 8h

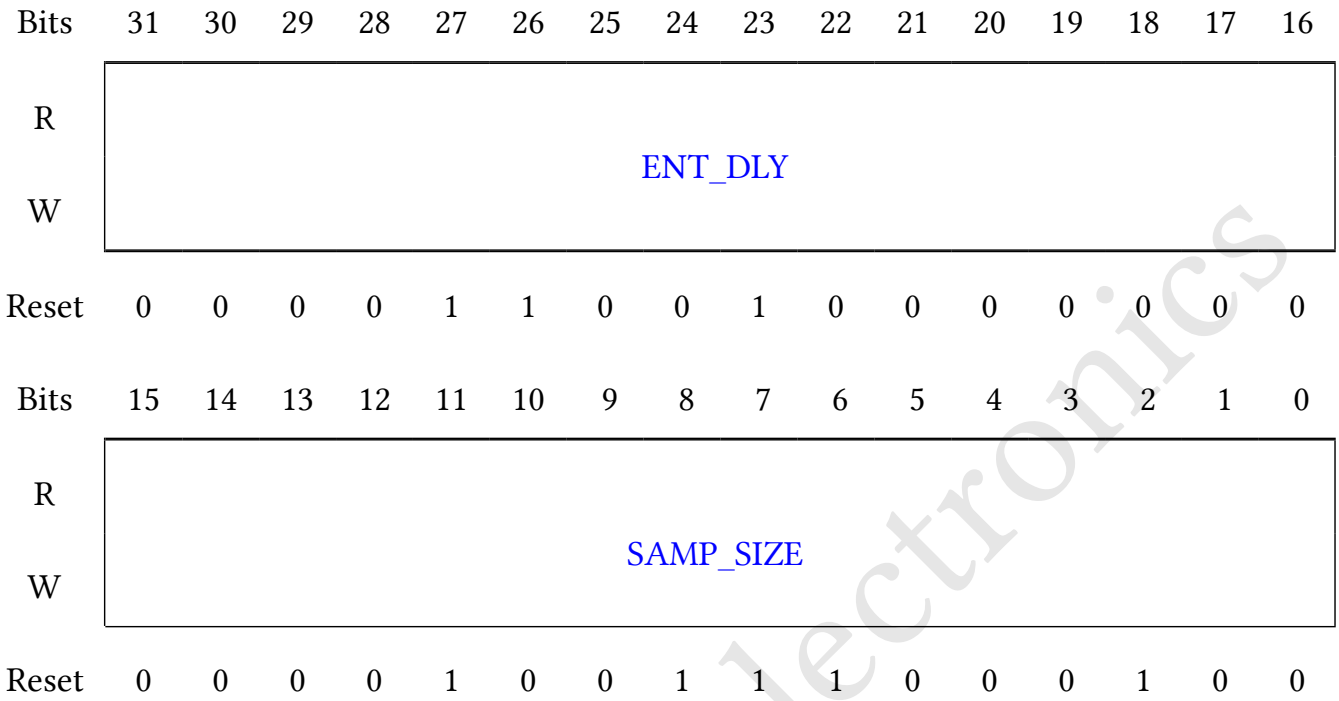


Table 39.4: TRNG SDCTL Register Description

Field	Function
31 - 16 ENT_DLY	**Entropy Delay** Defines the length of each entropy sample taken. This bit is writable only if UNLOCK bit is 1.
15 - 0 SAMP_SIZE	**Sample Size** Defines the total number of entropy samples taken during entropy generation. This bit is writable only if UNLOCK bit is 1.

39.2.1.4 TRNG FRQMIN Register

Offset: ch

Yuntu Microelectronics

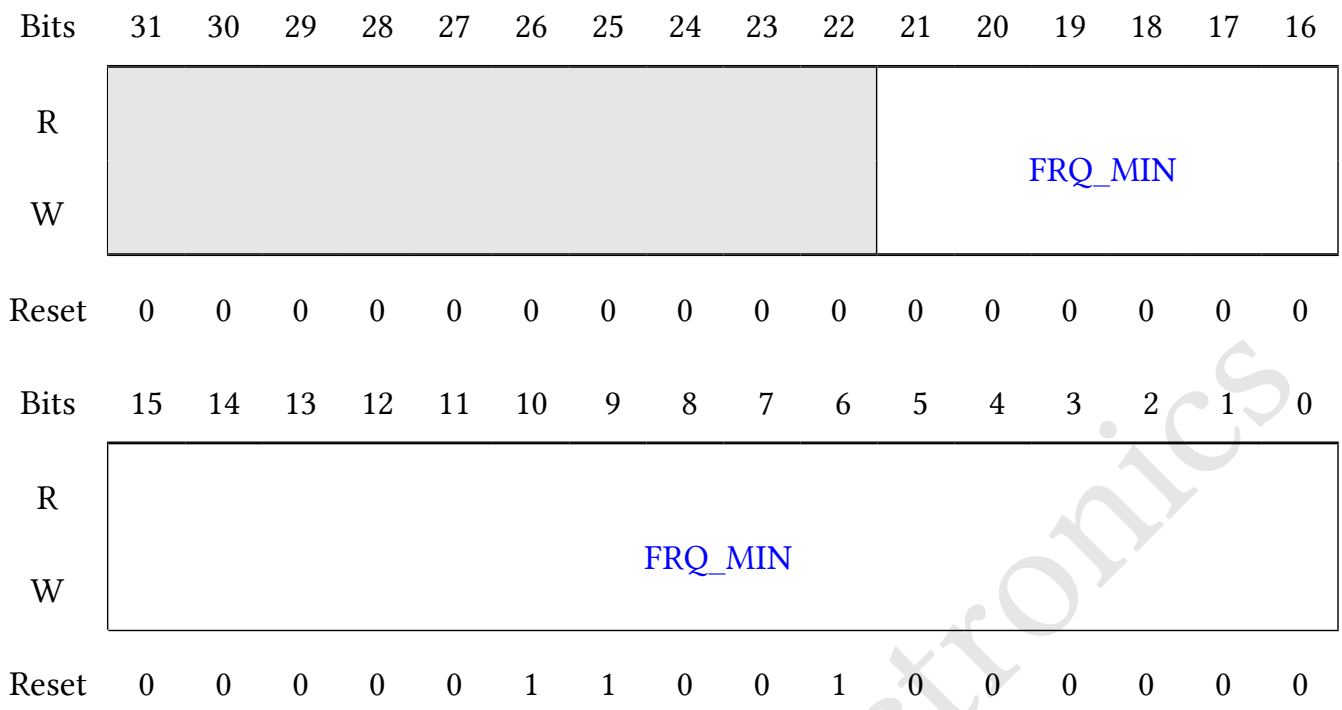


Table 39.5: TRNG FRQMIN Register Description

Field	Function
21 - 0 FRQ_MIN	**Frequency minimum limit value** Define the minimum allowable count taken during each entropy sample. This bit is writable only if UNLOCK bit is 1.

Yuntu Microelectronics

39.2.1.5 TRNG FRQMAX Register

Offset: 10h

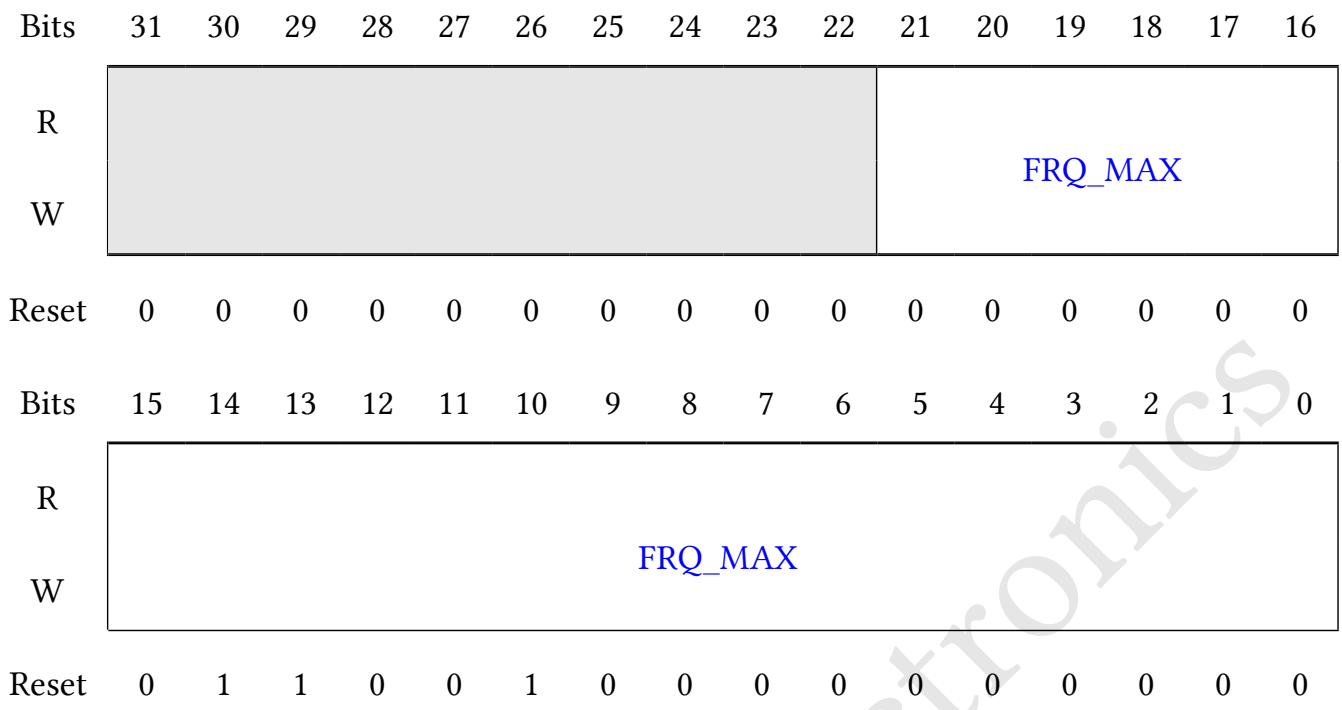


Table 39.6: TRNG FRQMAX Register Description

Field	Function
21 - 0 FRQ_MAX	**Frequency Maximum Limit Value** Define the maximum allowable count taken during each entropy sample. This bit is writable only if UNLOCK bit is 1.

Yuntu Microelectronics

39.2.1.6 TRNG FRQCNT Register

Offset: 14h

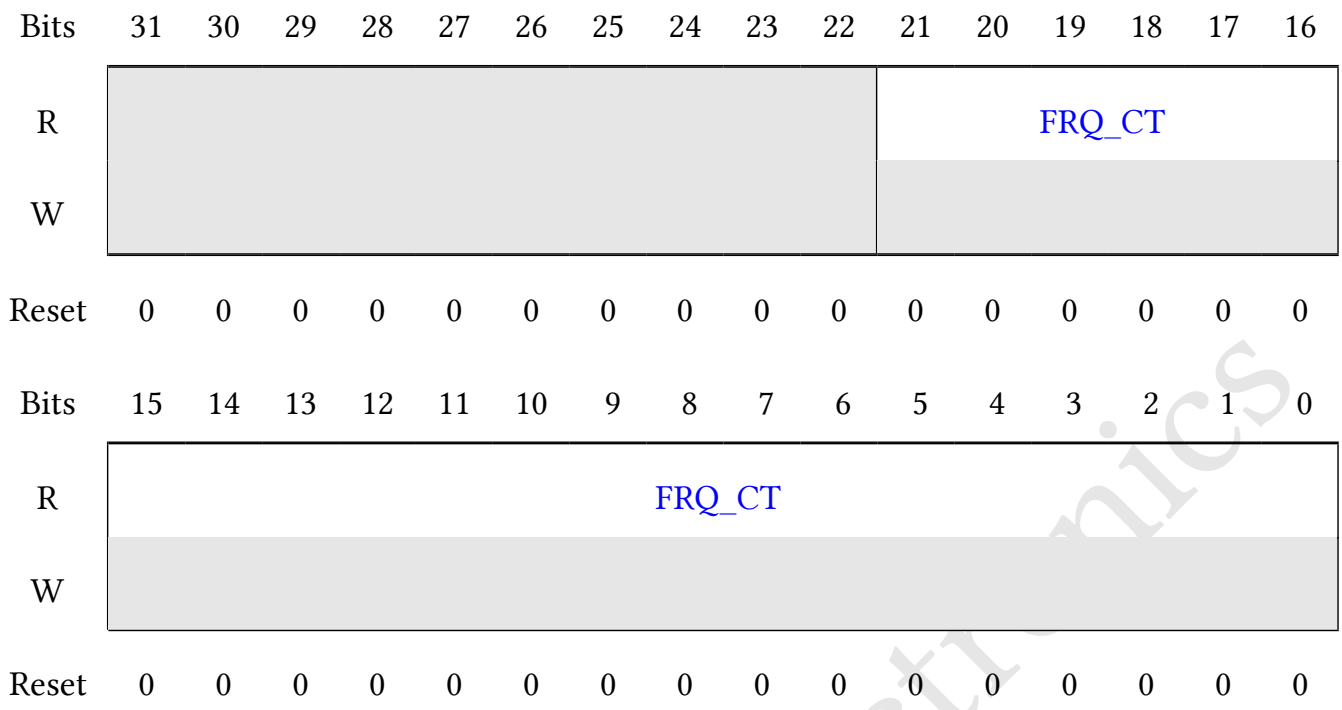


Table 39.7: TRNG FRQCNT Register Description

Field	Function
21 - 0 FRQ_CT	**Frequency Count Value** When STAT[ENT_VAL_IF]=1, reads a sample frequency count takes during entropy generation.

Yuntu Microelectronics

39.2.1.7 TRNG SCML Register

Offset: 18h

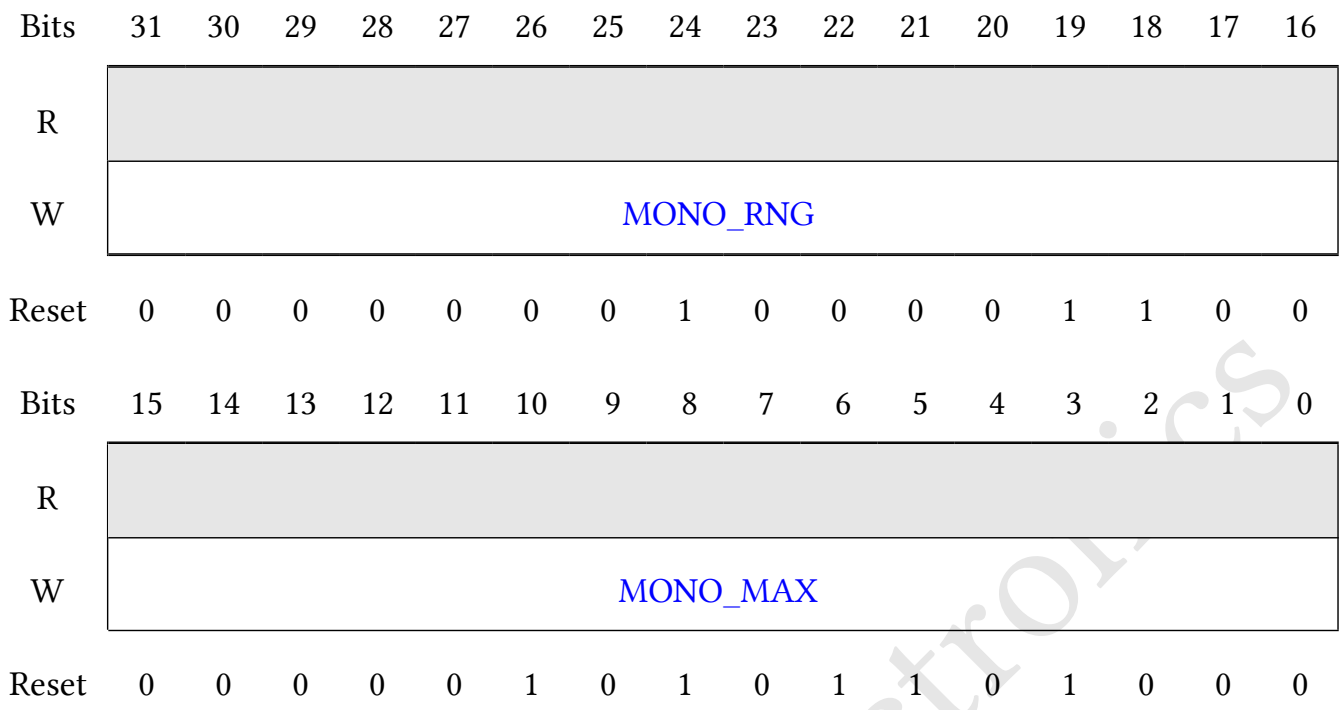


Table 39.8: TRNG SCML Register Description

Field	Function
31 - 16 MONO_RNG	**Monobit Range** The number of ones detected during entropy generation must be greater than MONO_MAX - MONO_RNG, otherwise a retry or error will occur. This field is writable only if UNLOCK bit is 1.
15 - 0 MONO_MAX	**Monobit Maximum Limit** The number of ones detected during entropy generation must be less than MONO_MAX, otherwise a retry or error will occur. This field is writable only if UNLOCK bit is 1.

Yuntu Microelectronics

39.2.1.8 TRNG SCMC Register

Offset: 1ch

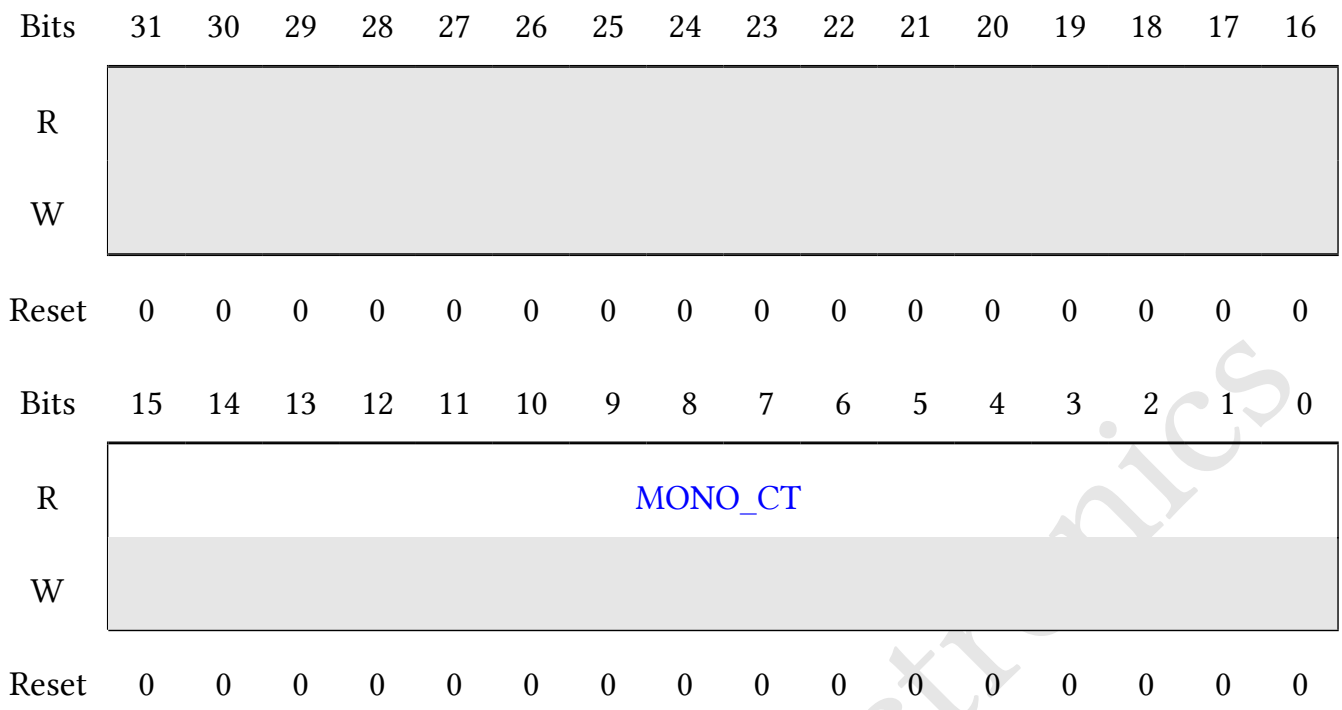


Table 39.9: TRNG SCMC Register Description

Field	Function
15 - 0 MONO_CT	**Monobit Count** Reads the final monobit count after entropy generation. Start from SCML[MONO_MAX], and is decremented each time a ”1” is sampled.

Yuntu Microelectronics

39.2.1.9 TRNG ENT Register

Offset: 40h + n * 4h

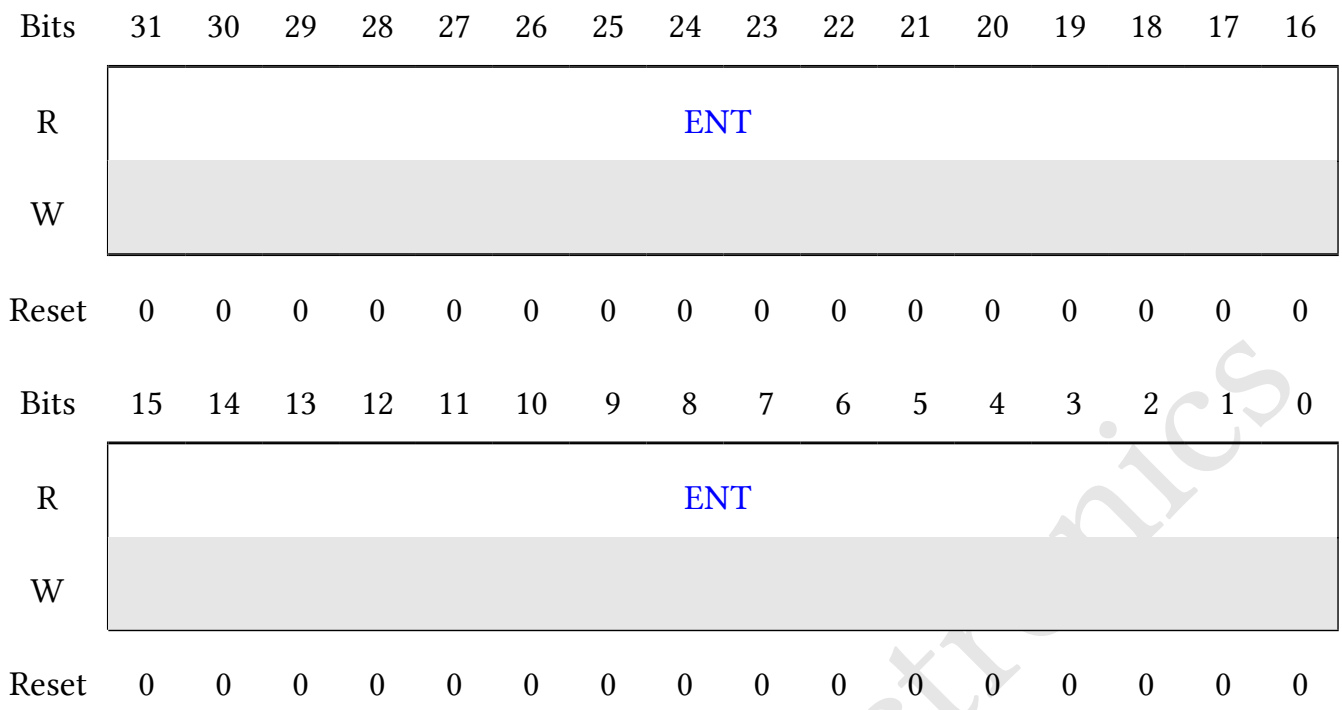


Table 39.10: TRNG ENT Register Description

Field	Function
31 - 0 ENT	**Entropy Value** The entropy value will be valid when STAT[ENT_VAL_IF] = 1. Reading the highest offset(ENT(7)) will clear the entire entropy value and start a new entropy generation.

39.3 Functional description

The TRNG consists of a ring oscillator, a random bits collector, a clock monitor and a fault detector.

39.3.1 Ring Oscillator

The ring oscillator is a random noise source. It is sensitive to the environment noise such as temperature, voltage, electromagnetic interference and other noise sources. There noise cause small, continuous and random changes on the output clock frequency of the ring oscillator. The various changes for each oscillator clock cycle is very small, but the count of the ring oscillator clock cycles which sampled after a certain cycles of the functional clock vary each time.

39.3.2 Random Bits Collector

Random bits collector consists of a timer and a counter. There timer is configured by SDCTL[ENT_DLY] and clocked by the functional clock. The counter resets and starts counting when the timer is enabled and the bit0 of the counter will be sampled when the timer reaches SDCTL[ENT_DLY]. The sampled bit will be shifted to the random data shift register.

Yuntu Microelectronics

39.3.3 Clock Monitor

The TRNG has a clock monitor for the ring oscillator. The frequency of the ring oscillator is able to read by FRECNT when STAS[ENT_VAL_IF] is valid. The maximum and minimum frequency of the ring oscillator is able to be limited by FRQMAX and FRQMIN. If the clock frequency does not meet the requirements, STAS[FRQ_CT_IF] will assert.

39.3.4 Fault Detector

The fault detector offers several statistical tests to generate entropy.

39.3.4.1 Long Run Detector

The CTRL[LRUN_MAX] gives the limit for the largest allowed number of consecutive samples of all 1 or all 0 during the entropy generation. If long run is detected, the collected data will be discarded and a new entropy generation starts.

39.3.4.2 Monobit Limitation

The SCML[MONO_RNG] and SCML[MONO_MAX] gives the limit for the ranges of the number of ones/zeros detected during entropy generation. If monobit test fails, the collected data will be discarded and a new entropy generation starts.

39.3.4.3 Retry Count Limitation

The CTRL[RTY_CT] indicates the number of times retry should occur before generating an error STAT[HW_ERR_IF]. Retry occurs when the statistical check fails during the entropy generation.

39.4 Glossary

TRNG True Random Number Generator

Yuntu Microelectronics

Chapter 40

Watch Dog (WDG)

40.1 Introduction

The Watchdog Timer(WDG) modules is an independent timer for safety monitor during software execution. It is a 32-bit timer that enables system to recover from unexpected situations like software stuck at a loop or executing unintended code. User can configure WDG to generate interrupt or reset for the first timeout, and it always generates reset request for the second consecutive timeout.

40.1.1 Features

- 32-bit countdown timer
- Functional clock can be selected from IPC and bus clock
- Support regular or window servicing mode
- Support reset request or interrupt for the first timeout
- Hard and soft configuration lock bits

40.2 Register Definition

This subsection provides a description of the memory map and registers.

40.2.1 WDG Memory Map

WDG0 base address 401a8000h

Table 40.1: WDG Memory Map

Offset	Register	Width	Access	Reset Value
0h	Service Code Register (SVCR)	32	WO	00000000h
4h	Control Register (CR)	32	RW	00000082h
8h	Lock Register (LR)	32	RW	00000001h
ch	Timer Overflow Value (TOVR)	32	RW	00000300h
10h	Window Value Register (WVR)	32	RW	00000000h
14h	Counter Current Value Register (CNTCVR)	32	RO	00000000h
18h	Interrupt Flag register (INTF)	32	RW	00000000h

Yuntu Microelectronics

40.2.1.1 WDG SVCR Register

Watchdog Service Code Register

Offset: 0h

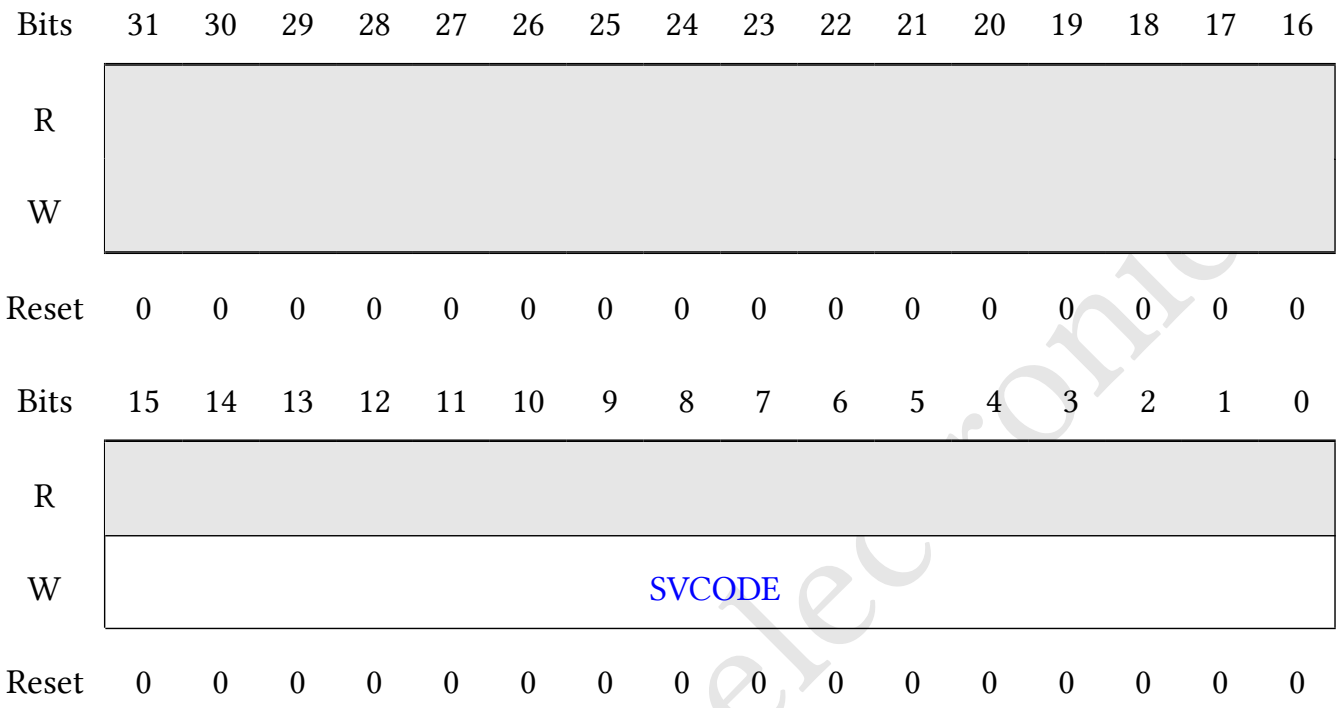


Table 40.2: WDG SVCR Register Description

Field	Function
15 - 0 SVCODE	**Watchdog Service Code** It can be used to service the watchdog or to clear the soft lock bit. Writing 0xB631 followed by 0xC278 will clear the soft lock bit, writing 0xA518 followed by 0xD826 to service the watchdog.

40.2.1.2 WDG CR Register

Watchdog Control Register

Offset: 4h

Yuntu Microelectronics

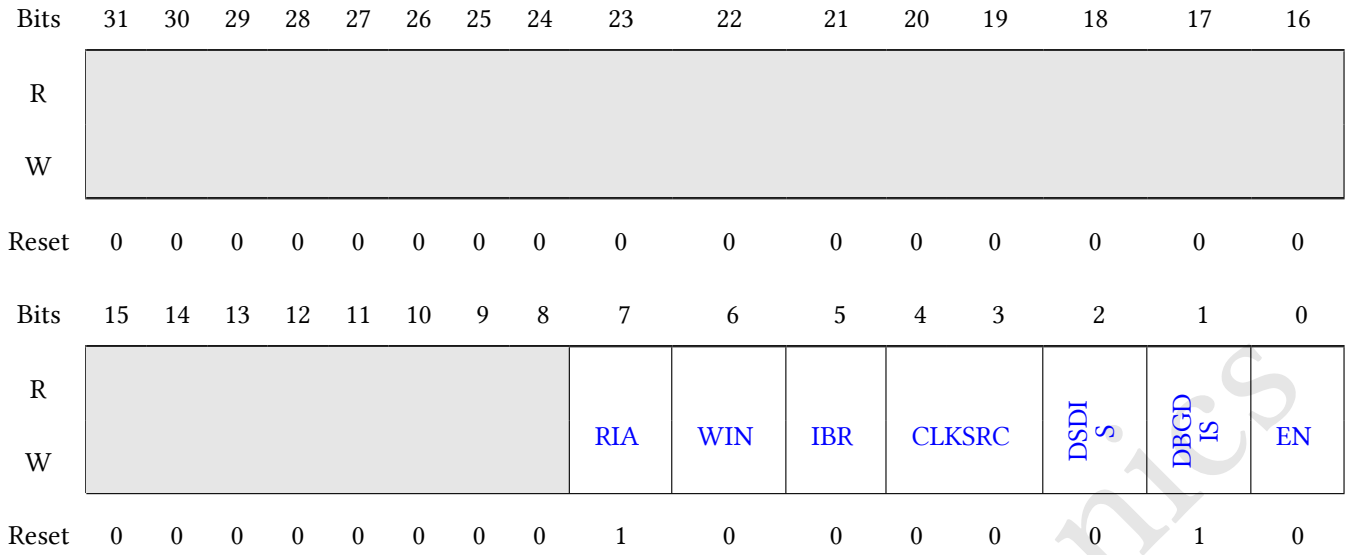


Table 40.3: WDG CR Register Description

Field	Function
7 RIA	**Reset Invalid Apb Enable** Enable reset for invalid APB access. 0b - Disable to generate reset by invalid APB access 1b - Enable to generate reset by invalid APB access
6 WIN	**Window Mode** Enable window mode. 0b - Disable window mode, watchdog service can be done at any time. 1b - Enable window mode, watchdog service is can only be done when timer value is less than the value of WIN register.
5 IBR	**Interrupt Before Reset** Enable interrupt before reset function 0b - Disable interrupt before reset. 1b - Enable interrupt before reset, when watchdog time-out, generate an interrupt first and when next time-out happens, generate reset.
4 - 3 CLKSRC	**Clock Source** 00b - Functional clock from IPC 01b - Bus clock 10b - Reserved 11b - Reserved
2 DSDIS	**Enable In Deepsleep Mode** Enable or disable watchdog in deepsleep mode 0b - Enable watchdog in deepsleep mode. 1b - Disable watchdog in deepsleep mode.
1	**Disable In Debug Mode**

Yuntu Microelectronics

Field	Function
DBGDIS	Enable or disable watchdog in debug mode. 0b - Enable watchdog in debug mode. 1b - Disable watchdog in debug mode.
0 EN	**Module Enable** Enable watchdog module. 0b - Disable watchdog module 1b - Enable watchdog module

40.2.1.3 WDG LR Register

Watchdog Lock Register

Offset: 8h

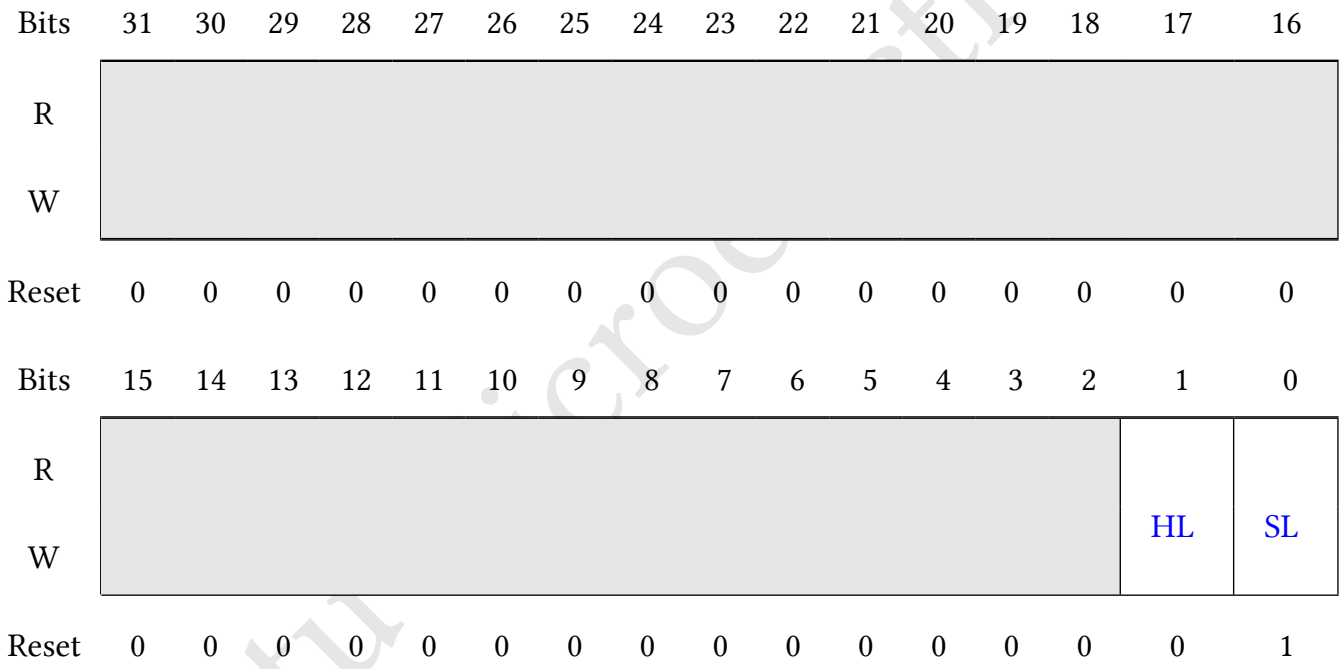


Table 40.4: WDG LR Register Description

Field	Function
1 HL	**Hard Lock** Hard lock, when this bit set, the CR, WIN, TOVR are read-only. once set, the bit can only be clear by reset.
0 SL	**Soft Lock** Soft lock, when this bit set, the CR, WIN, TOVR are read-only. this bit can be clear by writing a valid sequence to SVCR register.

Yuntu Microelectronics

40.2.1.4 WDG TOVR Register

Watchdog Timer Overflow Value

Offset: ch

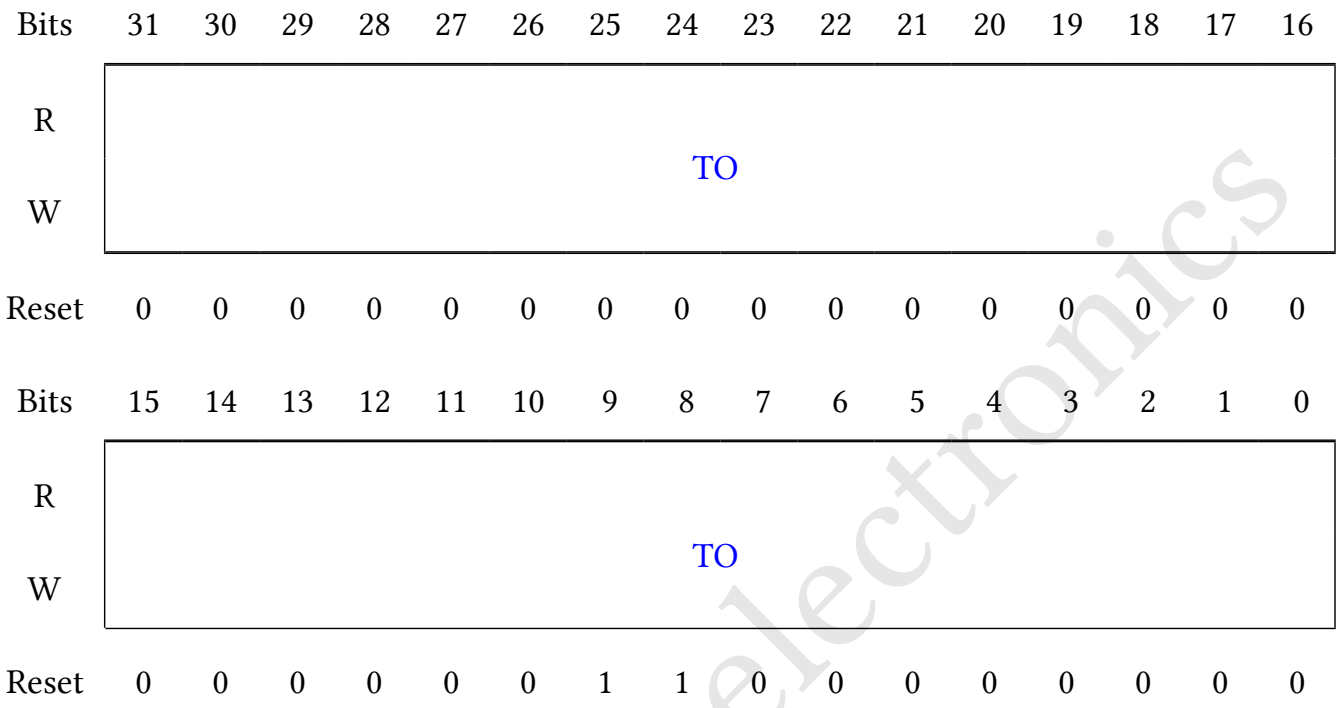


Table 40.5: WDG TOVR Register Description

Field	Function
31 - 0 TO	**Timer Overflow Value** A 32bit down counter uses this value to start count. Writing value should be greater than 0x100. Otherwise 0x100 is written to this register.

Yuntu Microelectronics

40.2.1.5 WDG WVR Register

Offset: 10h

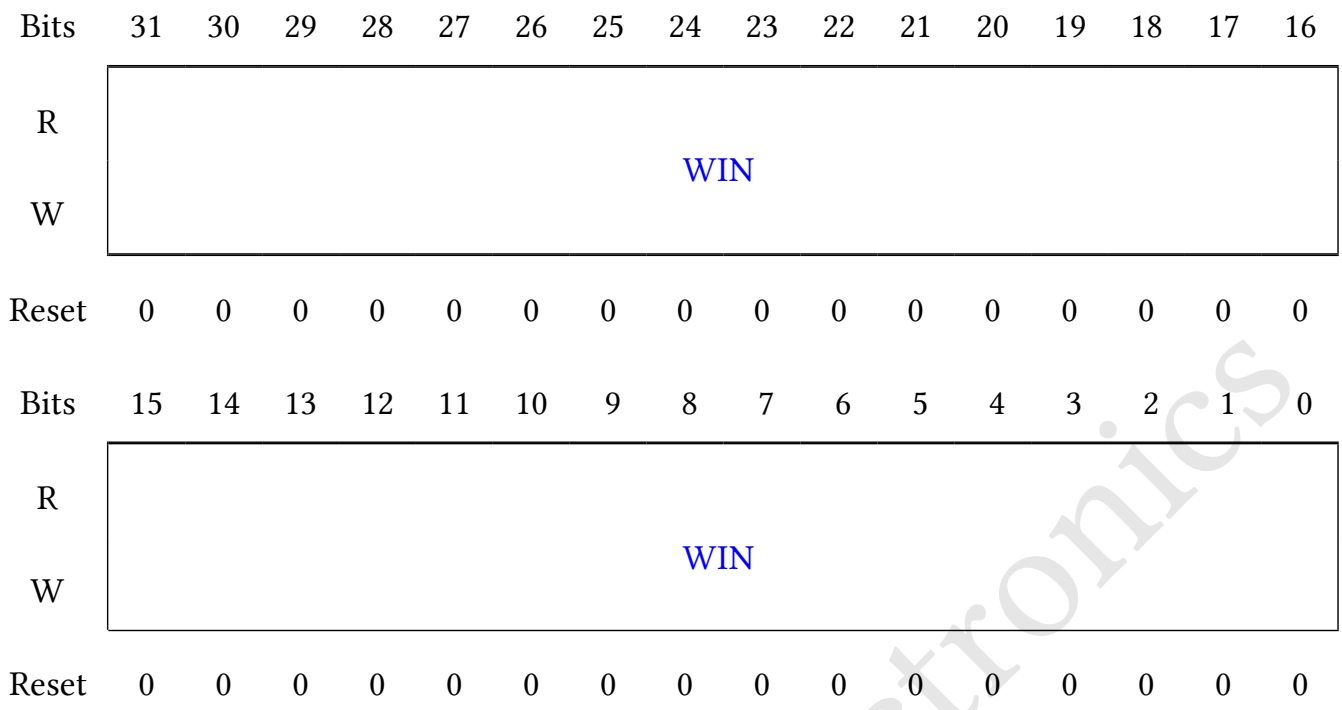


Table 40.6: WDG WVR Register Description

Field	Function
31 - 0 WIN	**Watchdog Window Value** When CR[WIN] is set, the window mode is enabled, the service sequence can only be written when the timer value is less than the value of this register.

Yuntu Microelectronics

40.2.1.6 WDG CNTCVR Register

Watchdog Counter Current Value Register

Offset: 14h

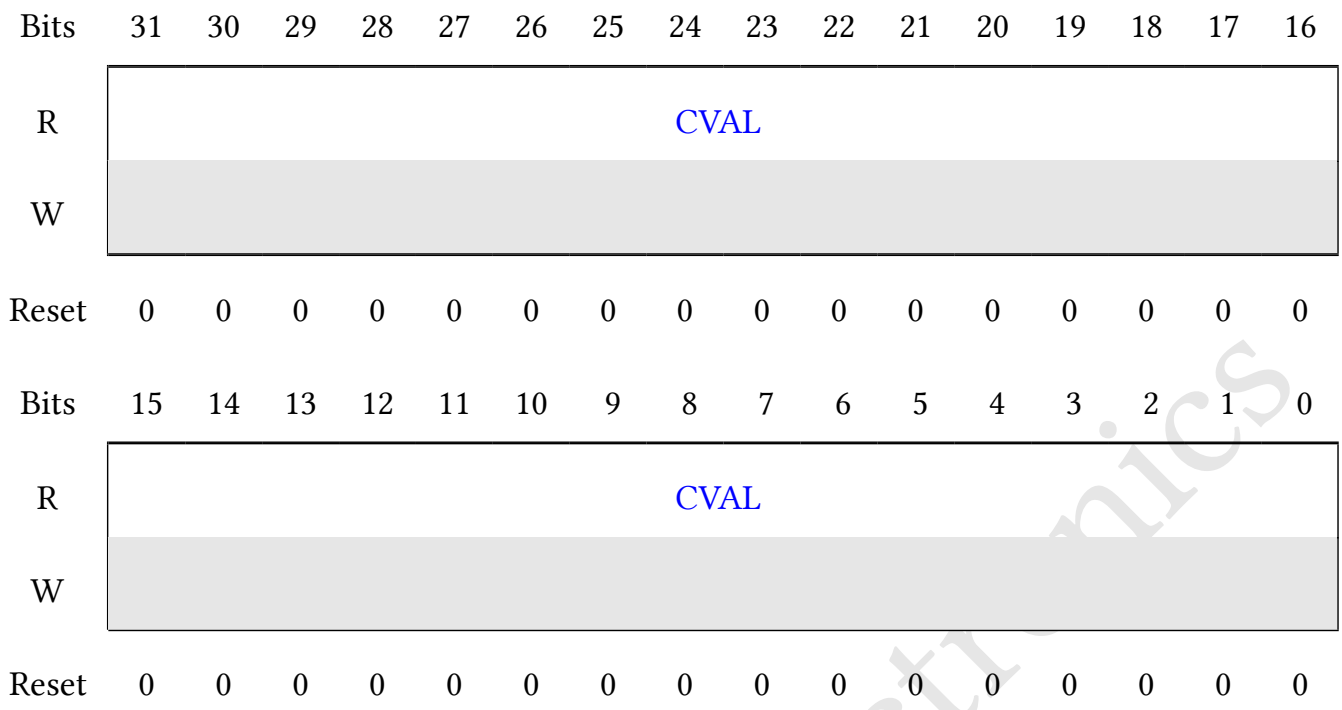


Table 40.7: WDG CNTCVR Register Description

Field	Function
31 - 0 CVAL	**Counter Current Value** Watchdog counter current value.

40.2.1.7 WDG INTF Register

Watchdog Interrupt Flag register

Offset: 18h

Yuntu Microelectronics

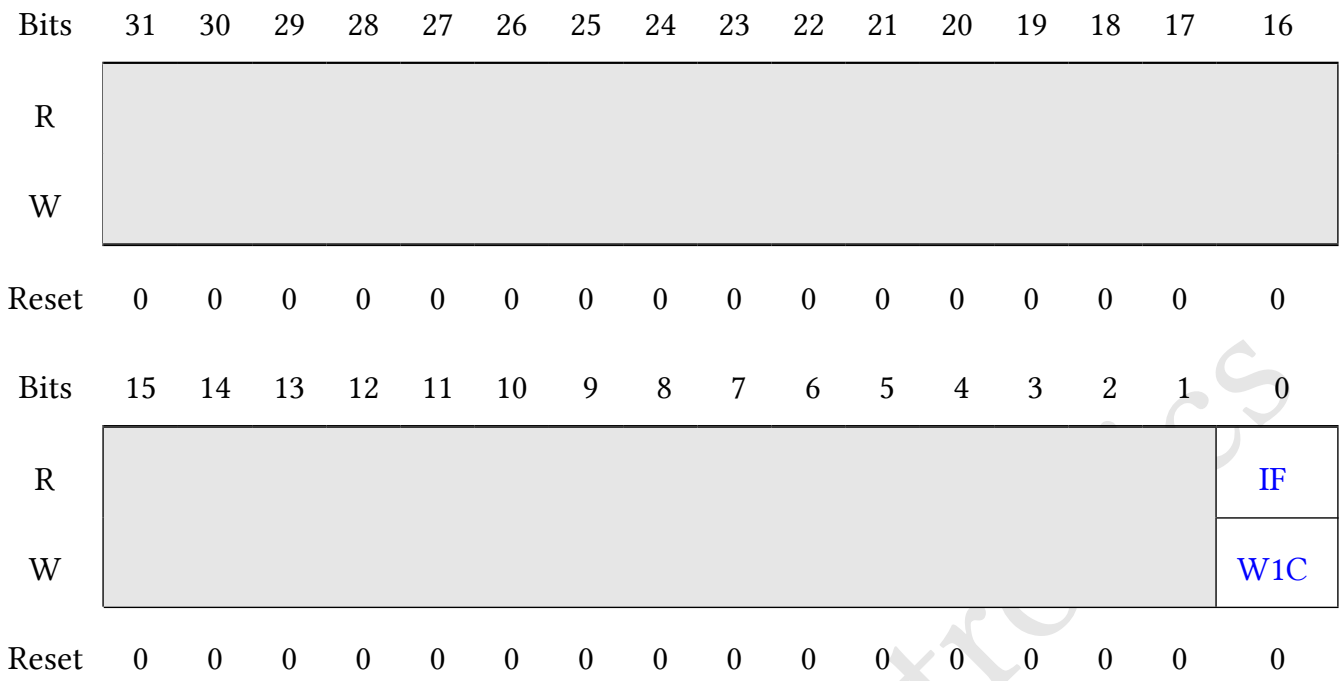


Table 40.8: WDG INTF Register Description

Field	Function
0 IF	**Interrupt Flag** Watchdog interrupt flag. Write 1 to this field to clear the flag and interrupt. 0b - No watchdog interrupt is pending 1b - Watchdog timer time-out interrupt is pending

40.3 Functional Description

40.3.1 General

This section gives detailed information on the internal operation of the module. The WDG is a 32-bit window watchdog that enables the system to recover from situations such as:

- Software trapped in a loop
- A bus transaction failing to terminate

In regular operation, WDG requires periodic execution of a watchdog servicing operation. The servicing operation resets the timer to a specified timeout period. If this servicing action does not occur before the timer expires, WDG generates an interrupt or a hardware reset request.

You can configure WDG to generate a reset request or an interrupt on the initial timeout. WDG generates a reset request if a second consecutive timeout occurs before writing the service sequence for the first timeout.

40.3.1.1 Operation Modes

WDG supports the chip modes of operation as follows:

Yuntu Microelectronics

- Normal(Active): When WDG is enabled(CR[EN]=1), the WDG timer counts down continuously.
- Debug: If CR[DBGDIS] = 1, WDG stops the timer; otherwise, the timer continues to run.
- Sleep: Timer continues to run when function clock is valid.
- DeepSleep: Timer continues to run when function clock is valid.
- Standby: Timer continues to run when function clock is valid.
- Powerdown: WDG is powered down and is stopped to operate. When out of Powerdown, WDG behaviors like out of POR.

As for how to enable the function clock in corresponding power mode, please see Power modes table for more information.

40.3.2 Interrupts

WDG module just supports one interrupt. Please refer to the table of interrupt vectors. WDG interrupt can be enabled by setting the CR[IBR] bit to 1. The WDG interrupt flag(INTF[IF]) is set to 1 while timeout occurs, and the interrupt flag can be cleared by setting INTF[IF] to 1.

Yuntu Microelectronics

Chapter 41

ECC Management Unit (EMU)

41.1 Chip-specific EMU Information

ECC Management Unit(EMU) have two main functions: Error Injection and Error Report.

There are six channels that could be injected and/or reported.

Table 41.1: EMU Channel Table

Channel No.	Memory	Data width	Check-bit width
0	OCRAML	64	8
1	OCRAMU	64	8
2	ITCM	64	8
3	DTCM0	32	7
4	DTCM1	32	7
5	ROM	64	8

41.2 Introduction

The ECC Management Unit (EMU) provides ECC error injection and ECC error report function.

41.2.1 Overview

41.2.1.1 Error Injection

Error injection is mainly used for diagnostic purposes. It provides a method for diagnostic coverage of on-chip memories. Error injection uses two-stage enable mechanism to protect against accidental enabling and reconfiguration of the error injection function of each channel.

Error injection provides support for inducing single-bit or multi-bit inversions on read data when accessing on-chip memories.

NOTE: more than two bits inversions may result in undefined behavior.

41.2.1.2 Error Report

Error report provides information and optionally interrupt notification on on-chip memories error events associated with ECC (Error Correction Code) and parity.

Yuntu Microelectronics

Error report provides count registers which count all correctable error so far for each channel.

41.2.2 Features

- Multiple channels of ECC injection and report
- Two-stage enable mechanism for ECC injection
- Location and correctable or uncorrectable error can be injected
- ECC error interrupt can be enabled with separated register bit
- The last ECC error data can be recorded into register

41.3 Register Definition

41.3.1 Programming Model Access

The EMU provides a programming model mapped to an on-platform peripheral slot.

- All system bus masters can access the programming model:
  - Only in supervisor mode
  - Using only 32-bit (word) access
- Any of the following attempted references to the programming model generates an bus error termination:
  - In user mode
  - Using non-32-bit access sizes
  - To undefined (reserved) addresses

41.3.2 Error Injection Registers

- Error injection have two-stage enable mechanism: EIGCR[GEN] and EICHCR[CHnEN].
- Error injection have six array registers for OCRAML, OCRAMU, ITCM, DTCM0, DTCM1 and ROM.
- Specifies an address that defines error will be injected to all read access or an read access(EICHDn_ADDR).
- Specifies a checkbit mask that defines which bits of the checkbit bus from target on-chip memories are inverted on a read access.
- Specifies a data mask that defines which bits of the data bus from target on-chip memories are inverted on a read access.

41.3.3 Error Report Registers

- Specifies a register for error status and a register for error interrupt enable.
- Error report have six array registers for OCRAML, OCRAMU, ITCM, DTCM0, DTCM1 and ROM.
- Error report will keep the last recorded ECC event(ADDR, SYND).
- Error report will count the correctable error so far.

41.3.4 EMU Memory Map

EMU0 base address 401c8000h

Yuntu Microelectronics

Table 41.2: EMU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Error Injection Global Control Register (EIGCR)	32	RW	00000000h
4h	Error Injection Channel Control Register (EICHCR)	32	RW	00000000h
10h	Channel Error Injection Address Register (EICHD0_ADDR)	32	RW	00000000h
14h	Error Injection CheckBit Mask Register (EICHD0_CHKBIT)	32	RW	00000000h
18h	Error Injection DATAL Mask Register (EICHD0_DATAL)	32	RW	00000000h
1ch	Error Injection DATAH Mask Register (EICHD0_DATAH)	32	RW	00000000h
20h	Channel Error Injection Address Register (EICHD1_ADDR)	32	RW	00000000h
24h	Error Injection CheckBit Mask Register (EICHD1_CHKBIT)	32	RW	00000000h
28h	Error Injection DATAL Mask Register (EICHD1_DATAL)	32	RW	00000000h
2ch	Error Injection DATAH Mask Register (EICHD1_DATAH)	32	RW	00000000h
30h	Channel Error Injection Address Register (EICHD2_ADDR)	32	RW	00000000h
34h	Error Injection CheckBit Mask Register (EICHD2_CHKBIT)	32	RW	00000000h
38h	Error Injection DATAL Mask Register (EICHD2_DATAL)	32	RW	00000000h
3ch	Error Injection DATAH Mask Register (EICHD2_DATAH)	32	RW	00000000h
40h	Channel Error Injection Address Register (EICHD3_ADDR)	32	RW	00000000h
44h	Error Injection CheckBit Mask Register (EICHD3_CHKBIT)	32	RW	00000000h
48h	Error Injection DATAL Mask Register (EICHD3_DATAL)	32	RW	00000000h
4ch	Error Injection DATAH Mask Register (EICHD3_DATAH)	32	RW	00000000h
50h	Channel Error Injection Address Register (EICHD4_ADDR)	32	RW	00000000h
54h	Error Injection CheckBit Mask Register (EICHD4_CHKBIT)	32	RW	00000000h
58h	Error Injection DATAL Mask Register (EICHD4_DATAL)	32	RW	00000000h
5ch	Error Injection DATAH Mask Register (EICHD4_DATAH)	32	RW	00000000h
60h	Channel Error Injection Address Register (EICHD5_ADDR)	32	RW	00000000h
64h	Error Injection CheckBit Mask Register (EICHD5_CHKBIT)	32	RW	00000000h
68h	Error Injection DATAL Mask Register (EICHD5_DATAL)	32	RW	00000000h
6ch	Error Injection DATAH Mask Register (EICHD5_DATAH)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
100h	Error Report Interrupt Enable Register (ERIE)	32	RW	00000000h
104h	Error Report Interrupt Flag Register (ERIF)	32	RW	00000000h
110h	Error Report Address Register (ERINFO0_ADDR)	32	RO	00000000h
114h	Error Report Syndrome Register (ERINFO0_SYND)	32	RO	00000000h
118h	Correctable Error Count Register (ERINFO0_CNT)	32	RW	00000000h
120h	Error Report Address Register (ERINFO1_ADDR)	32	RO	00000000h
124h	Error Report Syndrome Register (ERINFO1_SYND)	32	RO	00000000h
128h	Correctable Error Count Register (ERINFO1_CNT)	32	RW	00000000h
130h	Error Report Address Register (ERINFO2_ADDR)	32	RO	00000000h
134h	Error Report Syndrome Register (ERINFO2_SYND)	32	RO	00000000h
138h	Correctable Error Count Register (ERINFO2_CNT)	32	RW	00000000h
140h	Error Report Address Register (ERINFO3_ADDR)	32	RO	00000000h
144h	Error Report Syndrome Register (ERINFO3_SYND)	32	RO	00000000h
148h	Correctable Error Count Register (ERINFO3_CNT)	32	RW	00000000h
150h	Error Report Address Register (ERINFO4_ADDR)	32	RO	00000000h
154h	Error Report Syndrome Register (ERINFO4_SYND)	32	RO	00000000h
158h	Correctable Error Count Register (ERINFO4_CNT)	32	RW	00000000h
160h	Error Report Address Register (ERINFO5_ADDR)	32	RO	00000000h
164h	Error Report Syndrome Register (ERINFO5_SYND)	32	RO	00000000h
168h	Correctable Error Count Register (ERINFO5_CNT)	32	RW	00000000h

41.3.4.1 EMU EIGCR Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.3: EMU EIGCR Register Description

Field	Function
31 - 0 GEN	**Global Error Injection Enable** This bit will enable or disable all error injection functions. Write 0x89abcdef will enable this function. Read 0 means, global error injection is disabled Read 1 means, global error injection is enabled

Yuntu Microelectronics

41.3.4.2 EMU EICHCR Register

Offset: 4h

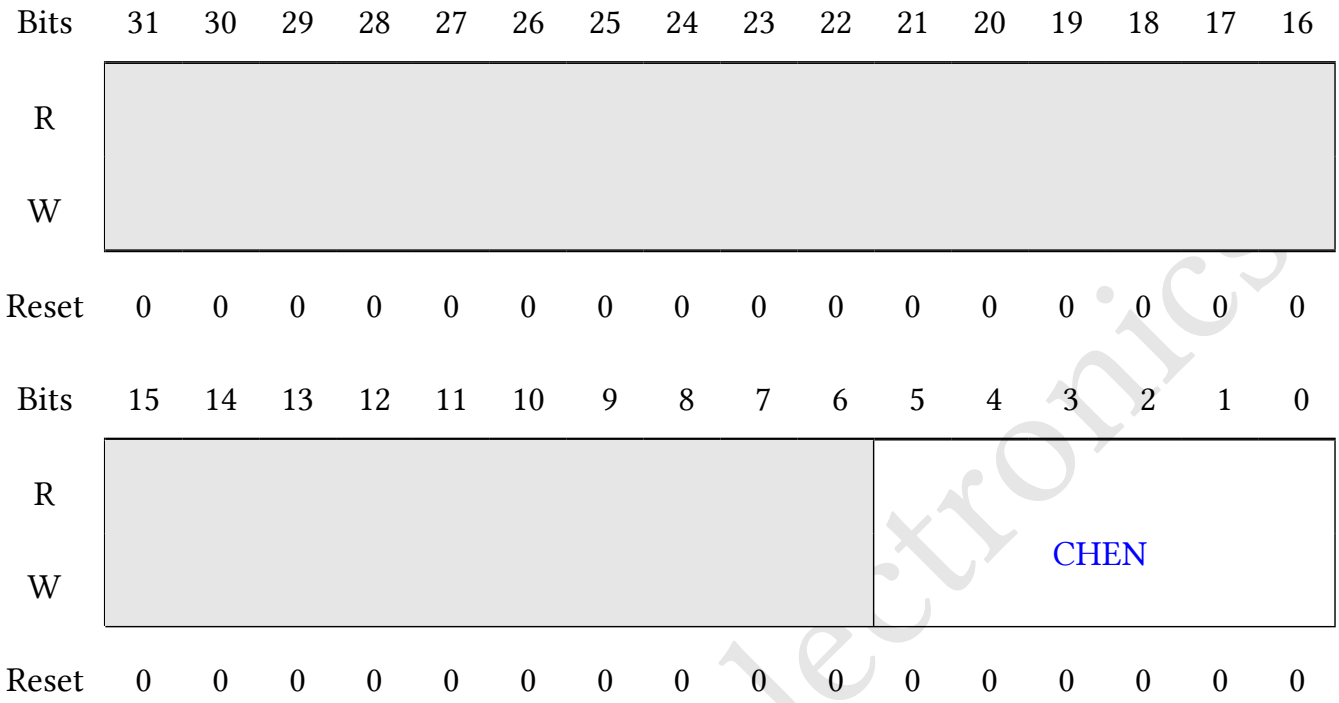


Table 41.4: EMU EICHCR Register Description

Field	Function
5 - 0 CHEN	**Channel n Error Injection Enable** This field enable or disable channel n error injection function. Enable channel n error injection function should enable this field and EIGCR[GEN]. Any write to EICHDn_ADDR, EICHDn_CHKBIT, EICHDn_DATAH/L register will clear this field. 0b - Channel n error injection is disabled 1b - Channel n error injection is enabled

41.3.4.3 EMU EICHD_ADDR Register

Offset: 10h + n * 10h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.5: EMU EICHD_ADDR Register Description

Field	Function
31 - 0 ADDR	**Channel n Error Injection Address** This field defines which address will be injected error when read access. when the ADDR is not 0, error will be injected according to this field's value. If ADDR is 0xFFFFFFFF, Error will be injected at all read access.

41.3.4.4 EMU EICHD_CHKBIT Register

Offset: 14h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W									CHKBIT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.6: EMU EICHD_CHKBIT Register Description

Field	Function
7 - 0 CHKBIT	**Error Injection CheckBit Mask** This field defines a bit-mapped mask that specifies whether the corresponding bit of the checkbit bus from the target on-chip memory should be inverted or remain unmodified. 0b - The corresponding bit of the checkbit bus remains unmodified. 1b - The corresponding bit of the checkbit bus is inverted.

Yuntu Microelectronics

41.3.4.5 EMU EICHD_DATAL Register

Offset: 18h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.7: EMU EICHD_DATAL Register Description

Field	Function
31 - 0 DATAL	**Data Mask** This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data bus from the target on-chip memory should be inverted or remain unmodified. 0b - The corresponding bit of the databit bus remains unmodified. 1b - The corresponding bit of the databit bus is inverted.

41.3.4.6 EMU EICHD_DATAH Register

Offset: 1ch + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATAH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATAH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.8: EMU EICHD_DATAH Register Description

Field	Function
31 - 0	**Data Mask**

Yuntu Microelectronics

Field	Function
DATAH	This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data bus from the target on-chip memory should be inverted or remain unmodified. 0b - The corresponding bit of the databit bus remains unmodified. 1b - The corresponding bit of the databit bus is inverted.

41.3.4.7 EMU ERIE Register

Offset: 100h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W											NCIE					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W											SCIE					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.9: EMU ERIE Register Description

Field	Function
21 - 16 NCIE	**Non-Correctable Interrupt Enable** Non-Correctable error interrupt enable. 0b - Channel n Non-Correctable Error interrupt is disabled 1b - Channel n Non-Correctable Error interrupt is enabled
5 - 0 SCIE	**Single Correctable Interrupt Enable** Single correctable error interrupt enable 0b - Channel n Single Correctable Error interrupt is disabled 1b - Channel n Single Correctable Error interrupt is enabled

41.3.4.8 EMU ERIF Register

Offset: 104h

Yuntu Microelectronics

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R											NCIF					
W											W1C					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R											SCIF					
W											W1C					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.10: EMU ERIF Register Description

Field	Function
21 - 16 NCIF	**Non-Correctable Interrupt Flag** Non-Correctable error interrupt flag. NCIF flag is kept when software reset but cleared when POR. 0b - Channel n Non-Correctable Error interrupt not detected 1b - Channel n Non-Correctable Error interrupt detected
5 - 0 SCIF	**Single Correctable Interrupt Flag** Single Correctable error interrupt flag. SCIF flag is kept when software reset but cleared when POR. 0b - Channel n Single Correctable Error interrupt not detected 1b - Channel n Single Correctable Error interrupt detected

41.3.4.9 EMU ERINFO_ADDR Register

Offset: 110h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.11: EMU ERINFO_ADDR Register Description

Field	Function
31 - 0	**Error Report Address n**

Yuntu Microelectronics

Field	Function
ADDR	This field contains the faulting system address of the last recorded ECC event on on-chip memories.

41.3.4.10 EMU ERINFO_SYND Register

Offset: 114h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									SYNDROME							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 41.12: EMU ERINFO_SYND Register Description

Field	Function
7 - 0 SYNDROME	**Error Report Syndrome n** This field contains the syndrome of the last recorded ECC event on on-chip memories. For 32 bit memory, just low 7 bits([6:0]) is effective. For 64 bit memory, the 8 bits ([7:0]) is effective.

41.3.4.11 EMU ERINFO_CNT Register

Offset: 118h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W									CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 41.13: EMU ERINFO_CNT Register Description

Field	Function
7 - 0 CNT	**Correctable Error Count** For each correctable error event, the EMU increments this field's error count value until the counter reaches its maximum value 0xFF. CNT will not change once it reaches 0xFF and will not wrap even if error occurs. Read this field to determine the correctable error count value so far. Write zero to this field to reset the counter. Writing non-zero (or partial zero) values has no effect.

41.4 Functional Description

41.4.1 General

This section gives detailed information on the internal operation of the module.

41.4.2 Error Injection

- Error data could be injected to read access.
- Error data could be injected to overall read access or a particular read access. When EICHn_ADDR is 0xFFFFFFFF, error data will be injected to all read access, otherwise, the error data will be injected to a particular read access.
- Injection method: when read access and address match, the bus read data will be the XOR of on-chip memories read data(data and its ECC check-bit) and EICHn_CHKBIT/DATAH/DATAL. EMU support single-bit error and multi-bit error, an attempt to invert more than 2 bits in one operation might result in undefined behavior.
- Use two-stage enable mechanism providing protection. After configuring each channel's EICHn_ADDR, EICHn_CHKBIT and EICHn_DATAH/L, to properly enable the error injection mechanism, please set channel enable and then set global enable.

41.4.3 Error Report

- Error Report could report error when reading on-chip memories(single-bit or multi-bit error). Each channel has an array registers to state the error(ERINFOn_ADDR/ERINFOn_SYND/ERINFO_CNT).
- When detecting an error(single-bit error or multi-bit error), the register will keep its address and syndrome. If the error is single-bit error, the ERINFOn_CNT will increase(until the count reaches its max value).
- Error Report provides an option to generate an interrupt, upon the report of an error and the corresponding interrupt enable bit set.
- To clear error flag, write 1 to the corresponding interrupt flag. To clear the correctable error count, write 0 to the count register.

Yuntu Microelectronics

Chapter 42

Interrupt Monitor (INTM)

42.1 Chip-specific INTM Information

42.1.1 Instantiation Information

This chip contains one INTM, the INTM contains four monitors. Each monitor could select the monitored interrupt source. Each monitor has a 24-bit timer to monitor the latency of the response on interrupt request.

42.2 Introduction

The INTM module provides a mechanism to monitor the latency of the responses on interrupt requests to ensure that the processing of these critical interrupts executes within the expected time frame, increasing the reliability of the device.

42.2.1 Features

- Up to 4 programmable monitors
- Programmable monitored interrupt source per monitor
- Programmable 24-bit counter threshold per monitor
- Timer expired status bit per monitor
- One overall interrupt acknowledge for all monitors
- One overall enable for all monitors

42.3 Register Definition

The INTM provides a programming model mapped to an on-platform peripheral slot.

- All system bus masters can access the programming model using only 32-bit (word) accesses
- Attempted access address which over the last effective address to the programming model will generate a bus error termination.

INTM provides a way to set a threshold value for the interrupt latency from interrupt request to interrupt acknowledge. The purpose of this function is to provide a mechanism to monitor the latency of interrupt sources to ensure that these critical interrupts execute within the expected time frame, increasing the reliability of the device. The hardware for this function is isolated in its own hierarchy to decrease the risk of fault interference.

Yuntu Microelectronics

An error will generate if the interrupt processing latency exceeds the programmed threshold. The error indication can be read from the Monitor Status register(MONn_STS).

42.3.1 INTM Memory Map

INTM base address 401c4000h

Table 42.1: INTM Memory Map

Offset	Register	Width	Access	Reset Value
0h	Global Control Register (GCTRL)	32	RW	00000000h
4h	Global Interrupt Acknowledge Register (GINTACK)	32	RW	000000FFh
10h	Monitor Channel Selection Register (MON0_SEL)	32	RW	000000FFh
14h	Monitor Timer Threshold Register (MON0_THD)	32	RW	00FFFFFFh
18h	Monitor Timer Value Register (MON0_TMR)	32	RW	00000000h
1ch	Monitor Status Register (MON0_STS)	32	RO	00000000h
20h	Monitor Channel Selection Register (MON1_SEL)	32	RW	000000FFh
24h	Monitor Timer Threshold Register (MON1_THD)	32	RW	00FFFFFFh
28h	Monitor Timer Value Register (MON1_TMR)	32	RW	00000000h
2ch	Monitor Status Register (MON1_STS)	32	RO	00000000h
30h	Monitor Channel Selection Register (MON2_SEL)	32	RW	000000FFh
34h	Monitor Timer Threshold Register (MON2_THD)	32	RW	00FFFFFFh
38h	Monitor Timer Value Register (MON2_TMR)	32	RW	00000000h
3ch	Monitor Status Register (MON2_STS)	32	RO	00000000h
40h	Monitor Channel Selection Register (MON3_SEL)	32	RW	000000FFh
44h	Monitor Timer Threshold Register (MON3_THD)	32	RW	00FFFFFFh
48h	Monitor Timer Value Register (MON3_TMR)	32	RW	00000000h
4ch	Monitor Status Register (MON3_STS)	32	RO	00000000h

Yuntu Microelectronics

42.3.1.1 INTM GCTRL Register

Offset: 0h

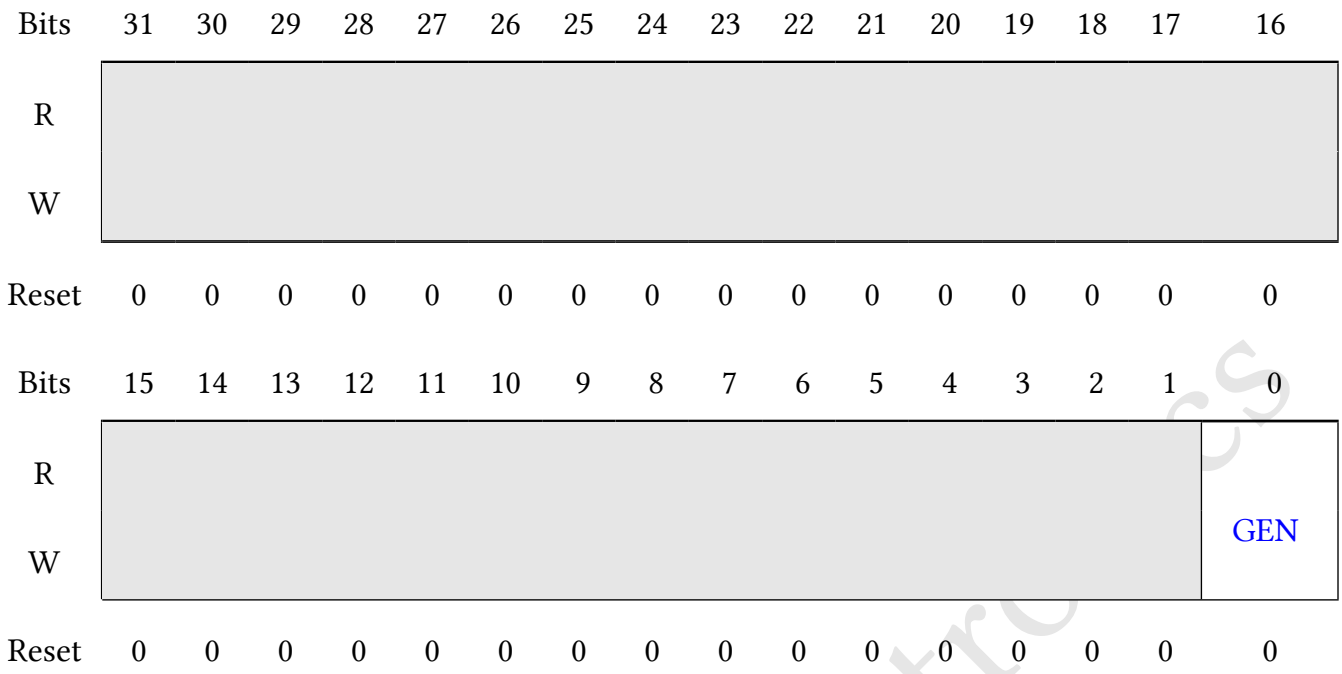


Table 42.2: INTM GCTRL Register Description

Field	Function
0 GEN	**Global Enable** The field is used to enable or disable all monitors 0b - interrupt monitor is disabled 1b - interrupt monitor is enabled

Yuntu Microelectronics

42.3.1.2 INTM GINTACK Register

Offset: 4h

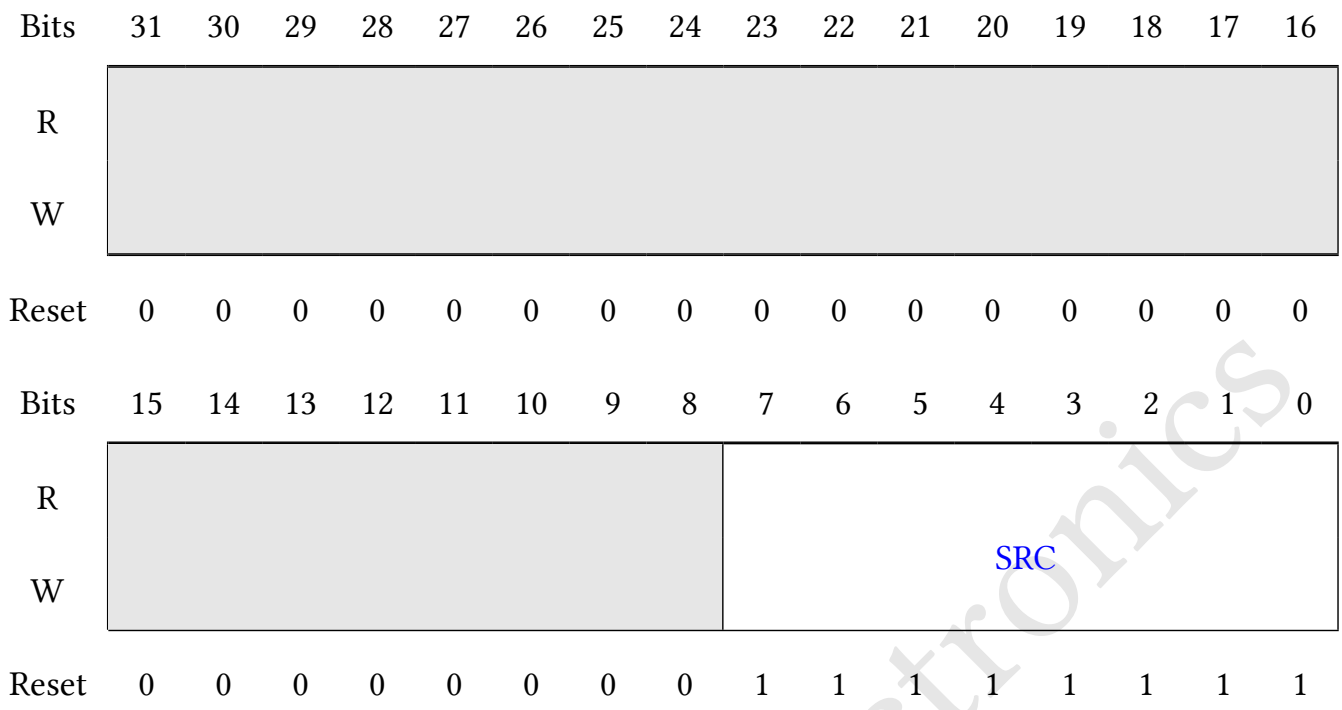


Table 42.3: INTM GINTACK Register Description

Field	Function
7 - 0 SRC	**Interrupt Source** This field is used to specify the interrupt request number to stop the TIMERn.

Yuntu Microelectronics

42.3.1.3 INTM MON_SEL Register

Offset: 10h + n * 10h

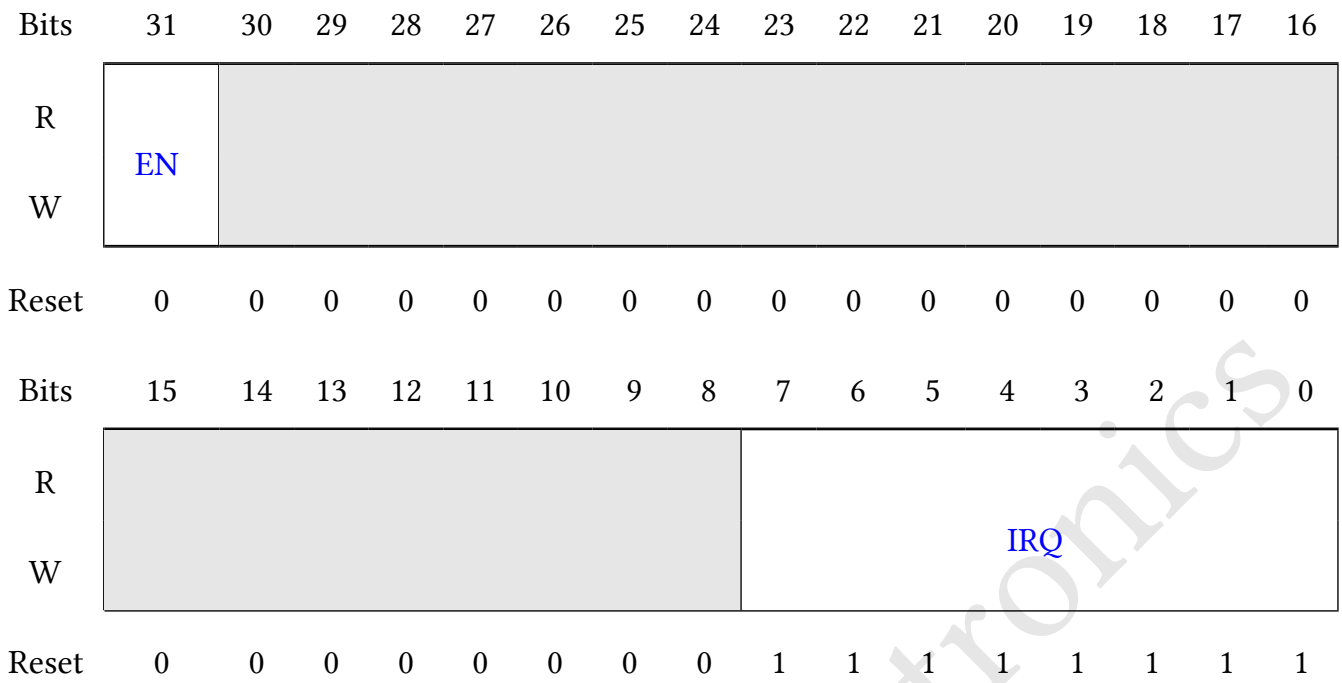
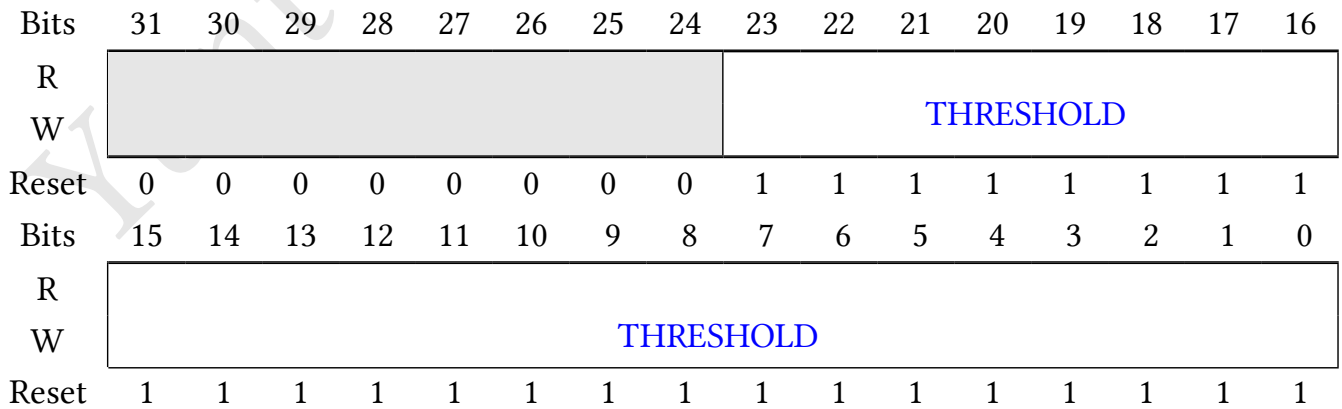


Table 42.4: INTM MON_SEL Register Description

Field	Function
31 EN	**MONn Enable** This field is used to enable or disable this monitor
7 - 0 IRQ	**MONn IRQ Number** This field is used to select the monitored interrupt source

42.3.1.4 INTM MON_THD Register

Offset: 14h + n * 10h



Yuntu Microelectronics

Table 42.5: INTM MON_THD Register Description

Field	Function
23 - 0 THRESHOLD	**MONn Timer Threshold** This field is used to configure the maximum number of INTM clock cycles allowed for the monitored interrupt request

42.3.1.5 INTM MON_TMR Register

Offset: 18h + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W									CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 42.6: INTM MON_TMR Register Description

Field	Function
23 - 0 CNT	**MONn Timer Value** This field is used to count the number of INTM clock cycles, up to 24 bits of resolution, write any value to reset the CNT.

42.3.1.6 INTM MON_STS Register

Offset: 1ch + n * 10h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W																OF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Yuntu Microelectronics

Table 42.7: INTM MON_STS Register Description

Field	Function
0 OF	**MONn Timer Overflow** This field indicates whether MONn_TMR value has exceeded the MONn_THRESHOLD value

42.4 Functional Description

Each monitor can be set enable or disable, and select a particular interrupt source to be monitored.

Before enabling the monitor, need to set the timer threshold(MONn_THD) for the interrupt latency from interrupt request to interrupt acknowledge. If no interrupt acknowledge before the timer(MONn_TMR) exceeding the threshold, this monitor will generate an interrupt, and the status will indicate the scene.

After receiving a monitored interrupt source request, it will reset the timer count, and the timer begins to increase until the monitored interrupt responds or the timer comes to the timer threshold.

When a monitored interrupt is generated, write the interrupt source number to GINTACK to clear the interrupt.

Yuntu Microelectronics

Chapter 43

Peripheral Protection Unit (PPU)

43.1 Introduction

The PPU (Peripheral Protection Unit) module is used to protect the defined memory-mapped address locations from being written.

This module is located between the module under protection and the peripheral interface.

43.1.1 Features

The PPU includes the following features:

- Write access for the module under protection can be restricted to the supervisor mode only
- Maximum 4KB register slot size of module under protection
- Multiple ways are present to set the lock bits
- Once the lock bits are set, the registers could be protected from modification

Yuntu Microelectronics

43.1.2 Block Diagram

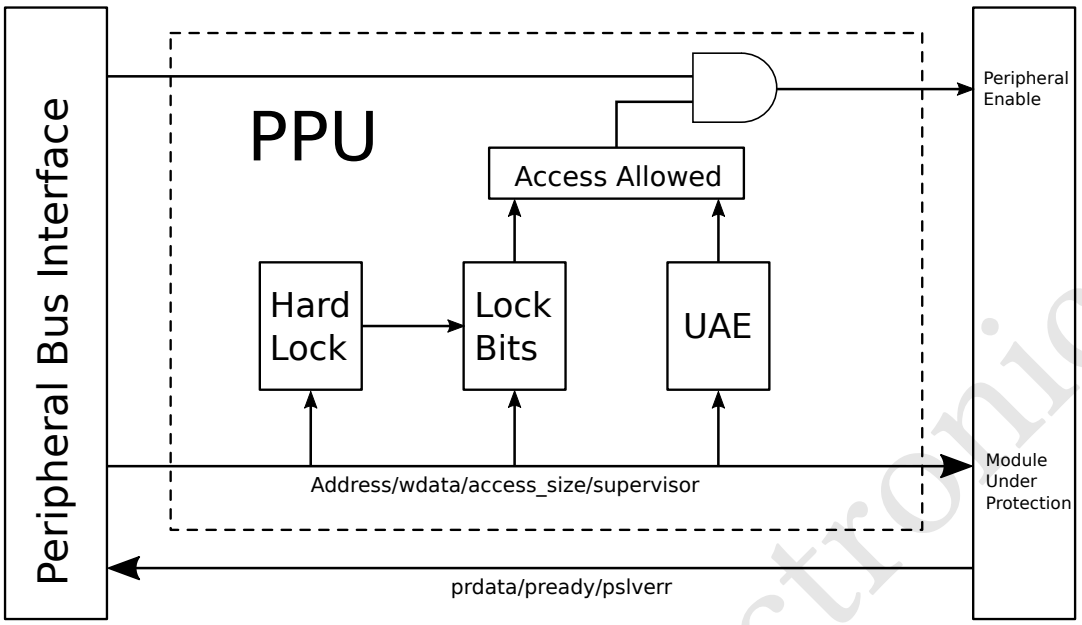


Figure 43.1: Block Diagram of PPU

43.1.3 Signal Description

The PPU module has no external pins.

43.1.4 PPU Protection IPs and Registers

Table 43.1: PPU protection table

Module name	Registers' name	Protected size in bytes	Offset
TMR0	CTRL	4	0x000
TMR0	PRS	4	0x004
TMR0	CNT	4	0x008
TMR0	CH0_CTRL	4	0x010
TMR0	CH0_INT	4	0x014
TMR0	CH0_CMP	4	0x018
TMR0	CH1_CTRL	4	0x020
TMR0	CH1_INT	4	0x024
TMR0	CH1_CMP	4	0x028
TMR0	CH2_CTRL	4	0x030
TMR0	CH2_INT	4	0x034
TMR0	CH2_CMP	4	0x038
TMR0	CH3_CTRL	4	0x040
TMR0	CH3_INT	4	0x044
TMR0	CH3_CMP	4	0x048
TMR1	CTRL	4	0x000
TMR1	PRS	4	0x004

Yuntu Microelectronics

Module name	Registers' name	Protected size in bytes	Offset
TMR1	CNT	4	0x008
TMR1	CH0_CTRL	4	0x010
TMR1	CH0_INT	4	0x014
TMR1	CH0_CMP	4	0x018
TMR1	CH1_CTRL	4	0x020
TMR1	CH1_INT	4	0x024
TMR1	CH1_CMP	4	0x028
TMR1	CH2_CTRL	4	0x030
TMR1	CH2_INT	4	0x034
TMR1	CH2_CMP	4	0x038
TMR1	CH3_CTRL	4	0x040
TMR1	CH3_INT	4	0x044
TMR1	CH3_CMP	4	0x048
SCU	CLKS	4	0x000
SCU	DIV	4	0x004
SCU	FIRC_CTRL	4	0x010
SCU	SIRC_CTRL	4	0x014
SCU	FXOSC_CTRL	4	0x018
SCU	SXOSC_CTRL	4	0x01C
SCU	PLL_CTRL	4	0x020
SCU	PLLCFG	4	0x024
SCU	PLLFM	4	0x028
SCU	PLLFD	4	0x02c
SCU	CLK_OUT	4	0x030
SCU	FIRCTRIM	4	0x034
SCU	SIRCTRIM	4	0x038
SCU	CMUCTRL	4	0x040
SCU	CMU0CMPH	4	0x048
SCU	CMU0CMPL	4	0x04C
SCU	CMU1CMPH	4	0x050
SCU	CMU1CMPL	4	0x054
SCU	CMU2CMPH	4	0x058
SCU	CMU2CMPL	4	0x05c
SCU	CMU3CMPH	4	0x060
SCU	CMU3CMPL	4	0x064
PCU	INTE	4	0x008
PCU	CTRL	4	0x00C
PCU	TRIM50	4	0x010
PCU	TRIM25	4	0x014
PCU	TRIM11	4	0x018
PCU	TEST	4	0x01C
RCU	RPFR	4	0x004
EFM	CTRL	4	0x000
EFM	CMD	4	0x008
EFM	CMD_PROT	4	0x00C
EFM	TIMING1	4	0x010
EFM	TIMING2	4	0x014
EFM	EFM_ECCERR_INJECT_MSK_0	4	0x018
EFM	EFM_ECCERR_INJECT_MSK_1	4	0x01C
EFM	EFM_ECCERR_INJECT_ADDR	4	0x020
EFM	PROT0	4	0x024

Yuntu Microelectronics

Module name	Registers' name	Protected size in bytes	Offset
EFM	PROT1	4	0x028
EFM	PROT2	4	0x02C
IPC	DMA_CTRL	4	0x020
IPC	GPIO_CTRL	4	0x044
IPC	PCTRLA_CTRL	4	0x048
IPC	PCTRLB_CTRL	4	0x04C
IPC	PCTRLC_CTRL	4	0x050
IPC	PCTRLD_CTRL	4	0x054
IPC	PCTRLE_CTRL	4	0x058
IPC	LINFLEX0_CTRL	4	0x06C
IPC	LINFLEX1_CTRL	4	0x070
IPC	LINFLEX2_CTRL	4	0x074
IPC	LINFLEX3_CTRL	4	0x078
IPC	LINFLEX4_CTRL	4	0x07C
IPC	LINFLEX5_CTRL	4	0x080
IPC	LINFLEX6_CTRL	4	0x084
IPC	LINFLEX7_CTRL	4	0x088
IPC	LINFLEX8_CTRL	4	0x08C
IPC	LINFLEX9_CTRL	4	0x090
IPC	I2C0_CTRL	4	0x094
IPC	I2C1_CTRL	4	0x098
IPC	I2C2_CTRL	4	0x09C
IPC	I2C3_CTRL	4	0x0A0
IPC	I2C4_CTRL	4	0x0A4
IPC	SPI0_CTRL	4	0x0A8
IPC	SPI1_CTRL	4	0x0AC
IPC	SPI2_CTRL	4	0x0B0
IPC	SPI3_CTRL	4	0x0B4
IPC	SPI4_CTRL	4	0x0B8
IPC	SPI5_CTRL	4	0x0BC
IPC	SPI6_CTRL	4	0x0C0
IPC	SPI7_CTRL	4	0x0C4
IPC	FlexCAN0_CTRL	4	0x0D0
IPC	FlexCAN1_CTRL	4	0x0D4
IPC	FlexCAN2_CTRL	4	0x0D8
IPC	FlexCAN3_CTRL	4	0x0DC
IPC	FlexCAN4_CTRL	4	0x0E0
IPC	FlexCAN5_CTRL	4	0x0E4
IPC	FlexCAN6_CTRL	4	0x0E8
IPC	FlexCAN7_CTRL	4	0x0EC
IPC	ADC0_CTRL	4	0x100
IPC	ADC1_CTRL	4	0x104
IPC	ACMP0_CTRL	4	0x10C
IPC	ACMP1_CTRL	4	0x110
IPC	PTU0_CTRL	4	0x118
IPC	PTU1_CTRL	4	0x11C
IPC	TMU_CTRL	4	0x120
IPC	eTMR0_CTRL	4	0x124
IPC	eTMR1_CTRL	4	0x128
IPC	eTMR2_CTRL	4	0x12C
IPC	eTMR3_CTRL	4	0x130

Yuntu Microelectronics

Module name	Registers' name	Protected size in bytes	Offset
IPC	eTMR4_CTRL	4	0x134
IPC	eTMR5_CTRL	4	0x138
IPC	MPWM0_CTRL	4	0X144
IPC	MPWM1_CTRL	4	0X148
IPC	MPWM2_CTRL	4	0X14C
IPC	TMR0_CTRL	4	0x150
IPC	TMR1_CTRL	4	0x154
IPC	SENT0_CTRL	4	0X160
IPC	SENT1_CTRL	4	0X164
IPC	pTMR0_CTRL	4	0x168
IPC	pTMR1_CTRL	4	0x16C
IPC	pTMR2_CTRL	4	0x170
IPC	lpTMR0_CTRL	4	0x174
IPC	RTC_CTRL	4	0x180
IPC	WKU_CTRL	4	0x188
IPC	PCRC_CTRL	4	0x194
IPC	TRNG_CTRL	4	0x198
IPC	HCU_CTRL	4	0x19C
IPC	QSPI_CTRL	4	0x1A0
IPC	WDG_CTRL	4	0x1A8
IPC	FMU_CTRL	4	0x1C0
IPC	EMU_CTRL	4	0x1C8
IPC	SAI0_CTRL	4	0X1CC
IPC	SAI1_CTRL	4	0X1D0
IPC	CIM_CTRL	4	0x1EC
IPC	SCU_CTRL	4	0x1F0
IPC	PCU_CTRL	4	0x1F8
IPC	RCU_CTRL	4	0x1FC
CIM	CTRL0	4	0x000
CIM	CTRL1	4	0x004
CIM	FPUIE	4	0x010
CIM	SWTRIG	4	0x014
CIM	ETMROPT0	4	0x018
CIM	ETMROPT1	4	0x01C
CIM	ETMROPT2	4	0x020
CIM	ETMROPT3	4	0x024
CIM	ETMROPT4	4	0x028
CIM	ETMROPT5	4	0x02C
CIM	ETMR_GLB_OPT	4	0x03C
CIM	SWFAULT	4	0x040
CIM	DCCM_EI	4	0x044
CIM	SBIR	4	0x048
CIM	BACKUP0	4	0x0A8
CIM	BACKUP1	4	0x0AC
CIM	BOOT_CFG	4	0x0C0

43.2 Register Definition

Yuntu Microelectronics

Table 43.2: PPU Register Map

Offset	Description
0x0000 ~ 0x0FFF	Normal register for each IP module
0x1000 ~ 0x1FFF	Reserved
0x2000 ~ 0x2FFF	Mirror register
0x3000 ~ 0x33FF	Soft Lock register
0x3400 ~ 0x3FF8	Reserved
0x3FFC	Configuration register

43.2.1 PPU Memory Map

Table 43.3: PPU Memory Map

Offset	Register	Width	Access	Reset Value
3000h	Soft Lock Register (SLOCK0)	8	RW	00h
3001h	Soft Lock Register (SLOCK1)	8	RW	00h
3002h	Soft Lock Register (SLOCK2)	8	RW	00h
3003h	Soft Lock Register (SLOCK3)	8	RW	00h
3004h	Soft Lock Register (SLOCK4)	8	RW	00h
3005h	Soft Lock Register (SLOCK5)	8	RW	00h
3006h	Soft Lock Register (SLOCK6)	8	RW	00h
3007h	Soft Lock Register (SLOCK7)	8	RW	00h
3008h	Soft Lock Register (SLOCK8)	8	RW	00h
3009h	Soft Lock Register (SLOCK9)	8	RW	00h
300ah	Soft Lock Register (SLOCK10)	8	RW	00h
300bh	Soft Lock Register (SLOCK11)	8	RW	00h
300ch	Soft Lock Register (SLOCK12)	8	RW	00h
300dh	Soft Lock Register (SLOCK13)	8	RW	00h
300eh	Soft Lock Register (SLOCK14)	8	RW	00h
300fh	Soft Lock Register (SLOCK15)	8	RW	00h
3010h	Soft Lock Register (SLOCK16)	8	RW	00h
3011h	Soft Lock Register (SLOCK17)	8	RW	00h
3012h	Soft Lock Register (SLOCK18)	8	RW	00h
3013h	Soft Lock Register (SLOCK19)	8	RW	00h
3014h	Soft Lock Register (SLOCK20)	8	RW	00h
3015h	Soft Lock Register (SLOCK21)	8	RW	00h
3016h	Soft Lock Register (SLOCK22)	8	RW	00h
3017h	Soft Lock Register (SLOCK23)	8	RW	00h
3018h	Soft Lock Register (SLOCK24)	8	RW	00h
3019h	Soft Lock Register (SLOCK25)	8	RW	00h
301ah	Soft Lock Register (SLOCK26)	8	RW	00h
301bh	Soft Lock Register (SLOCK27)	8	RW	00h
301ch	Soft Lock Register (SLOCK28)	8	RW	00h
301dh	Soft Lock Register (SLOCK29)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
301eh	Soft Lock Register (SLOCK30)	8	RW	00h
301fh	Soft Lock Register (SLOCK31)	8	RW	00h
3020h	Soft Lock Register (SLOCK32)	8	RW	00h
3021h	Soft Lock Register (SLOCK33)	8	RW	00h
3022h	Soft Lock Register (SLOCK34)	8	RW	00h
3023h	Soft Lock Register (SLOCK35)	8	RW	00h
3024h	Soft Lock Register (SLOCK36)	8	RW	00h
3025h	Soft Lock Register (SLOCK37)	8	RW	00h
3026h	Soft Lock Register (SLOCK38)	8	RW	00h
3027h	Soft Lock Register (SLOCK39)	8	RW	00h
3028h	Soft Lock Register (SLOCK40)	8	RW	00h
3029h	Soft Lock Register (SLOCK41)	8	RW	00h
302ah	Soft Lock Register (SLOCK42)	8	RW	00h
302bh	Soft Lock Register (SLOCK43)	8	RW	00h
302ch	Soft Lock Register (SLOCK44)	8	RW	00h
302dh	Soft Lock Register (SLOCK45)	8	RW	00h
302eh	Soft Lock Register (SLOCK46)	8	RW	00h
302fh	Soft Lock Register (SLOCK47)	8	RW	00h
3030h	Soft Lock Register (SLOCK48)	8	RW	00h
3031h	Soft Lock Register (SLOCK49)	8	RW	00h
3032h	Soft Lock Register (SLOCK50)	8	RW	00h
3033h	Soft Lock Register (SLOCK51)	8	RW	00h
3034h	Soft Lock Register (SLOCK52)	8	RW	00h
3035h	Soft Lock Register (SLOCK53)	8	RW	00h
3036h	Soft Lock Register (SLOCK54)	8	RW	00h
3037h	Soft Lock Register (SLOCK55)	8	RW	00h
3038h	Soft Lock Register (SLOCK56)	8	RW	00h
3039h	Soft Lock Register (SLOCK57)	8	RW	00h
303ah	Soft Lock Register (SLOCK58)	8	RW	00h
303bh	Soft Lock Register (SLOCK59)	8	RW	00h
303ch	Soft Lock Register (SLOCK60)	8	RW	00h
303dh	Soft Lock Register (SLOCK61)	8	RW	00h
303eh	Soft Lock Register (SLOCK62)	8	RW	00h
303fh	Soft Lock Register (SLOCK63)	8	RW	00h
3040h	Soft Lock Register (SLOCK64)	8	RW	00h
3041h	Soft Lock Register (SLOCK65)	8	RW	00h
3042h	Soft Lock Register (SLOCK66)	8	RW	00h
3043h	Soft Lock Register (SLOCK67)	8	RW	00h
3044h	Soft Lock Register (SLOCK68)	8	RW	00h
3045h	Soft Lock Register (SLOCK69)	8	RW	00h
3046h	Soft Lock Register (SLOCK70)	8	RW	00h
3047h	Soft Lock Register (SLOCK71)	8	RW	00h
3048h	Soft Lock Register (SLOCK72)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3049h	Soft Lock Register (SLOCK73)	8	RW	00h
304ah	Soft Lock Register (SLOCK74)	8	RW	00h
304bh	Soft Lock Register (SLOCK75)	8	RW	00h
304ch	Soft Lock Register (SLOCK76)	8	RW	00h
304dh	Soft Lock Register (SLOCK77)	8	RW	00h
304eh	Soft Lock Register (SLOCK78)	8	RW	00h
304fh	Soft Lock Register (SLOCK79)	8	RW	00h
3050h	Soft Lock Register (SLOCK80)	8	RW	00h
3051h	Soft Lock Register (SLOCK81)	8	RW	00h
3052h	Soft Lock Register (SLOCK82)	8	RW	00h
3053h	Soft Lock Register (SLOCK83)	8	RW	00h
3054h	Soft Lock Register (SLOCK84)	8	RW	00h
3055h	Soft Lock Register (SLOCK85)	8	RW	00h
3056h	Soft Lock Register (SLOCK86)	8	RW	00h
3057h	Soft Lock Register (SLOCK87)	8	RW	00h
3058h	Soft Lock Register (SLOCK88)	8	RW	00h
3059h	Soft Lock Register (SLOCK89)	8	RW	00h
305ah	Soft Lock Register (SLOCK90)	8	RW	00h
305bh	Soft Lock Register (SLOCK91)	8	RW	00h
305ch	Soft Lock Register (SLOCK92)	8	RW	00h
305dh	Soft Lock Register (SLOCK93)	8	RW	00h
305eh	Soft Lock Register (SLOCK94)	8	RW	00h
305fh	Soft Lock Register (SLOCK95)	8	RW	00h
3060h	Soft Lock Register (SLOCK96)	8	RW	00h
3061h	Soft Lock Register (SLOCK97)	8	RW	00h
3062h	Soft Lock Register (SLOCK98)	8	RW	00h
3063h	Soft Lock Register (SLOCK99)	8	RW	00h
3064h	Soft Lock Register (SLOCK100)	8	RW	00h
3065h	Soft Lock Register (SLOCK101)	8	RW	00h
3066h	Soft Lock Register (SLOCK102)	8	RW	00h
3067h	Soft Lock Register (SLOCK103)	8	RW	00h
3068h	Soft Lock Register (SLOCK104)	8	RW	00h
3069h	Soft Lock Register (SLOCK105)	8	RW	00h
306ah	Soft Lock Register (SLOCK106)	8	RW	00h
306bh	Soft Lock Register (SLOCK107)	8	RW	00h
306ch	Soft Lock Register (SLOCK108)	8	RW	00h
306dh	Soft Lock Register (SLOCK109)	8	RW	00h
306eh	Soft Lock Register (SLOCK110)	8	RW	00h
306fh	Soft Lock Register (SLOCK111)	8	RW	00h
3070h	Soft Lock Register (SLOCK112)	8	RW	00h
3071h	Soft Lock Register (SLOCK113)	8	RW	00h
3072h	Soft Lock Register (SLOCK114)	8	RW	00h
3073h	Soft Lock Register (SLOCK115)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3074h	Soft Lock Register (SLOCK116)	8	RW	00h
3075h	Soft Lock Register (SLOCK117)	8	RW	00h
3076h	Soft Lock Register (SLOCK118)	8	RW	00h
3077h	Soft Lock Register (SLOCK119)	8	RW	00h
3078h	Soft Lock Register (SLOCK120)	8	RW	00h
3079h	Soft Lock Register (SLOCK121)	8	RW	00h
307ah	Soft Lock Register (SLOCK122)	8	RW	00h
307bh	Soft Lock Register (SLOCK123)	8	RW	00h
307ch	Soft Lock Register (SLOCK124)	8	RW	00h
307dh	Soft Lock Register (SLOCK125)	8	RW	00h
307eh	Soft Lock Register (SLOCK126)	8	RW	00h
307fh	Soft Lock Register (SLOCK127)	8	RW	00h
3080h	Soft Lock Register (SLOCK128)	8	RW	00h
3081h	Soft Lock Register (SLOCK129)	8	RW	00h
3082h	Soft Lock Register (SLOCK130)	8	RW	00h
3083h	Soft Lock Register (SLOCK131)	8	RW	00h
3084h	Soft Lock Register (SLOCK132)	8	RW	00h
3085h	Soft Lock Register (SLOCK133)	8	RW	00h
3086h	Soft Lock Register (SLOCK134)	8	RW	00h
3087h	Soft Lock Register (SLOCK135)	8	RW	00h
3088h	Soft Lock Register (SLOCK136)	8	RW	00h
3089h	Soft Lock Register (SLOCK137)	8	RW	00h
308ah	Soft Lock Register (SLOCK138)	8	RW	00h
308bh	Soft Lock Register (SLOCK139)	8	RW	00h
308ch	Soft Lock Register (SLOCK140)	8	RW	00h
308dh	Soft Lock Register (SLOCK141)	8	RW	00h
308eh	Soft Lock Register (SLOCK142)	8	RW	00h
308fh	Soft Lock Register (SLOCK143)	8	RW	00h
3090h	Soft Lock Register (SLOCK144)	8	RW	00h
3091h	Soft Lock Register (SLOCK145)	8	RW	00h
3092h	Soft Lock Register (SLOCK146)	8	RW	00h
3093h	Soft Lock Register (SLOCK147)	8	RW	00h
3094h	Soft Lock Register (SLOCK148)	8	RW	00h
3095h	Soft Lock Register (SLOCK149)	8	RW	00h
3096h	Soft Lock Register (SLOCK150)	8	RW	00h
3097h	Soft Lock Register (SLOCK151)	8	RW	00h
3098h	Soft Lock Register (SLOCK152)	8	RW	00h
3099h	Soft Lock Register (SLOCK153)	8	RW	00h
309ah	Soft Lock Register (SLOCK154)	8	RW	00h
309bh	Soft Lock Register (SLOCK155)	8	RW	00h
309ch	Soft Lock Register (SLOCK156)	8	RW	00h
309dh	Soft Lock Register (SLOCK157)	8	RW	00h
309eh	Soft Lock Register (SLOCK158)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
309fh	Soft Lock Register (SLOCK159)	8	RW	00h
30a0h	Soft Lock Register (SLOCK160)	8	RW	00h
30a1h	Soft Lock Register (SLOCK161)	8	RW	00h
30a2h	Soft Lock Register (SLOCK162)	8	RW	00h
30a3h	Soft Lock Register (SLOCK163)	8	RW	00h
30a4h	Soft Lock Register (SLOCK164)	8	RW	00h
30a5h	Soft Lock Register (SLOCK165)	8	RW	00h
30a6h	Soft Lock Register (SLOCK166)	8	RW	00h
30a7h	Soft Lock Register (SLOCK167)	8	RW	00h
30a8h	Soft Lock Register (SLOCK168)	8	RW	00h
30a9h	Soft Lock Register (SLOCK169)	8	RW	00h
30aah	Soft Lock Register (SLOCK170)	8	RW	00h
30abh	Soft Lock Register (SLOCK171)	8	RW	00h
30ach	Soft Lock Register (SLOCK172)	8	RW	00h
30adh	Soft Lock Register (SLOCK173)	8	RW	00h
30aeh	Soft Lock Register (SLOCK174)	8	RW	00h
30afh	Soft Lock Register (SLOCK175)	8	RW	00h
30b0h	Soft Lock Register (SLOCK176)	8	RW	00h
30b1h	Soft Lock Register (SLOCK177)	8	RW	00h
30b2h	Soft Lock Register (SLOCK178)	8	RW	00h
30b3h	Soft Lock Register (SLOCK179)	8	RW	00h
30b4h	Soft Lock Register (SLOCK180)	8	RW	00h
30b5h	Soft Lock Register (SLOCK181)	8	RW	00h
30b6h	Soft Lock Register (SLOCK182)	8	RW	00h
30b7h	Soft Lock Register (SLOCK183)	8	RW	00h
30b8h	Soft Lock Register (SLOCK184)	8	RW	00h
30b9h	Soft Lock Register (SLOCK185)	8	RW	00h
30bah	Soft Lock Register (SLOCK186)	8	RW	00h
30bbh	Soft Lock Register (SLOCK187)	8	RW	00h
30bch	Soft Lock Register (SLOCK188)	8	RW	00h
30bdh	Soft Lock Register (SLOCK189)	8	RW	00h
30beh	Soft Lock Register (SLOCK190)	8	RW	00h
30bfh	Soft Lock Register (SLOCK191)	8	RW	00h
30c0h	Soft Lock Register (SLOCK192)	8	RW	00h
30c1h	Soft Lock Register (SLOCK193)	8	RW	00h
30c2h	Soft Lock Register (SLOCK194)	8	RW	00h
30c3h	Soft Lock Register (SLOCK195)	8	RW	00h
30c4h	Soft Lock Register (SLOCK196)	8	RW	00h
30c5h	Soft Lock Register (SLOCK197)	8	RW	00h
30c6h	Soft Lock Register (SLOCK198)	8	RW	00h
30c7h	Soft Lock Register (SLOCK199)	8	RW	00h
30c8h	Soft Lock Register (SLOCK200)	8	RW	00h
30c9h	Soft Lock Register (SLOCK201)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
30cah	Soft Lock Register (SLOCK202)	8	RW	00h
30cbh	Soft Lock Register (SLOCK203)	8	RW	00h
30cch	Soft Lock Register (SLOCK204)	8	RW	00h
30cdh	Soft Lock Register (SLOCK205)	8	RW	00h
30ceh	Soft Lock Register (SLOCK206)	8	RW	00h
30cfh	Soft Lock Register (SLOCK207)	8	RW	00h
30d0h	Soft Lock Register (SLOCK208)	8	RW	00h
30d1h	Soft Lock Register (SLOCK209)	8	RW	00h
30d2h	Soft Lock Register (SLOCK210)	8	RW	00h
30d3h	Soft Lock Register (SLOCK211)	8	RW	00h
30d4h	Soft Lock Register (SLOCK212)	8	RW	00h
30d5h	Soft Lock Register (SLOCK213)	8	RW	00h
30d6h	Soft Lock Register (SLOCK214)	8	RW	00h
30d7h	Soft Lock Register (SLOCK215)	8	RW	00h
30d8h	Soft Lock Register (SLOCK216)	8	RW	00h
30d9h	Soft Lock Register (SLOCK217)	8	RW	00h
30dah	Soft Lock Register (SLOCK218)	8	RW	00h
30dbh	Soft Lock Register (SLOCK219)	8	RW	00h
30dch	Soft Lock Register (SLOCK220)	8	RW	00h
30ddh	Soft Lock Register (SLOCK221)	8	RW	00h
30deh	Soft Lock Register (SLOCK222)	8	RW	00h
30dfh	Soft Lock Register (SLOCK223)	8	RW	00h
30e0h	Soft Lock Register (SLOCK224)	8	RW	00h
30e1h	Soft Lock Register (SLOCK225)	8	RW	00h
30e2h	Soft Lock Register (SLOCK226)	8	RW	00h
30e3h	Soft Lock Register (SLOCK227)	8	RW	00h
30e4h	Soft Lock Register (SLOCK228)	8	RW	00h
30e5h	Soft Lock Register (SLOCK229)	8	RW	00h
30e6h	Soft Lock Register (SLOCK230)	8	RW	00h
30e7h	Soft Lock Register (SLOCK231)	8	RW	00h
30e8h	Soft Lock Register (SLOCK232)	8	RW	00h
30e9h	Soft Lock Register (SLOCK233)	8	RW	00h
30eah	Soft Lock Register (SLOCK234)	8	RW	00h
30ebh	Soft Lock Register (SLOCK235)	8	RW	00h
30ech	Soft Lock Register (SLOCK236)	8	RW	00h
30edh	Soft Lock Register (SLOCK237)	8	RW	00h
30eeh	Soft Lock Register (SLOCK238)	8	RW	00h
30efh	Soft Lock Register (SLOCK239)	8	RW	00h
30f0h	Soft Lock Register (SLOCK240)	8	RW	00h
30f1h	Soft Lock Register (SLOCK241)	8	RW	00h
30f2h	Soft Lock Register (SLOCK242)	8	RW	00h
30f3h	Soft Lock Register (SLOCK243)	8	RW	00h
30f4h	Soft Lock Register (SLOCK244)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
30f5h	Soft Lock Register (SLOCK245)	8	RW	00h
30f6h	Soft Lock Register (SLOCK246)	8	RW	00h
30f7h	Soft Lock Register (SLOCK247)	8	RW	00h
30f8h	Soft Lock Register (SLOCK248)	8	RW	00h
30f9h	Soft Lock Register (SLOCK249)	8	RW	00h
30fah	Soft Lock Register (SLOCK250)	8	RW	00h
30fbh	Soft Lock Register (SLOCK251)	8	RW	00h
30fch	Soft Lock Register (SLOCK252)	8	RW	00h
30fdh	Soft Lock Register (SLOCK253)	8	RW	00h
30feh	Soft Lock Register (SLOCK254)	8	RW	00h
30ffh	Soft Lock Register (SLOCK255)	8	RW	00h
3100h	Soft Lock Register (SLOCK256)	8	RW	00h
3101h	Soft Lock Register (SLOCK257)	8	RW	00h
3102h	Soft Lock Register (SLOCK258)	8	RW	00h
3103h	Soft Lock Register (SLOCK259)	8	RW	00h
3104h	Soft Lock Register (SLOCK260)	8	RW	00h
3105h	Soft Lock Register (SLOCK261)	8	RW	00h
3106h	Soft Lock Register (SLOCK262)	8	RW	00h
3107h	Soft Lock Register (SLOCK263)	8	RW	00h
3108h	Soft Lock Register (SLOCK264)	8	RW	00h
3109h	Soft Lock Register (SLOCK265)	8	RW	00h
310ah	Soft Lock Register (SLOCK266)	8	RW	00h
310bh	Soft Lock Register (SLOCK267)	8	RW	00h
310ch	Soft Lock Register (SLOCK268)	8	RW	00h
310dh	Soft Lock Register (SLOCK269)	8	RW	00h
310eh	Soft Lock Register (SLOCK270)	8	RW	00h
310fh	Soft Lock Register (SLOCK271)	8	RW	00h
3110h	Soft Lock Register (SLOCK272)	8	RW	00h
3111h	Soft Lock Register (SLOCK273)	8	RW	00h
3112h	Soft Lock Register (SLOCK274)	8	RW	00h
3113h	Soft Lock Register (SLOCK275)	8	RW	00h
3114h	Soft Lock Register (SLOCK276)	8	RW	00h
3115h	Soft Lock Register (SLOCK277)	8	RW	00h
3116h	Soft Lock Register (SLOCK278)	8	RW	00h
3117h	Soft Lock Register (SLOCK279)	8	RW	00h
3118h	Soft Lock Register (SLOCK280)	8	RW	00h
3119h	Soft Lock Register (SLOCK281)	8	RW	00h
311ah	Soft Lock Register (SLOCK282)	8	RW	00h
311bh	Soft Lock Register (SLOCK283)	8	RW	00h
311ch	Soft Lock Register (SLOCK284)	8	RW	00h
311dh	Soft Lock Register (SLOCK285)	8	RW	00h
311eh	Soft Lock Register (SLOCK286)	8	RW	00h
311fh	Soft Lock Register (SLOCK287)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3120h	Soft Lock Register (SLOCK288)	8	RW	00h
3121h	Soft Lock Register (SLOCK289)	8	RW	00h
3122h	Soft Lock Register (SLOCK290)	8	RW	00h
3123h	Soft Lock Register (SLOCK291)	8	RW	00h
3124h	Soft Lock Register (SLOCK292)	8	RW	00h
3125h	Soft Lock Register (SLOCK293)	8	RW	00h
3126h	Soft Lock Register (SLOCK294)	8	RW	00h
3127h	Soft Lock Register (SLOCK295)	8	RW	00h
3128h	Soft Lock Register (SLOCK296)	8	RW	00h
3129h	Soft Lock Register (SLOCK297)	8	RW	00h
312ah	Soft Lock Register (SLOCK298)	8	RW	00h
312bh	Soft Lock Register (SLOCK299)	8	RW	00h
312ch	Soft Lock Register (SLOCK300)	8	RW	00h
312dh	Soft Lock Register (SLOCK301)	8	RW	00h
312eh	Soft Lock Register (SLOCK302)	8	RW	00h
312fh	Soft Lock Register (SLOCK303)	8	RW	00h
3130h	Soft Lock Register (SLOCK304)	8	RW	00h
3131h	Soft Lock Register (SLOCK305)	8	RW	00h
3132h	Soft Lock Register (SLOCK306)	8	RW	00h
3133h	Soft Lock Register (SLOCK307)	8	RW	00h
3134h	Soft Lock Register (SLOCK308)	8	RW	00h
3135h	Soft Lock Register (SLOCK309)	8	RW	00h
3136h	Soft Lock Register (SLOCK310)	8	RW	00h
3137h	Soft Lock Register (SLOCK311)	8	RW	00h
3138h	Soft Lock Register (SLOCK312)	8	RW	00h
3139h	Soft Lock Register (SLOCK313)	8	RW	00h
313ah	Soft Lock Register (SLOCK314)	8	RW	00h
313bh	Soft Lock Register (SLOCK315)	8	RW	00h
313ch	Soft Lock Register (SLOCK316)	8	RW	00h
313dh	Soft Lock Register (SLOCK317)	8	RW	00h
313eh	Soft Lock Register (SLOCK318)	8	RW	00h
313fh	Soft Lock Register (SLOCK319)	8	RW	00h
3140h	Soft Lock Register (SLOCK320)	8	RW	00h
3141h	Soft Lock Register (SLOCK321)	8	RW	00h
3142h	Soft Lock Register (SLOCK322)	8	RW	00h
3143h	Soft Lock Register (SLOCK323)	8	RW	00h
3144h	Soft Lock Register (SLOCK324)	8	RW	00h
3145h	Soft Lock Register (SLOCK325)	8	RW	00h
3146h	Soft Lock Register (SLOCK326)	8	RW	00h
3147h	Soft Lock Register (SLOCK327)	8	RW	00h
3148h	Soft Lock Register (SLOCK328)	8	RW	00h
3149h	Soft Lock Register (SLOCK329)	8	RW	00h
314ah	Soft Lock Register (SLOCK330)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
314bh	Soft Lock Register (SLOCK331)	8	RW	00h
314ch	Soft Lock Register (SLOCK332)	8	RW	00h
314dh	Soft Lock Register (SLOCK333)	8	RW	00h
314eh	Soft Lock Register (SLOCK334)	8	RW	00h
314fh	Soft Lock Register (SLOCK335)	8	RW	00h
3150h	Soft Lock Register (SLOCK336)	8	RW	00h
3151h	Soft Lock Register (SLOCK337)	8	RW	00h
3152h	Soft Lock Register (SLOCK338)	8	RW	00h
3153h	Soft Lock Register (SLOCK339)	8	RW	00h
3154h	Soft Lock Register (SLOCK340)	8	RW	00h
3155h	Soft Lock Register (SLOCK341)	8	RW	00h
3156h	Soft Lock Register (SLOCK342)	8	RW	00h
3157h	Soft Lock Register (SLOCK343)	8	RW	00h
3158h	Soft Lock Register (SLOCK344)	8	RW	00h
3159h	Soft Lock Register (SLOCK345)	8	RW	00h
315ah	Soft Lock Register (SLOCK346)	8	RW	00h
315bh	Soft Lock Register (SLOCK347)	8	RW	00h
315ch	Soft Lock Register (SLOCK348)	8	RW	00h
315dh	Soft Lock Register (SLOCK349)	8	RW	00h
315eh	Soft Lock Register (SLOCK350)	8	RW	00h
315fh	Soft Lock Register (SLOCK351)	8	RW	00h
3160h	Soft Lock Register (SLOCK352)	8	RW	00h
3161h	Soft Lock Register (SLOCK353)	8	RW	00h
3162h	Soft Lock Register (SLOCK354)	8	RW	00h
3163h	Soft Lock Register (SLOCK355)	8	RW	00h
3164h	Soft Lock Register (SLOCK356)	8	RW	00h
3165h	Soft Lock Register (SLOCK357)	8	RW	00h
3166h	Soft Lock Register (SLOCK358)	8	RW	00h
3167h	Soft Lock Register (SLOCK359)	8	RW	00h
3168h	Soft Lock Register (SLOCK360)	8	RW	00h
3169h	Soft Lock Register (SLOCK361)	8	RW	00h
316ah	Soft Lock Register (SLOCK362)	8	RW	00h
316bh	Soft Lock Register (SLOCK363)	8	RW	00h
316ch	Soft Lock Register (SLOCK364)	8	RW	00h
316dh	Soft Lock Register (SLOCK365)	8	RW	00h
316eh	Soft Lock Register (SLOCK366)	8	RW	00h
316fh	Soft Lock Register (SLOCK367)	8	RW	00h
3170h	Soft Lock Register (SLOCK368)	8	RW	00h
3171h	Soft Lock Register (SLOCK369)	8	RW	00h
3172h	Soft Lock Register (SLOCK370)	8	RW	00h
3173h	Soft Lock Register (SLOCK371)	8	RW	00h
3174h	Soft Lock Register (SLOCK372)	8	RW	00h
3175h	Soft Lock Register (SLOCK373)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3176h	Soft Lock Register (SLOCK374)	8	RW	00h
3177h	Soft Lock Register (SLOCK375)	8	RW	00h
3178h	Soft Lock Register (SLOCK376)	8	RW	00h
3179h	Soft Lock Register (SLOCK377)	8	RW	00h
317ah	Soft Lock Register (SLOCK378)	8	RW	00h
317bh	Soft Lock Register (SLOCK379)	8	RW	00h
317ch	Soft Lock Register (SLOCK380)	8	RW	00h
317dh	Soft Lock Register (SLOCK381)	8	RW	00h
317eh	Soft Lock Register (SLOCK382)	8	RW	00h
317fh	Soft Lock Register (SLOCK383)	8	RW	00h
3180h	Soft Lock Register (SLOCK384)	8	RW	00h
3181h	Soft Lock Register (SLOCK385)	8	RW	00h
3182h	Soft Lock Register (SLOCK386)	8	RW	00h
3183h	Soft Lock Register (SLOCK387)	8	RW	00h
3184h	Soft Lock Register (SLOCK388)	8	RW	00h
3185h	Soft Lock Register (SLOCK389)	8	RW	00h
3186h	Soft Lock Register (SLOCK390)	8	RW	00h
3187h	Soft Lock Register (SLOCK391)	8	RW	00h
3188h	Soft Lock Register (SLOCK392)	8	RW	00h
3189h	Soft Lock Register (SLOCK393)	8	RW	00h
318ah	Soft Lock Register (SLOCK394)	8	RW	00h
318bh	Soft Lock Register (SLOCK395)	8	RW	00h
318ch	Soft Lock Register (SLOCK396)	8	RW	00h
318dh	Soft Lock Register (SLOCK397)	8	RW	00h
318eh	Soft Lock Register (SLOCK398)	8	RW	00h
318fh	Soft Lock Register (SLOCK399)	8	RW	00h
3190h	Soft Lock Register (SLOCK400)	8	RW	00h
3191h	Soft Lock Register (SLOCK401)	8	RW	00h
3192h	Soft Lock Register (SLOCK402)	8	RW	00h
3193h	Soft Lock Register (SLOCK403)	8	RW	00h
3194h	Soft Lock Register (SLOCK404)	8	RW	00h
3195h	Soft Lock Register (SLOCK405)	8	RW	00h
3196h	Soft Lock Register (SLOCK406)	8	RW	00h
3197h	Soft Lock Register (SLOCK407)	8	RW	00h
3198h	Soft Lock Register (SLOCK408)	8	RW	00h
3199h	Soft Lock Register (SLOCK409)	8	RW	00h
319ah	Soft Lock Register (SLOCK410)	8	RW	00h
319bh	Soft Lock Register (SLOCK411)	8	RW	00h
319ch	Soft Lock Register (SLOCK412)	8	RW	00h
319dh	Soft Lock Register (SLOCK413)	8	RW	00h
319eh	Soft Lock Register (SLOCK414)	8	RW	00h
319fh	Soft Lock Register (SLOCK415)	8	RW	00h
31a0h	Soft Lock Register (SLOCK416)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
31a1h	Soft Lock Register (SLOCK417)	8	RW	00h
31a2h	Soft Lock Register (SLOCK418)	8	RW	00h
31a3h	Soft Lock Register (SLOCK419)	8	RW	00h
31a4h	Soft Lock Register (SLOCK420)	8	RW	00h
31a5h	Soft Lock Register (SLOCK421)	8	RW	00h
31a6h	Soft Lock Register (SLOCK422)	8	RW	00h
31a7h	Soft Lock Register (SLOCK423)	8	RW	00h
31a8h	Soft Lock Register (SLOCK424)	8	RW	00h
31a9h	Soft Lock Register (SLOCK425)	8	RW	00h
31aah	Soft Lock Register (SLOCK426)	8	RW	00h
31abh	Soft Lock Register (SLOCK427)	8	RW	00h
31ach	Soft Lock Register (SLOCK428)	8	RW	00h
31adh	Soft Lock Register (SLOCK429)	8	RW	00h
31aeh	Soft Lock Register (SLOCK430)	8	RW	00h
31afh	Soft Lock Register (SLOCK431)	8	RW	00h
31b0h	Soft Lock Register (SLOCK432)	8	RW	00h
31b1h	Soft Lock Register (SLOCK433)	8	RW	00h
31b2h	Soft Lock Register (SLOCK434)	8	RW	00h
31b3h	Soft Lock Register (SLOCK435)	8	RW	00h
31b4h	Soft Lock Register (SLOCK436)	8	RW	00h
31b5h	Soft Lock Register (SLOCK437)	8	RW	00h
31b6h	Soft Lock Register (SLOCK438)	8	RW	00h
31b7h	Soft Lock Register (SLOCK439)	8	RW	00h
31b8h	Soft Lock Register (SLOCK440)	8	RW	00h
31b9h	Soft Lock Register (SLOCK441)	8	RW	00h
31bah	Soft Lock Register (SLOCK442)	8	RW	00h
31bbh	Soft Lock Register (SLOCK443)	8	RW	00h
31bch	Soft Lock Register (SLOCK444)	8	RW	00h
31bdh	Soft Lock Register (SLOCK445)	8	RW	00h
31beh	Soft Lock Register (SLOCK446)	8	RW	00h
31bfh	Soft Lock Register (SLOCK447)	8	RW	00h
31c0h	Soft Lock Register (SLOCK448)	8	RW	00h
31c1h	Soft Lock Register (SLOCK449)	8	RW	00h
31c2h	Soft Lock Register (SLOCK450)	8	RW	00h
31c3h	Soft Lock Register (SLOCK451)	8	RW	00h
31c4h	Soft Lock Register (SLOCK452)	8	RW	00h
31c5h	Soft Lock Register (SLOCK453)	8	RW	00h
31c6h	Soft Lock Register (SLOCK454)	8	RW	00h
31c7h	Soft Lock Register (SLOCK455)	8	RW	00h
31c8h	Soft Lock Register (SLOCK456)	8	RW	00h
31c9h	Soft Lock Register (SLOCK457)	8	RW	00h
31cah	Soft Lock Register (SLOCK458)	8	RW	00h
31cbh	Soft Lock Register (SLOCK459)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
31cch	Soft Lock Register (SLOCK460)	8	RW	00h
31cdh	Soft Lock Register (SLOCK461)	8	RW	00h
31ceh	Soft Lock Register (SLOCK462)	8	RW	00h
31cfh	Soft Lock Register (SLOCK463)	8	RW	00h
31d0h	Soft Lock Register (SLOCK464)	8	RW	00h
31d1h	Soft Lock Register (SLOCK465)	8	RW	00h
31d2h	Soft Lock Register (SLOCK466)	8	RW	00h
31d3h	Soft Lock Register (SLOCK467)	8	RW	00h
31d4h	Soft Lock Register (SLOCK468)	8	RW	00h
31d5h	Soft Lock Register (SLOCK469)	8	RW	00h
31d6h	Soft Lock Register (SLOCK470)	8	RW	00h
31d7h	Soft Lock Register (SLOCK471)	8	RW	00h
31d8h	Soft Lock Register (SLOCK472)	8	RW	00h
31d9h	Soft Lock Register (SLOCK473)	8	RW	00h
31dah	Soft Lock Register (SLOCK474)	8	RW	00h
31dbh	Soft Lock Register (SLOCK475)	8	RW	00h
31dch	Soft Lock Register (SLOCK476)	8	RW	00h
31ddh	Soft Lock Register (SLOCK477)	8	RW	00h
31deh	Soft Lock Register (SLOCK478)	8	RW	00h
31dfh	Soft Lock Register (SLOCK479)	8	RW	00h
31e0h	Soft Lock Register (SLOCK480)	8	RW	00h
31e1h	Soft Lock Register (SLOCK481)	8	RW	00h
31e2h	Soft Lock Register (SLOCK482)	8	RW	00h
31e3h	Soft Lock Register (SLOCK483)	8	RW	00h
31e4h	Soft Lock Register (SLOCK484)	8	RW	00h
31e5h	Soft Lock Register (SLOCK485)	8	RW	00h
31e6h	Soft Lock Register (SLOCK486)	8	RW	00h
31e7h	Soft Lock Register (SLOCK487)	8	RW	00h
31e8h	Soft Lock Register (SLOCK488)	8	RW	00h
31e9h	Soft Lock Register (SLOCK489)	8	RW	00h
31eah	Soft Lock Register (SLOCK490)	8	RW	00h
31ebh	Soft Lock Register (SLOCK491)	8	RW	00h
31ech	Soft Lock Register (SLOCK492)	8	RW	00h
31edh	Soft Lock Register (SLOCK493)	8	RW	00h
31eeh	Soft Lock Register (SLOCK494)	8	RW	00h
31efh	Soft Lock Register (SLOCK495)	8	RW	00h
31f0h	Soft Lock Register (SLOCK496)	8	RW	00h
31f1h	Soft Lock Register (SLOCK497)	8	RW	00h
31f2h	Soft Lock Register (SLOCK498)	8	RW	00h
31f3h	Soft Lock Register (SLOCK499)	8	RW	00h
31f4h	Soft Lock Register (SLOCK500)	8	RW	00h
31f5h	Soft Lock Register (SLOCK501)	8	RW	00h
31f6h	Soft Lock Register (SLOCK502)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
31f7h	Soft Lock Register (SLOCK503)	8	RW	00h
31f8h	Soft Lock Register (SLOCK504)	8	RW	00h
31f9h	Soft Lock Register (SLOCK505)	8	RW	00h
31fah	Soft Lock Register (SLOCK506)	8	RW	00h
31fbh	Soft Lock Register (SLOCK507)	8	RW	00h
31fch	Soft Lock Register (SLOCK508)	8	RW	00h
31fdh	Soft Lock Register (SLOCK509)	8	RW	00h
31feh	Soft Lock Register (SLOCK510)	8	RW	00h
31ffh	Soft Lock Register (SLOCK511)	8	RW	00h
3200h	Soft Lock Register (SLOCK512)	8	RW	00h
3201h	Soft Lock Register (SLOCK513)	8	RW	00h
3202h	Soft Lock Register (SLOCK514)	8	RW	00h
3203h	Soft Lock Register (SLOCK515)	8	RW	00h
3204h	Soft Lock Register (SLOCK516)	8	RW	00h
3205h	Soft Lock Register (SLOCK517)	8	RW	00h
3206h	Soft Lock Register (SLOCK518)	8	RW	00h
3207h	Soft Lock Register (SLOCK519)	8	RW	00h
3208h	Soft Lock Register (SLOCK520)	8	RW	00h
3209h	Soft Lock Register (SLOCK521)	8	RW	00h
320ah	Soft Lock Register (SLOCK522)	8	RW	00h
320bh	Soft Lock Register (SLOCK523)	8	RW	00h
320ch	Soft Lock Register (SLOCK524)	8	RW	00h
320dh	Soft Lock Register (SLOCK525)	8	RW	00h
320eh	Soft Lock Register (SLOCK526)	8	RW	00h
320fh	Soft Lock Register (SLOCK527)	8	RW	00h
3210h	Soft Lock Register (SLOCK528)	8	RW	00h
3211h	Soft Lock Register (SLOCK529)	8	RW	00h
3212h	Soft Lock Register (SLOCK530)	8	RW	00h
3213h	Soft Lock Register (SLOCK531)	8	RW	00h
3214h	Soft Lock Register (SLOCK532)	8	RW	00h
3215h	Soft Lock Register (SLOCK533)	8	RW	00h
3216h	Soft Lock Register (SLOCK534)	8	RW	00h
3217h	Soft Lock Register (SLOCK535)	8	RW	00h
3218h	Soft Lock Register (SLOCK536)	8	RW	00h
3219h	Soft Lock Register (SLOCK537)	8	RW	00h
321ah	Soft Lock Register (SLOCK538)	8	RW	00h
321bh	Soft Lock Register (SLOCK539)	8	RW	00h
321ch	Soft Lock Register (SLOCK540)	8	RW	00h
321dh	Soft Lock Register (SLOCK541)	8	RW	00h
321eh	Soft Lock Register (SLOCK542)	8	RW	00h
321fh	Soft Lock Register (SLOCK543)	8	RW	00h
3220h	Soft Lock Register (SLOCK544)	8	RW	00h
3221h	Soft Lock Register (SLOCK545)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3222h	Soft Lock Register (SLOCK546)	8	RW	00h
3223h	Soft Lock Register (SLOCK547)	8	RW	00h
3224h	Soft Lock Register (SLOCK548)	8	RW	00h
3225h	Soft Lock Register (SLOCK549)	8	RW	00h
3226h	Soft Lock Register (SLOCK550)	8	RW	00h
3227h	Soft Lock Register (SLOCK551)	8	RW	00h
3228h	Soft Lock Register (SLOCK552)	8	RW	00h
3229h	Soft Lock Register (SLOCK553)	8	RW	00h
322ah	Soft Lock Register (SLOCK554)	8	RW	00h
322bh	Soft Lock Register (SLOCK555)	8	RW	00h
322ch	Soft Lock Register (SLOCK556)	8	RW	00h
322dh	Soft Lock Register (SLOCK557)	8	RW	00h
322eh	Soft Lock Register (SLOCK558)	8	RW	00h
322fh	Soft Lock Register (SLOCK559)	8	RW	00h
3230h	Soft Lock Register (SLOCK560)	8	RW	00h
3231h	Soft Lock Register (SLOCK561)	8	RW	00h
3232h	Soft Lock Register (SLOCK562)	8	RW	00h
3233h	Soft Lock Register (SLOCK563)	8	RW	00h
3234h	Soft Lock Register (SLOCK564)	8	RW	00h
3235h	Soft Lock Register (SLOCK565)	8	RW	00h
3236h	Soft Lock Register (SLOCK566)	8	RW	00h
3237h	Soft Lock Register (SLOCK567)	8	RW	00h
3238h	Soft Lock Register (SLOCK568)	8	RW	00h
3239h	Soft Lock Register (SLOCK569)	8	RW	00h
323ah	Soft Lock Register (SLOCK570)	8	RW	00h
323bh	Soft Lock Register (SLOCK571)	8	RW	00h
323ch	Soft Lock Register (SLOCK572)	8	RW	00h
323dh	Soft Lock Register (SLOCK573)	8	RW	00h
323eh	Soft Lock Register (SLOCK574)	8	RW	00h
323fh	Soft Lock Register (SLOCK575)	8	RW	00h
3240h	Soft Lock Register (SLOCK576)	8	RW	00h
3241h	Soft Lock Register (SLOCK577)	8	RW	00h
3242h	Soft Lock Register (SLOCK578)	8	RW	00h
3243h	Soft Lock Register (SLOCK579)	8	RW	00h
3244h	Soft Lock Register (SLOCK580)	8	RW	00h
3245h	Soft Lock Register (SLOCK581)	8	RW	00h
3246h	Soft Lock Register (SLOCK582)	8	RW	00h
3247h	Soft Lock Register (SLOCK583)	8	RW	00h
3248h	Soft Lock Register (SLOCK584)	8	RW	00h
3249h	Soft Lock Register (SLOCK585)	8	RW	00h
324ah	Soft Lock Register (SLOCK586)	8	RW	00h
324bh	Soft Lock Register (SLOCK587)	8	RW	00h
324ch	Soft Lock Register (SLOCK588)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
324dh	Soft Lock Register (SLOCK589)	8	RW	00h
324eh	Soft Lock Register (SLOCK590)	8	RW	00h
324fh	Soft Lock Register (SLOCK591)	8	RW	00h
3250h	Soft Lock Register (SLOCK592)	8	RW	00h
3251h	Soft Lock Register (SLOCK593)	8	RW	00h
3252h	Soft Lock Register (SLOCK594)	8	RW	00h
3253h	Soft Lock Register (SLOCK595)	8	RW	00h
3254h	Soft Lock Register (SLOCK596)	8	RW	00h
3255h	Soft Lock Register (SLOCK597)	8	RW	00h
3256h	Soft Lock Register (SLOCK598)	8	RW	00h
3257h	Soft Lock Register (SLOCK599)	8	RW	00h
3258h	Soft Lock Register (SLOCK600)	8	RW	00h
3259h	Soft Lock Register (SLOCK601)	8	RW	00h
325ah	Soft Lock Register (SLOCK602)	8	RW	00h
325bh	Soft Lock Register (SLOCK603)	8	RW	00h
325ch	Soft Lock Register (SLOCK604)	8	RW	00h
325dh	Soft Lock Register (SLOCK605)	8	RW	00h
325eh	Soft Lock Register (SLOCK606)	8	RW	00h
325fh	Soft Lock Register (SLOCK607)	8	RW	00h
3260h	Soft Lock Register (SLOCK608)	8	RW	00h
3261h	Soft Lock Register (SLOCK609)	8	RW	00h
3262h	Soft Lock Register (SLOCK610)	8	RW	00h
3263h	Soft Lock Register (SLOCK611)	8	RW	00h
3264h	Soft Lock Register (SLOCK612)	8	RW	00h
3265h	Soft Lock Register (SLOCK613)	8	RW	00h
3266h	Soft Lock Register (SLOCK614)	8	RW	00h
3267h	Soft Lock Register (SLOCK615)	8	RW	00h
3268h	Soft Lock Register (SLOCK616)	8	RW	00h
3269h	Soft Lock Register (SLOCK617)	8	RW	00h
326ah	Soft Lock Register (SLOCK618)	8	RW	00h
326bh	Soft Lock Register (SLOCK619)	8	RW	00h
326ch	Soft Lock Register (SLOCK620)	8	RW	00h
326dh	Soft Lock Register (SLOCK621)	8	RW	00h
326eh	Soft Lock Register (SLOCK622)	8	RW	00h
326fh	Soft Lock Register (SLOCK623)	8	RW	00h
3270h	Soft Lock Register (SLOCK624)	8	RW	00h
3271h	Soft Lock Register (SLOCK625)	8	RW	00h
3272h	Soft Lock Register (SLOCK626)	8	RW	00h
3273h	Soft Lock Register (SLOCK627)	8	RW	00h
3274h	Soft Lock Register (SLOCK628)	8	RW	00h
3275h	Soft Lock Register (SLOCK629)	8	RW	00h
3276h	Soft Lock Register (SLOCK630)	8	RW	00h
3277h	Soft Lock Register (SLOCK631)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3278h	Soft Lock Register (SLOCK632)	8	RW	00h
3279h	Soft Lock Register (SLOCK633)	8	RW	00h
327ah	Soft Lock Register (SLOCK634)	8	RW	00h
327bh	Soft Lock Register (SLOCK635)	8	RW	00h
327ch	Soft Lock Register (SLOCK636)	8	RW	00h
327dh	Soft Lock Register (SLOCK637)	8	RW	00h
327eh	Soft Lock Register (SLOCK638)	8	RW	00h
327fh	Soft Lock Register (SLOCK639)	8	RW	00h
3280h	Soft Lock Register (SLOCK640)	8	RW	00h
3281h	Soft Lock Register (SLOCK641)	8	RW	00h
3282h	Soft Lock Register (SLOCK642)	8	RW	00h
3283h	Soft Lock Register (SLOCK643)	8	RW	00h
3284h	Soft Lock Register (SLOCK644)	8	RW	00h
3285h	Soft Lock Register (SLOCK645)	8	RW	00h
3286h	Soft Lock Register (SLOCK646)	8	RW	00h
3287h	Soft Lock Register (SLOCK647)	8	RW	00h
3288h	Soft Lock Register (SLOCK648)	8	RW	00h
3289h	Soft Lock Register (SLOCK649)	8	RW	00h
328ah	Soft Lock Register (SLOCK650)	8	RW	00h
328bh	Soft Lock Register (SLOCK651)	8	RW	00h
328ch	Soft Lock Register (SLOCK652)	8	RW	00h
328dh	Soft Lock Register (SLOCK653)	8	RW	00h
328eh	Soft Lock Register (SLOCK654)	8	RW	00h
328fh	Soft Lock Register (SLOCK655)	8	RW	00h
3290h	Soft Lock Register (SLOCK656)	8	RW	00h
3291h	Soft Lock Register (SLOCK657)	8	RW	00h
3292h	Soft Lock Register (SLOCK658)	8	RW	00h
3293h	Soft Lock Register (SLOCK659)	8	RW	00h
3294h	Soft Lock Register (SLOCK660)	8	RW	00h
3295h	Soft Lock Register (SLOCK661)	8	RW	00h
3296h	Soft Lock Register (SLOCK662)	8	RW	00h
3297h	Soft Lock Register (SLOCK663)	8	RW	00h
3298h	Soft Lock Register (SLOCK664)	8	RW	00h
3299h	Soft Lock Register (SLOCK665)	8	RW	00h
329ah	Soft Lock Register (SLOCK666)	8	RW	00h
329bh	Soft Lock Register (SLOCK667)	8	RW	00h
329ch	Soft Lock Register (SLOCK668)	8	RW	00h
329dh	Soft Lock Register (SLOCK669)	8	RW	00h
329eh	Soft Lock Register (SLOCK670)	8	RW	00h
329fh	Soft Lock Register (SLOCK671)	8	RW	00h
32a0h	Soft Lock Register (SLOCK672)	8	RW	00h
32a1h	Soft Lock Register (SLOCK673)	8	RW	00h
32a2h	Soft Lock Register (SLOCK674)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
32a3h	Soft Lock Register (SLOCK675)	8	RW	00h
32a4h	Soft Lock Register (SLOCK676)	8	RW	00h
32a5h	Soft Lock Register (SLOCK677)	8	RW	00h
32a6h	Soft Lock Register (SLOCK678)	8	RW	00h
32a7h	Soft Lock Register (SLOCK679)	8	RW	00h
32a8h	Soft Lock Register (SLOCK680)	8	RW	00h
32a9h	Soft Lock Register (SLOCK681)	8	RW	00h
32aah	Soft Lock Register (SLOCK682)	8	RW	00h
32abh	Soft Lock Register (SLOCK683)	8	RW	00h
32ach	Soft Lock Register (SLOCK684)	8	RW	00h
32adh	Soft Lock Register (SLOCK685)	8	RW	00h
32aeh	Soft Lock Register (SLOCK686)	8	RW	00h
32afh	Soft Lock Register (SLOCK687)	8	RW	00h
32b0h	Soft Lock Register (SLOCK688)	8	RW	00h
32b1h	Soft Lock Register (SLOCK689)	8	RW	00h
32b2h	Soft Lock Register (SLOCK690)	8	RW	00h
32b3h	Soft Lock Register (SLOCK691)	8	RW	00h
32b4h	Soft Lock Register (SLOCK692)	8	RW	00h
32b5h	Soft Lock Register (SLOCK693)	8	RW	00h
32b6h	Soft Lock Register (SLOCK694)	8	RW	00h
32b7h	Soft Lock Register (SLOCK695)	8	RW	00h
32b8h	Soft Lock Register (SLOCK696)	8	RW	00h
32b9h	Soft Lock Register (SLOCK697)	8	RW	00h
32bah	Soft Lock Register (SLOCK698)	8	RW	00h
32bbh	Soft Lock Register (SLOCK699)	8	RW	00h
32bch	Soft Lock Register (SLOCK700)	8	RW	00h
32bdh	Soft Lock Register (SLOCK701)	8	RW	00h
32beh	Soft Lock Register (SLOCK702)	8	RW	00h
32bfh	Soft Lock Register (SLOCK703)	8	RW	00h
32c0h	Soft Lock Register (SLOCK704)	8	RW	00h
32c1h	Soft Lock Register (SLOCK705)	8	RW	00h
32c2h	Soft Lock Register (SLOCK706)	8	RW	00h
32c3h	Soft Lock Register (SLOCK707)	8	RW	00h
32c4h	Soft Lock Register (SLOCK708)	8	RW	00h
32c5h	Soft Lock Register (SLOCK709)	8	RW	00h
32c6h	Soft Lock Register (SLOCK710)	8	RW	00h
32c7h	Soft Lock Register (SLOCK711)	8	RW	00h
32c8h	Soft Lock Register (SLOCK712)	8	RW	00h
32c9h	Soft Lock Register (SLOCK713)	8	RW	00h
32cah	Soft Lock Register (SLOCK714)	8	RW	00h
32cbh	Soft Lock Register (SLOCK715)	8	RW	00h
32cch	Soft Lock Register (SLOCK716)	8	RW	00h
32cdh	Soft Lock Register (SLOCK717)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
32ceh	Soft Lock Register (SLOCK718)	8	RW	00h
32cfh	Soft Lock Register (SLOCK719)	8	RW	00h
32d0h	Soft Lock Register (SLOCK720)	8	RW	00h
32d1h	Soft Lock Register (SLOCK721)	8	RW	00h
32d2h	Soft Lock Register (SLOCK722)	8	RW	00h
32d3h	Soft Lock Register (SLOCK723)	8	RW	00h
32d4h	Soft Lock Register (SLOCK724)	8	RW	00h
32d5h	Soft Lock Register (SLOCK725)	8	RW	00h
32d6h	Soft Lock Register (SLOCK726)	8	RW	00h
32d7h	Soft Lock Register (SLOCK727)	8	RW	00h
32d8h	Soft Lock Register (SLOCK728)	8	RW	00h
32d9h	Soft Lock Register (SLOCK729)	8	RW	00h
32dah	Soft Lock Register (SLOCK730)	8	RW	00h
32dbh	Soft Lock Register (SLOCK731)	8	RW	00h
32dch	Soft Lock Register (SLOCK732)	8	RW	00h
32ddh	Soft Lock Register (SLOCK733)	8	RW	00h
32deh	Soft Lock Register (SLOCK734)	8	RW	00h
32dfh	Soft Lock Register (SLOCK735)	8	RW	00h
32e0h	Soft Lock Register (SLOCK736)	8	RW	00h
32e1h	Soft Lock Register (SLOCK737)	8	RW	00h
32e2h	Soft Lock Register (SLOCK738)	8	RW	00h
32e3h	Soft Lock Register (SLOCK739)	8	RW	00h
32e4h	Soft Lock Register (SLOCK740)	8	RW	00h
32e5h	Soft Lock Register (SLOCK741)	8	RW	00h
32e6h	Soft Lock Register (SLOCK742)	8	RW	00h
32e7h	Soft Lock Register (SLOCK743)	8	RW	00h
32e8h	Soft Lock Register (SLOCK744)	8	RW	00h
32e9h	Soft Lock Register (SLOCK745)	8	RW	00h
32eah	Soft Lock Register (SLOCK746)	8	RW	00h
32ebh	Soft Lock Register (SLOCK747)	8	RW	00h
32ech	Soft Lock Register (SLOCK748)	8	RW	00h
32edh	Soft Lock Register (SLOCK749)	8	RW	00h
32eeh	Soft Lock Register (SLOCK750)	8	RW	00h
32efh	Soft Lock Register (SLOCK751)	8	RW	00h
32f0h	Soft Lock Register (SLOCK752)	8	RW	00h
32f1h	Soft Lock Register (SLOCK753)	8	RW	00h
32f2h	Soft Lock Register (SLOCK754)	8	RW	00h
32f3h	Soft Lock Register (SLOCK755)	8	RW	00h
32f4h	Soft Lock Register (SLOCK756)	8	RW	00h
32f5h	Soft Lock Register (SLOCK757)	8	RW	00h
32f6h	Soft Lock Register (SLOCK758)	8	RW	00h
32f7h	Soft Lock Register (SLOCK759)	8	RW	00h
32f8h	Soft Lock Register (SLOCK760)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
32f9h	Soft Lock Register (SLOCK761)	8	RW	00h
32fah	Soft Lock Register (SLOCK762)	8	RW	00h
32fbh	Soft Lock Register (SLOCK763)	8	RW	00h
32fch	Soft Lock Register (SLOCK764)	8	RW	00h
32fdh	Soft Lock Register (SLOCK765)	8	RW	00h
32feh	Soft Lock Register (SLOCK766)	8	RW	00h
32ffh	Soft Lock Register (SLOCK767)	8	RW	00h
3300h	Soft Lock Register (SLOCK768)	8	RW	00h
3301h	Soft Lock Register (SLOCK769)	8	RW	00h
3302h	Soft Lock Register (SLOCK770)	8	RW	00h
3303h	Soft Lock Register (SLOCK771)	8	RW	00h
3304h	Soft Lock Register (SLOCK772)	8	RW	00h
3305h	Soft Lock Register (SLOCK773)	8	RW	00h
3306h	Soft Lock Register (SLOCK774)	8	RW	00h
3307h	Soft Lock Register (SLOCK775)	8	RW	00h
3308h	Soft Lock Register (SLOCK776)	8	RW	00h
3309h	Soft Lock Register (SLOCK777)	8	RW	00h
330ah	Soft Lock Register (SLOCK778)	8	RW	00h
330bh	Soft Lock Register (SLOCK779)	8	RW	00h
330ch	Soft Lock Register (SLOCK780)	8	RW	00h
330dh	Soft Lock Register (SLOCK781)	8	RW	00h
330eh	Soft Lock Register (SLOCK782)	8	RW	00h
330fh	Soft Lock Register (SLOCK783)	8	RW	00h
3310h	Soft Lock Register (SLOCK784)	8	RW	00h
3311h	Soft Lock Register (SLOCK785)	8	RW	00h
3312h	Soft Lock Register (SLOCK786)	8	RW	00h
3313h	Soft Lock Register (SLOCK787)	8	RW	00h
3314h	Soft Lock Register (SLOCK788)	8	RW	00h
3315h	Soft Lock Register (SLOCK789)	8	RW	00h
3316h	Soft Lock Register (SLOCK790)	8	RW	00h
3317h	Soft Lock Register (SLOCK791)	8	RW	00h
3318h	Soft Lock Register (SLOCK792)	8	RW	00h
3319h	Soft Lock Register (SLOCK793)	8	RW	00h
331ah	Soft Lock Register (SLOCK794)	8	RW	00h
331bh	Soft Lock Register (SLOCK795)	8	RW	00h
331ch	Soft Lock Register (SLOCK796)	8	RW	00h
331dh	Soft Lock Register (SLOCK797)	8	RW	00h
331eh	Soft Lock Register (SLOCK798)	8	RW	00h
331fh	Soft Lock Register (SLOCK799)	8	RW	00h
3320h	Soft Lock Register (SLOCK800)	8	RW	00h
3321h	Soft Lock Register (SLOCK801)	8	RW	00h
3322h	Soft Lock Register (SLOCK802)	8	RW	00h
3323h	Soft Lock Register (SLOCK803)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
3324h	Soft Lock Register (SLOCK804)	8	RW	00h
3325h	Soft Lock Register (SLOCK805)	8	RW	00h
3326h	Soft Lock Register (SLOCK806)	8	RW	00h
3327h	Soft Lock Register (SLOCK807)	8	RW	00h
3328h	Soft Lock Register (SLOCK808)	8	RW	00h
3329h	Soft Lock Register (SLOCK809)	8	RW	00h
332ah	Soft Lock Register (SLOCK810)	8	RW	00h
332bh	Soft Lock Register (SLOCK811)	8	RW	00h
332ch	Soft Lock Register (SLOCK812)	8	RW	00h
332dh	Soft Lock Register (SLOCK813)	8	RW	00h
332eh	Soft Lock Register (SLOCK814)	8	RW	00h
332fh	Soft Lock Register (SLOCK815)	8	RW	00h
3330h	Soft Lock Register (SLOCK816)	8	RW	00h
3331h	Soft Lock Register (SLOCK817)	8	RW	00h
3332h	Soft Lock Register (SLOCK818)	8	RW	00h
3333h	Soft Lock Register (SLOCK819)	8	RW	00h
3334h	Soft Lock Register (SLOCK820)	8	RW	00h
3335h	Soft Lock Register (SLOCK821)	8	RW	00h
3336h	Soft Lock Register (SLOCK822)	8	RW	00h
3337h	Soft Lock Register (SLOCK823)	8	RW	00h
3338h	Soft Lock Register (SLOCK824)	8	RW	00h
3339h	Soft Lock Register (SLOCK825)	8	RW	00h
333ah	Soft Lock Register (SLOCK826)	8	RW	00h
333bh	Soft Lock Register (SLOCK827)	8	RW	00h
333ch	Soft Lock Register (SLOCK828)	8	RW	00h
333dh	Soft Lock Register (SLOCK829)	8	RW	00h
333eh	Soft Lock Register (SLOCK830)	8	RW	00h
333fh	Soft Lock Register (SLOCK831)	8	RW	00h
3340h	Soft Lock Register (SLOCK832)	8	RW	00h
3341h	Soft Lock Register (SLOCK833)	8	RW	00h
3342h	Soft Lock Register (SLOCK834)	8	RW	00h
3343h	Soft Lock Register (SLOCK835)	8	RW	00h
3344h	Soft Lock Register (SLOCK836)	8	RW	00h
3345h	Soft Lock Register (SLOCK837)	8	RW	00h
3346h	Soft Lock Register (SLOCK838)	8	RW	00h
3347h	Soft Lock Register (SLOCK839)	8	RW	00h
3348h	Soft Lock Register (SLOCK840)	8	RW	00h
3349h	Soft Lock Register (SLOCK841)	8	RW	00h
334ah	Soft Lock Register (SLOCK842)	8	RW	00h
334bh	Soft Lock Register (SLOCK843)	8	RW	00h
334ch	Soft Lock Register (SLOCK844)	8	RW	00h
334dh	Soft Lock Register (SLOCK845)	8	RW	00h
334eh	Soft Lock Register (SLOCK846)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
334fh	Soft Lock Register (SLOCK847)	8	RW	00h
3350h	Soft Lock Register (SLOCK848)	8	RW	00h
3351h	Soft Lock Register (SLOCK849)	8	RW	00h
3352h	Soft Lock Register (SLOCK850)	8	RW	00h
3353h	Soft Lock Register (SLOCK851)	8	RW	00h
3354h	Soft Lock Register (SLOCK852)	8	RW	00h
3355h	Soft Lock Register (SLOCK853)	8	RW	00h
3356h	Soft Lock Register (SLOCK854)	8	RW	00h
3357h	Soft Lock Register (SLOCK855)	8	RW	00h
3358h	Soft Lock Register (SLOCK856)	8	RW	00h
3359h	Soft Lock Register (SLOCK857)	8	RW	00h
335ah	Soft Lock Register (SLOCK858)	8	RW	00h
335bh	Soft Lock Register (SLOCK859)	8	RW	00h
335ch	Soft Lock Register (SLOCK860)	8	RW	00h
335dh	Soft Lock Register (SLOCK861)	8	RW	00h
335eh	Soft Lock Register (SLOCK862)	8	RW	00h
335fh	Soft Lock Register (SLOCK863)	8	RW	00h
3360h	Soft Lock Register (SLOCK864)	8	RW	00h
3361h	Soft Lock Register (SLOCK865)	8	RW	00h
3362h	Soft Lock Register (SLOCK866)	8	RW	00h
3363h	Soft Lock Register (SLOCK867)	8	RW	00h
3364h	Soft Lock Register (SLOCK868)	8	RW	00h
3365h	Soft Lock Register (SLOCK869)	8	RW	00h
3366h	Soft Lock Register (SLOCK870)	8	RW	00h
3367h	Soft Lock Register (SLOCK871)	8	RW	00h
3368h	Soft Lock Register (SLOCK872)	8	RW	00h
3369h	Soft Lock Register (SLOCK873)	8	RW	00h
336ah	Soft Lock Register (SLOCK874)	8	RW	00h
336bh	Soft Lock Register (SLOCK875)	8	RW	00h
336ch	Soft Lock Register (SLOCK876)	8	RW	00h
336dh	Soft Lock Register (SLOCK877)	8	RW	00h
336eh	Soft Lock Register (SLOCK878)	8	RW	00h
336fh	Soft Lock Register (SLOCK879)	8	RW	00h
3370h	Soft Lock Register (SLOCK880)	8	RW	00h
3371h	Soft Lock Register (SLOCK881)	8	RW	00h
3372h	Soft Lock Register (SLOCK882)	8	RW	00h
3373h	Soft Lock Register (SLOCK883)	8	RW	00h
3374h	Soft Lock Register (SLOCK884)	8	RW	00h
3375h	Soft Lock Register (SLOCK885)	8	RW	00h
3376h	Soft Lock Register (SLOCK886)	8	RW	00h
3377h	Soft Lock Register (SLOCK887)	8	RW	00h
3378h	Soft Lock Register (SLOCK888)	8	RW	00h
3379h	Soft Lock Register (SLOCK889)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
337ah	Soft Lock Register (SLOCK890)	8	RW	00h
337bh	Soft Lock Register (SLOCK891)	8	RW	00h
337ch	Soft Lock Register (SLOCK892)	8	RW	00h
337dh	Soft Lock Register (SLOCK893)	8	RW	00h
337eh	Soft Lock Register (SLOCK894)	8	RW	00h
337fh	Soft Lock Register (SLOCK895)	8	RW	00h
3380h	Soft Lock Register (SLOCK896)	8	RW	00h
3381h	Soft Lock Register (SLOCK897)	8	RW	00h
3382h	Soft Lock Register (SLOCK898)	8	RW	00h
3383h	Soft Lock Register (SLOCK899)	8	RW	00h
3384h	Soft Lock Register (SLOCK900)	8	RW	00h
3385h	Soft Lock Register (SLOCK901)	8	RW	00h
3386h	Soft Lock Register (SLOCK902)	8	RW	00h
3387h	Soft Lock Register (SLOCK903)	8	RW	00h
3388h	Soft Lock Register (SLOCK904)	8	RW	00h
3389h	Soft Lock Register (SLOCK905)	8	RW	00h
338ah	Soft Lock Register (SLOCK906)	8	RW	00h
338bh	Soft Lock Register (SLOCK907)	8	RW	00h
338ch	Soft Lock Register (SLOCK908)	8	RW	00h
338dh	Soft Lock Register (SLOCK909)	8	RW	00h
338eh	Soft Lock Register (SLOCK910)	8	RW	00h
338fh	Soft Lock Register (SLOCK911)	8	RW	00h
3390h	Soft Lock Register (SLOCK912)	8	RW	00h
3391h	Soft Lock Register (SLOCK913)	8	RW	00h
3392h	Soft Lock Register (SLOCK914)	8	RW	00h
3393h	Soft Lock Register (SLOCK915)	8	RW	00h
3394h	Soft Lock Register (SLOCK916)	8	RW	00h
3395h	Soft Lock Register (SLOCK917)	8	RW	00h
3396h	Soft Lock Register (SLOCK918)	8	RW	00h
3397h	Soft Lock Register (SLOCK919)	8	RW	00h
3398h	Soft Lock Register (SLOCK920)	8	RW	00h
3399h	Soft Lock Register (SLOCK921)	8	RW	00h
339ah	Soft Lock Register (SLOCK922)	8	RW	00h
339bh	Soft Lock Register (SLOCK923)	8	RW	00h
339ch	Soft Lock Register (SLOCK924)	8	RW	00h
339dh	Soft Lock Register (SLOCK925)	8	RW	00h
339eh	Soft Lock Register (SLOCK926)	8	RW	00h
339fh	Soft Lock Register (SLOCK927)	8	RW	00h
33a0h	Soft Lock Register (SLOCK928)	8	RW	00h
33a1h	Soft Lock Register (SLOCK929)	8	RW	00h
33a2h	Soft Lock Register (SLOCK930)	8	RW	00h
33a3h	Soft Lock Register (SLOCK931)	8	RW	00h
33a4h	Soft Lock Register (SLOCK932)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
33a5h	Soft Lock Register (SLOCK933)	8	RW	00h
33a6h	Soft Lock Register (SLOCK934)	8	RW	00h
33a7h	Soft Lock Register (SLOCK935)	8	RW	00h
33a8h	Soft Lock Register (SLOCK936)	8	RW	00h
33a9h	Soft Lock Register (SLOCK937)	8	RW	00h
33aah	Soft Lock Register (SLOCK938)	8	RW	00h
33abh	Soft Lock Register (SLOCK939)	8	RW	00h
33ach	Soft Lock Register (SLOCK940)	8	RW	00h
33adh	Soft Lock Register (SLOCK941)	8	RW	00h
33aeh	Soft Lock Register (SLOCK942)	8	RW	00h
33afh	Soft Lock Register (SLOCK943)	8	RW	00h
33b0h	Soft Lock Register (SLOCK944)	8	RW	00h
33b1h	Soft Lock Register (SLOCK945)	8	RW	00h
33b2h	Soft Lock Register (SLOCK946)	8	RW	00h
33b3h	Soft Lock Register (SLOCK947)	8	RW	00h
33b4h	Soft Lock Register (SLOCK948)	8	RW	00h
33b5h	Soft Lock Register (SLOCK949)	8	RW	00h
33b6h	Soft Lock Register (SLOCK950)	8	RW	00h
33b7h	Soft Lock Register (SLOCK951)	8	RW	00h
33b8h	Soft Lock Register (SLOCK952)	8	RW	00h
33b9h	Soft Lock Register (SLOCK953)	8	RW	00h
33bah	Soft Lock Register (SLOCK954)	8	RW	00h
33bbh	Soft Lock Register (SLOCK955)	8	RW	00h
33bch	Soft Lock Register (SLOCK956)	8	RW	00h
33bdh	Soft Lock Register (SLOCK957)	8	RW	00h
33beh	Soft Lock Register (SLOCK958)	8	RW	00h
33bfh	Soft Lock Register (SLOCK959)	8	RW	00h
33c0h	Soft Lock Register (SLOCK960)	8	RW	00h
33c1h	Soft Lock Register (SLOCK961)	8	RW	00h
33c2h	Soft Lock Register (SLOCK962)	8	RW	00h
33c3h	Soft Lock Register (SLOCK963)	8	RW	00h
33c4h	Soft Lock Register (SLOCK964)	8	RW	00h
33c5h	Soft Lock Register (SLOCK965)	8	RW	00h
33c6h	Soft Lock Register (SLOCK966)	8	RW	00h
33c7h	Soft Lock Register (SLOCK967)	8	RW	00h
33c8h	Soft Lock Register (SLOCK968)	8	RW	00h
33c9h	Soft Lock Register (SLOCK969)	8	RW	00h
33cah	Soft Lock Register (SLOCK970)	8	RW	00h
33cbh	Soft Lock Register (SLOCK971)	8	RW	00h
33cch	Soft Lock Register (SLOCK972)	8	RW	00h
33cdh	Soft Lock Register (SLOCK973)	8	RW	00h
33ceh	Soft Lock Register (SLOCK974)	8	RW	00h
33cfh	Soft Lock Register (SLOCK975)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
33d0h	Soft Lock Register (SLOCK976)	8	RW	00h
33d1h	Soft Lock Register (SLOCK977)	8	RW	00h
33d2h	Soft Lock Register (SLOCK978)	8	RW	00h
33d3h	Soft Lock Register (SLOCK979)	8	RW	00h
33d4h	Soft Lock Register (SLOCK980)	8	RW	00h
33d5h	Soft Lock Register (SLOCK981)	8	RW	00h
33d6h	Soft Lock Register (SLOCK982)	8	RW	00h
33d7h	Soft Lock Register (SLOCK983)	8	RW	00h
33d8h	Soft Lock Register (SLOCK984)	8	RW	00h
33d9h	Soft Lock Register (SLOCK985)	8	RW	00h
33dah	Soft Lock Register (SLOCK986)	8	RW	00h
33dbh	Soft Lock Register (SLOCK987)	8	RW	00h
33dch	Soft Lock Register (SLOCK988)	8	RW	00h
33ddh	Soft Lock Register (SLOCK989)	8	RW	00h
33deh	Soft Lock Register (SLOCK990)	8	RW	00h
33dfh	Soft Lock Register (SLOCK991)	8	RW	00h
33e0h	Soft Lock Register (SLOCK992)	8	RW	00h
33e1h	Soft Lock Register (SLOCK993)	8	RW	00h
33e2h	Soft Lock Register (SLOCK994)	8	RW	00h
33e3h	Soft Lock Register (SLOCK995)	8	RW	00h
33e4h	Soft Lock Register (SLOCK996)	8	RW	00h
33e5h	Soft Lock Register (SLOCK997)	8	RW	00h
33e6h	Soft Lock Register (SLOCK998)	8	RW	00h
33e7h	Soft Lock Register (SLOCK999)	8	RW	00h
33e8h	Soft Lock Register (SLOCK1000)	8	RW	00h
33e9h	Soft Lock Register (SLOCK1001)	8	RW	00h
33eah	Soft Lock Register (SLOCK1002)	8	RW	00h
33ebh	Soft Lock Register (SLOCK1003)	8	RW	00h
33ech	Soft Lock Register (SLOCK1004)	8	RW	00h
33edh	Soft Lock Register (SLOCK1005)	8	RW	00h
33eeh	Soft Lock Register (SLOCK1006)	8	RW	00h
33efh	Soft Lock Register (SLOCK1007)	8	RW	00h
33f0h	Soft Lock Register (SLOCK1008)	8	RW	00h
33f1h	Soft Lock Register (SLOCK1009)	8	RW	00h
33f2h	Soft Lock Register (SLOCK1010)	8	RW	00h
33f3h	Soft Lock Register (SLOCK1011)	8	RW	00h
33f4h	Soft Lock Register (SLOCK1012)	8	RW	00h
33f5h	Soft Lock Register (SLOCK1013)	8	RW	00h
33f6h	Soft Lock Register (SLOCK1014)	8	RW	00h
33f7h	Soft Lock Register (SLOCK1015)	8	RW	00h
33f8h	Soft Lock Register (SLOCK1016)	8	RW	00h
33f9h	Soft Lock Register (SLOCK1017)	8	RW	00h
33fah	Soft Lock Register (SLOCK1018)	8	RW	00h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
33fbh	Soft Lock Register (SLOCK1019)	8	RW	00h
33fch	Soft Lock Register (SLOCK1020)	8	RW	00h
33fdh	Soft Lock Register (SLOCK1021)	8	RW	00h
33feh	Soft Lock Register (SLOCK1022)	8	RW	00h
33ffh	Soft Lock Register (SLOCK1023)	8	RW	00h
3ffch	Protect Configuration Register (PROTCFG)	32	RW	00000000h

43.2.1.1 PPU SLOCK Register

Offset: 3000h + n * 1h

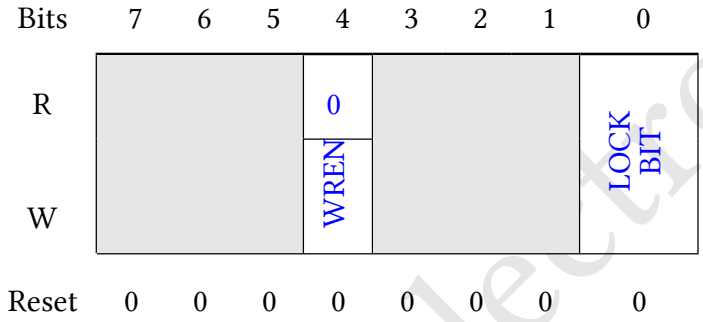


Table 43.4: PPU SLOCK Register Description

Field	Function
4 WREN	**Write Enable** The WREN bit indicates whether the LOCKBIT could be modified. User could set WREN to 1 to make the LOCKBIT to be writeable by software. 0b - LOCKBIT could not be modified 1b - LOCKBIT could be modified
0 LOCKBIT	**Lock Bit** The LOCKBIT bit indicates whether the associated 32-bit register is locked. SLOCK is associated to the 32-bit register that address offset is n*4. 0b - Associated 32-bit register is unprotected and writeable 1b - Associated 32-bit register is locked and not able to write

43.2.1.2 PPU PROTCFG Register

Offset: 3ffch

Yuntu Microelectronics

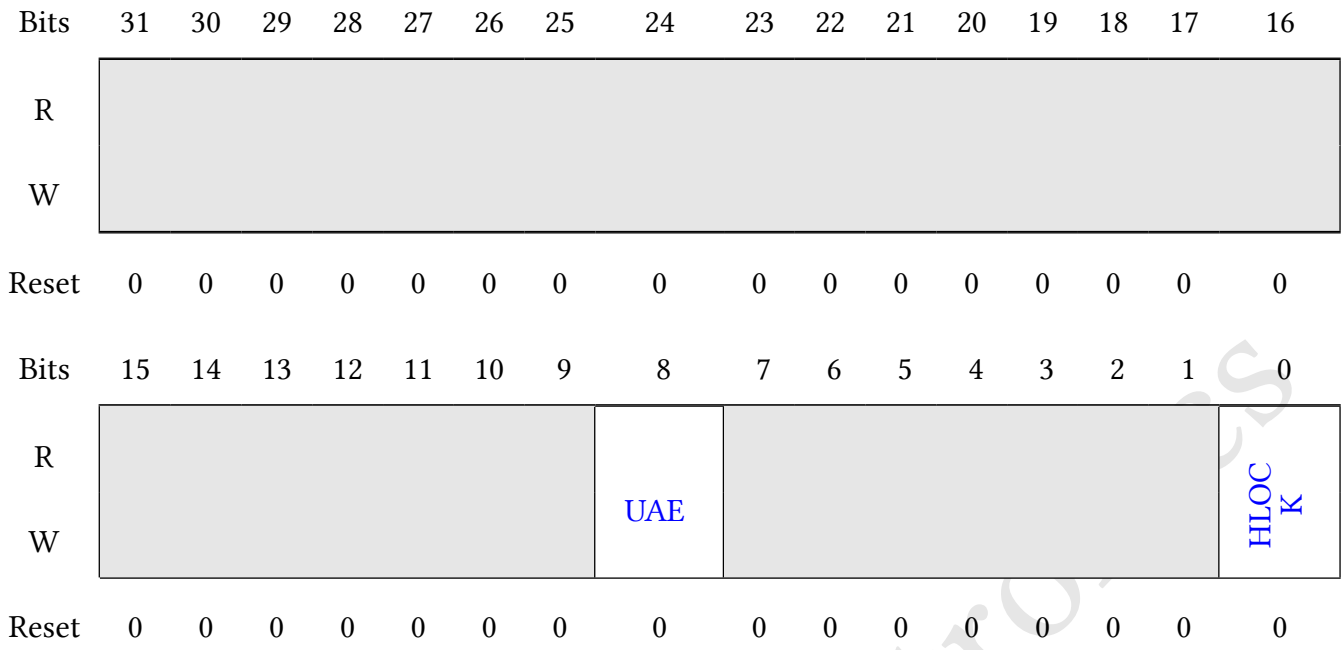


Table 43.5: PPU PROTCFG Register Description

Field	Function
8 UAE	**User Access Enable** The registers in the module under protection can be written in the supervisor or user mode. 0b - The registers in the module under protection can be only written in the supervisor. 1b - The registers in the module under protection can be written in the supervisor and user mode.
0 HLOCK	**Hard Lock** The HLOCK bit indicates whether all LOCKBITs could be modified. If HLOCK is set, the LOCKBIT is not able to be modified whatever the value of WREN is. 0b - Writeable of all LOCKBITs depend on corresponding WREN bit 1b - ALL LOCKBITs could not be writeable whatever the value of WREN bit is

43.3 Functional Description

43.3.1 Soft Lock

There are 2 ways to make registers to be protected.

1. Write access to mirror register(32-bit register) will set corresponding SLOCK[LOCKBIT] in soft lock register(8-bit register) to be 1, then this mirror register and corresponding normal register will be protected.

2. Write access to soft lock register(8-bit register) and set SLOCK[LOCKBIT] to be 1 will make the corresponding normal and mirror register(32-bit register) to be protected.

Yuntu Microelectronics

Once LOCKBIT is set, any write access to the corresponding normal and mirror register will cause transfer error.

43.3.2 Hard Lock

The PROTCFG[HLOCK] bit is used to decide whether all the mirror register and soft lock register could be written.

Once enabled, any write access to mirror register and soft lock register will cause transfer error.

43.3.3 User Access Enable

The PROTCFG[UAE] bit is used to decide whether the registers that under protection can be written in the user mode.

Once disabled, any write access to the registers will cause transfer error.

Yuntu Microelectronics

Chapter 44

Fault Management Unit(FMU)

44.1 Chip-specific Information

44.1.1 FMU Fault Configuration

YTM32B1HA0x supports 36 fault channels from safety-relevant functions on the device. For the fault sources please refer to the table below.

Table 44.1: FMU Fault Source Table

No.	Fault Source	Module	Software Recovery
0	Core lockup	CM7 Core	No
1	Core lockstep fault 0	CM7 Core	Yes
2	Core lockstep fault 1	CM7 Core	Yes
3	Core I-Cache tag fatal error	CM7 Core	Yes
4	Core I-Cache data fatal error	CM7 Core	Yes
5	Core D-Cache tag fatal error	CM7 Core	Yes
6	Core D-Cache data fatal error	CM7 Core	Yes
7	ITCM ECC fatal error	ITCM	Yes
8	D0TCM ECC fatal error	D0TCM	Yes
9	D1TCM ECC fatal error	D1TCM	Yes
10	Accidental backdoor access	Platform	Yes
11	Debug activation error	CM7 Core	No
12	OCRAML ECC multi-bit error	OCRAM	Yes
13	OCRAMU ECC multi-bit error	OCRAM	Yes
14	Flash ECC unrecoverable error	Flash	Yes
15	ROM ECC unrecoverable error	ROM	Yes
16	-	-	-
17	-	-	-
18	-	-	-
19	-	-	-
20	-	-	-
21	-	-	-
22	-	-	-
23	-	-	-
24	-	-	-

Yuntu Microelectronics

Table 44.1 continued from previous page

No.	Fault Source	Module	Software Recovery
25	-	-	-
26	-	-	-
27	High voltage detect	PCU	No
28	Interrupt monitor error	INTM	Yes
29	Clock monitor interrupt 0	CMU0	No
30	Clock monitor interrupt 1	CMU1	No
31	Clock monitor interrupt 2	CMU2	No
32	Clock monitor interrupt 3	CMU3	No
33	Watch dog timeout interrupt	WDG	No
34	Test modes activation error	TEST	No
35	Software fault	CIM	Yes

44.1.2 FMU Clocking

FMU has following clocks for FMU register access, fault source process and reaction, FMU fail to react detection.

NOTE. FMU clock configuration should be done when FMU function is static.

Table 44.2: FMU Clocking Table

Clock Name	Usage	Sub Module	Clock source
PCLK	Register access	REG	IPC, PCLK_FMU
FUNC_CLK	FMU main functional clock	FSM, FAULT_SYNC	SIRC_DIV4, FIRC, SXOSC, FXOSC, PLL
FFTR_CLK	Clock used to FMU fail to react detection	FFTR	SIRC_DIV4, FIRC, SXOSC, FXOSC, PLL

44.2 Introduction

44.2.1 Overview

The Fault Management Unit (FMU) collects the faults on the device or from an external device via IO. If a fault occurs, the faulted channel can trigger an alarm interrupt, an NMI (Non-Maskable Interrupt), or a reset. Furthermore, the FMU can also output a fault indication to the external device. With the help of FMU, the system becomes more robust to detect errors and respond in time, placing the device in a safe state.

44.2.2 Features

Features of the FMU module include:

- Support up to 36 fault channels from safety-relevant functions on the device
- Configurable FMU clock sources
  - SIRC_DIV4

Yuntu Microelectronics

  - FIRC
  - SXOSC
  - FXOSC
  - PLL
- Software fault recovery management
- Fault injection
- Lockable fault configuration
  - Changes are possible in the CFG state only
  - Configuration changes observed by CFG counter
- Configurable fault control
- Support up to 4 fault inputs via IO (FMU_ERR_IN function)
- Support 2 outputs for fault indication via IO (FMU_ERR_OUT function)
  - Configurable output window
  - Configurable output polarity
- Configurable chip reactions for each fault
  - Alarm interruption (for ALARM state)
  - FMU functional reset
  - NMI
  - No reaction
- Configurable chip reactions for fault input(FMU_ERR_IN)
  - FMU functional reset
  - NMI
- Support FMU Fail To React (FFTR) detection
  - Configurable FFTR counter timeout threshold
  - Support FFTR event reset chip

Yuntu Microelectronics

44.2.3 Block diagram

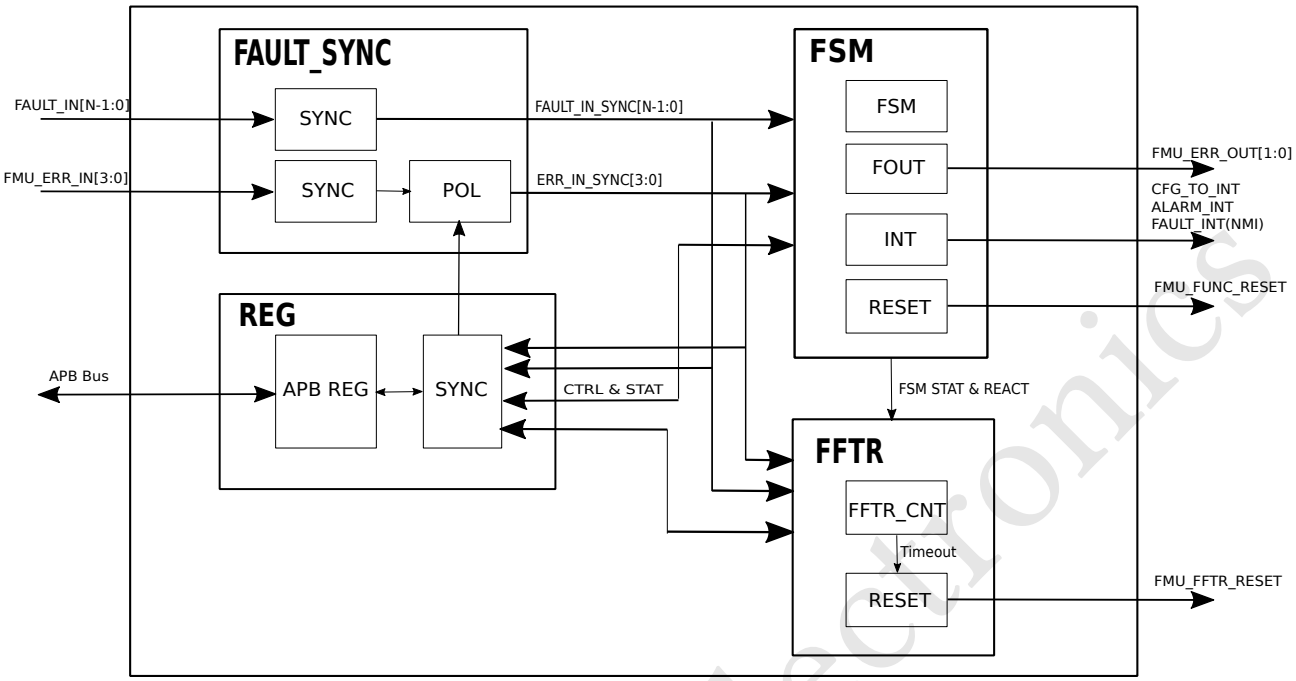


Figure 44.1: Block Diagram of FMU

44.3 Register Definition

The FMU only allows 32-bit accesses, using non-32-bit access sizes will result a bus error.

The register bits can be reset by different reset sources.

- If no specific description about register reset, the register bit is reset by any reset, including Power-on Reset, System Reset and IPC software reset if IPC module exists in the document.
- If the register bit is described as “reset by System Reset”, the register bit can be reset by System Reset and Power-on Reset, and cannot be reset by IPC[SWR].
- If the register bit is described as “reset by Power-on Reset”, the register bit can be reset by Power-on Reset only, and cannot be reset by System Reset and IPC[SWR].
- If the register bit is described as “not able to be reset”, the register bit can be reset by any reset, which means those register bits have no reset value after any reset.

44.3.1 FMU Memory Map

FMU0 base address 401c0000h

Table 44.3: FMU Memory Map

Offset	Register	Width	Access	Reset Value
0h	Lock Protection Register (LOCK)	32	RW	00000000h
4h	Operation Register (OP)	32	RW	00000000h
8h	Control Register (CTRL)	32	RW	00010005h
ch	Status Register (ST)	32	RW	00001100h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
10h	CFG State Counter (CFG_CNT)	32	RO	00000000h
14h	FAULT Status Register (FAULT_ST0)	32	RW	00000000h
18h	FAULT Status Register (FAULT_ST1)	32	RW	00000000h
34h	FAULT_ST Write Protection Key (FAULT_ST_KEY)	32	RW	see description
38h	FAULT FSM Status (FAULT_FSM_ST)	32	RO	00000000h
3ch	ALARM Counter Timeout Threshold (ALARM_CNT_TO_TH)	32	RW	0003A980h
40h	ALARM Counter (ALARM_CNT)	32	RO	00000000h
48h	FMU_ERR_OUT Configuration (FOUT_CFG)	32	RW	00000000h
4ch	FMU_ERR_OUT Minimum Window (FOUT_MIN_WIN)	32	RW	00000000h
50h	Software Control FMU_ERR_OUT (FOUT_FRC)	32	RW	00000001h
54h	FMU_ERR_OUT Window Counter (FOUT_CNT)	32	RO	FFFFFFFFh
58h	FMU Fail To React Control (FFTR_CTRL)	32	RW	00000000h
5ch	FMU Fail To React Counter Timeout Threshold (FFTR_CNT_TO_TH)	32	RW	00000000h
60h	FMU Fail To React Counter (FFTR_CNT)	32	RO	00000000h
64h	Fault Injection Register (FAULT_INJ)	32	RW	00010000h
68h	FMU_ERR_IN Control Register (FIN_CTRL)	32	RW	00000000h
100h	FMU Fault Channel Control Register n (FCH_CTRL0)	32	RW	00000000h
104h	FMU Fault Channel Control Register n (FCH_CTRL1)	32	RW	00000000h
108h	FMU Fault Channel Control Register n (FCH_CTRL2)	32	RW	00000000h
10ch	FMU Fault Channel Control Register n (FCH_CTRL3)	32	RW	00000000h
110h	FMU Fault Channel Control Register n (FCH_CTRL4)	32	RW	00000000h
114h	FMU Fault Channel Control Register n (FCH_CTRL5)	32	RW	00000000h
118h	FMU Fault Channel Control Register n (FCH_CTRL6)	32	RW	00000000h
11ch	FMU Fault Channel Control Register n (FCH_CTRL7)	32	RW	00000000h
120h	FMU Fault Channel Control Register n (FCH_CTRL8)	32	RW	00000000h
124h	FMU Fault Channel Control Register n (FCH_CTRL9)	32	RW	00000000h
128h	FMU Fault Channel Control Register n (FCH_CTRL10)	32	RW	00000000h
12ch	FMU Fault Channel Control Register n (FCH_CTRL11)	32	RW	00000000h
130h	FMU Fault Channel Control Register n (FCH_CTRL12)	32	RW	00000000h
134h	FMU Fault Channel Control Register n (FCH_CTRL13)	32	RW	00000000h
138h	FMU Fault Channel Control Register n (FCH_CTRL14)	32	RW	00000000h
13ch	FMU Fault Channel Control Register n (FCH_CTRL15)	32	RW	00000000h
140h	FMU Fault Channel Control Register n (FCH_CTRL16)	32	RW	00000000h
144h	FMU Fault Channel Control Register n (FCH_CTRL17)	32	RW	00000000h
148h	FMU Fault Channel Control Register n (FCH_CTRL18)	32	RW	00000000h
14ch	FMU Fault Channel Control Register n (FCH_CTRL19)	32	RW	00000000h
150h	FMU Fault Channel Control Register n (FCH_CTRL20)	32	RW	00000000h
154h	FMU Fault Channel Control Register n (FCH_CTRL21)	32	RW	00000000h
158h	FMU Fault Channel Control Register n (FCH_CTRL22)	32	RW	00000000h
15ch	FMU Fault Channel Control Register n (FCH_CTRL23)	32	RW	00000000h
160h	FMU Fault Channel Control Register n (FCH_CTRL24)	32	RW	00000000h

Yuntu Microelectronics

Offset	Register	Width	Access	Reset Value
164h	FMU Fault Channel Control Register n (FCH_CTRL25)	32	RW	00000000h
168h	FMU Fault Channel Control Register n (FCH_CTRL26)	32	RW	00000000h
16ch	FMU Fault Channel Control Register n (FCH_CTRL27)	32	RW	00000000h
170h	FMU Fault Channel Control Register n (FCH_CTRL28)	32	RW	00000000h
174h	FMU Fault Channel Control Register n (FCH_CTRL29)	32	RW	00000000h
178h	FMU Fault Channel Control Register n (FCH_CTRL30)	32	RW	00000000h
17ch	FMU Fault Channel Control Register n (FCH_CTRL31)	32	RW	00000000h
180h	FMU Fault Channel Control Register n (FCH_CTRL32)	32	RW	00000000h
184h	FMU Fault Channel Control Register n (FCH_CTRL33)	32	RW	00000000h
188h	FMU Fault Channel Control Register n (FCH_CTRL34)	32	RW	00000000h
18ch	FMU Fault Channel Control Register n (FCH_CTRL35)	32	RW	00000000h

44.3.1.1 FMU LOCK Register

Offset: 0h

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	KEY															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 44.4: FMU LOCK Register Description

Field	Function
15 - 0 KEY	**Key for Unlock and Lock Protection of All FMU Registers** Configure to unlock or lock protection.Write 0xA765 to this register to unlock;write 0x46B0 to lock. When read this register, it returns the lock status. Read value 0x0 means lock state. Read value 0x1 means unlock state.

Yuntu Microelectronics

44.3.1.2 FMU OP Register

Offset: 4h

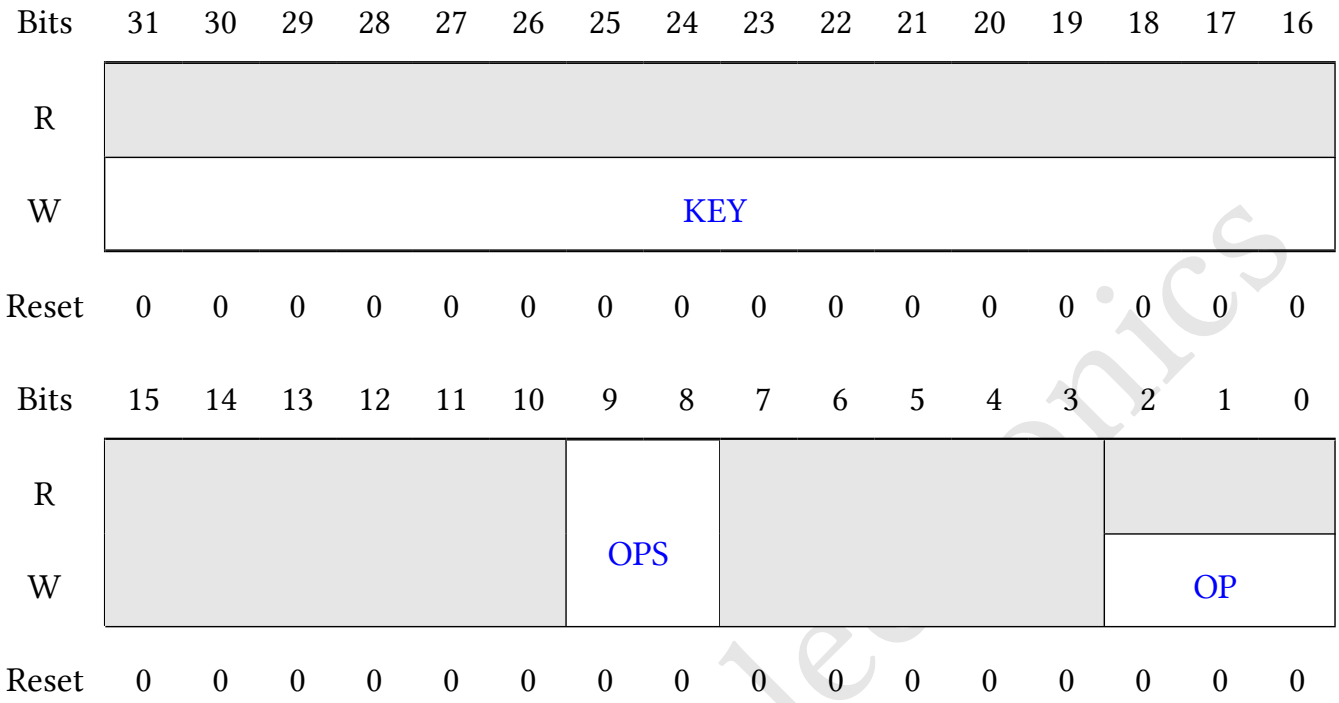


Table 44.5: FMU OP Register Description

Field	Function
31 - 16 KEY	**Operation Configuration Lock Key** Write Only. OP can only be configured if KEY is written 0xFA65 at the same time.
9 - 8 OPS	**Operation Status** Read Only. OPS indicates the operation status. 00b - IDLE 01b - In progress 10b - Aborted 11b - Successful
2 - 0 OP	**Operation Codes** Write only. Configure OP to change FMU status or clear status register. OP5~OP7 are reserved. 000b - OP0, no operation 001b - OP1, put FMU in CFG state 010b - OP2, put FMU in RUN state 011b - OP3, clear FAULT_FSM_ST and ALARM_FSM_ST register 100b - OP4, set OPS to IDLE

Yuntu Microelectronics

44.3.1.3 FMU CTRL Register

This register is reset by Power-on Reset.

Offset: 8h

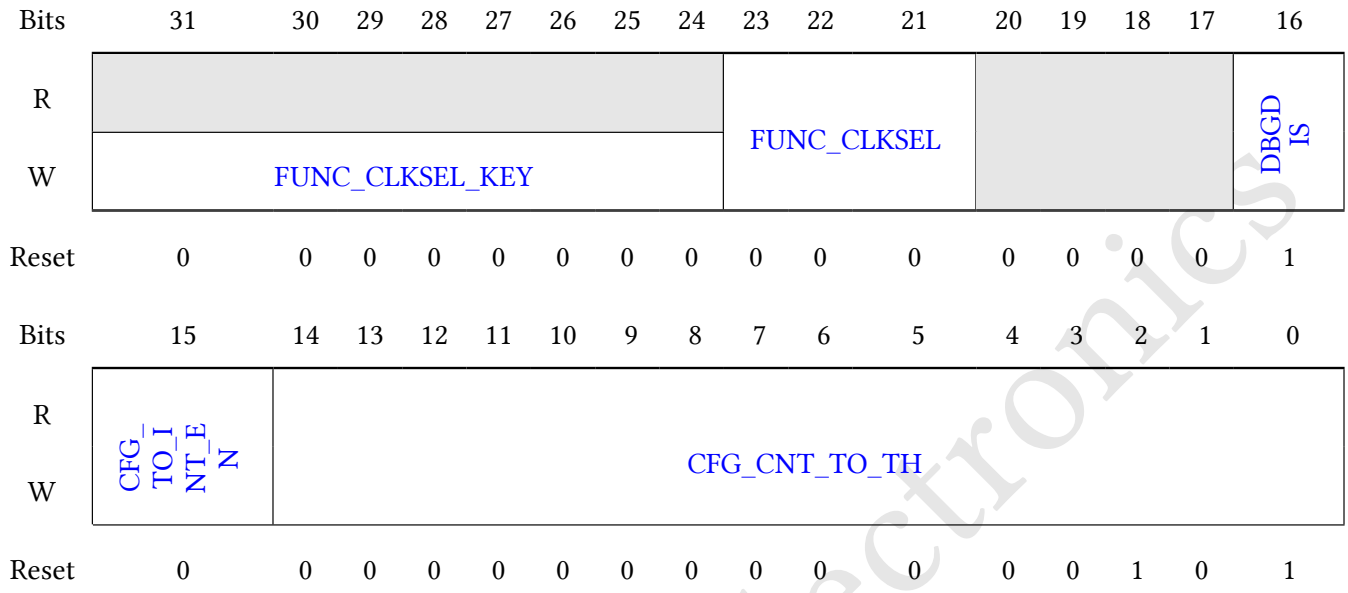


Table 44.6: FMU CTRL Register Description

Field	Function
31 - 24 FUNC_CLKSEL_KEY	**Key For Module Functional Clock Select** Write only. FUNC_CLKSEL can be configured only if FUNC_CLKSEL_KEY is written 0x5A at the same time.
23 - 21 FUNC_CLKSEL	**Module Functional Clock Select** Select FMU module's functional clock source. This bit is reset by POR and should be configured when FMU module is static. 0x5~0x7 are reserved, no clock source selected. 000b - SIRC_DIV4(SIRC clock divided by 4) 001b - FIRC 010b - SXOSC 011b - FXOSC 100b - PLL
16 DBGDIS	**Disable Fault Input in Debug Mode** Configure to enable or disable fault input in debug mode. This bit is reset by POR. 0b - Fault input is enabled in debug mode 1b - Fault input is disabled in debug mode
15	**CFG State Timeout Interrupt Enable**

Yuntu Microelectronics

Field	Function
CFG_TO_INT_EN	Configure this bit to enable CFG state timeout interrupt. This bit is reset by POR. 0b - CFG state timeout interrupt is disabled 1b - CFG state timeout interrupt is enabled
14 - 0 CFG_CNT_TO_TH	**CFG State Counter Timeout Threshold** Configure the CFG state counter timeout threshold. The CFG state counter timeout if it reaches (CFG_CNT_TO_TH * 32). CFG_CNT_TO_TH is reset by POR.

44.3.1.4 FMU ST Register

Offset: ch

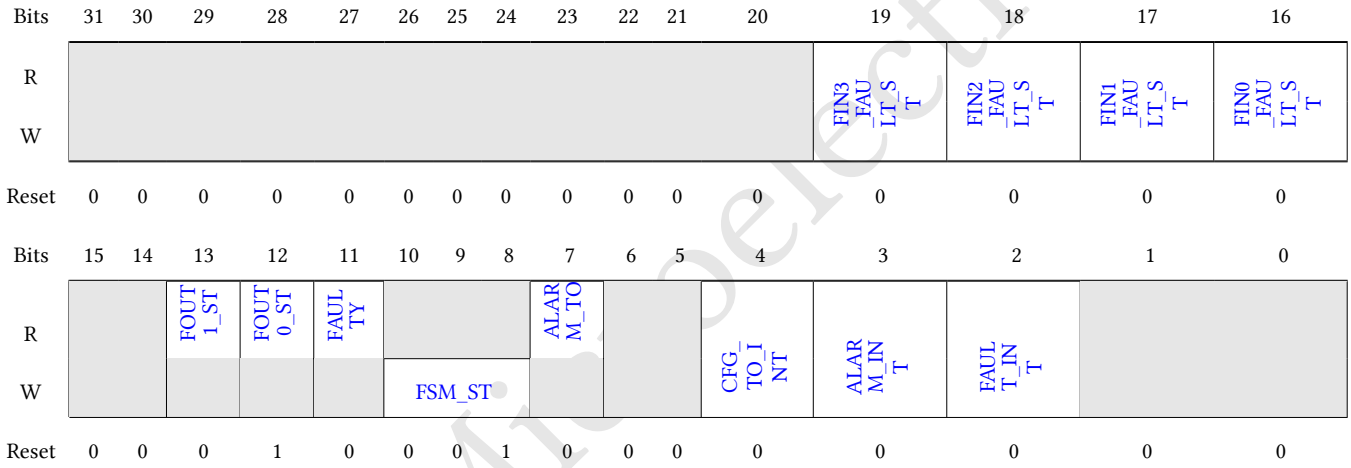


Table 44.7: FMU ST Register Description

Field	Function
19 FIN3_FAULT_ST	**FMU_ERR_IN3 Fault Status** FMU_ERR_IN3 fault status. Write 1 to clear. This bit is reset by Power-on Reset. 0b - No FMU_ERR_IN3 fault 1b - FMU_ERR_IN3 fault has occurred
18 FIN2_FAULT_ST	**FMU_ERR_IN2 Fault Status** FMU_ERR_IN2 fault status. Write 1 to clear. This bit is reset by Power-on Reset. 0b - No FMU_ERR_IN2 fault 1b - FMU_ERR_IN2 fault has occurred
17	**FMU_ERR_IN1 Fault Status**

Yuntu Microelectronics

Field	Function
FIN1_FAULT_ST	FMU_ERR_IN1 fault status. Write 1 to clear. This bit is reset by Power-on Reset. 0b - No FMU_ERR_IN1 fault 1b - FMU_ERR_IN1 fault has occurred
16 FIN0_FAULT_ST	**FMU_ERR_IN0 Fault Status** FMU_ERR_IN0 fault status. Write 1 to clear. This bit is reset by Power-on Reset. 0b - No FMU_ERR_IN0 fault 1b - FMU_ERR_IN0 fault has occurred
13 FOUT1_ST	**FOUT1 Status** Read Only. FOUT1_ST monitors fault output 1. For test.
12 FOUT0_ST	**FOUT0 Status** Read Only. FOUT0_ST monitors fault output 0. For test.
11 FAULTY	**FMU FSM is FAULT State** Read Only. FAULTY indicates whether FMU FSM is FAULT state. 0b - FMU FSM is not in FAULT state 1b - FMU FSM is in FAULT state.
10 - 8 FSM_ST	**FMU FSM Status** Read Only. Read FSM_ST to check FMU FSM current state. Other value(0x5~0x7) are reserved. 000b - IDLE 001b - RUN 010b - CFG 011b - ALARM 100b - FAULT
7 ALARM_TO	**ALARM State Timeout Flag** Read Only. ALARM_TO indicates whether ALARM state timeout. 0b - ALARM state not timeout. 1b - ALARM state timeout.
4 CFG_TO_INT	**CFG State Timeout Interrupt Flag** CFG state timeout interrupt flag. Write 1 to clear. 0b - No CFG state timeout interrupt 1b - CFG state timeout interrupt has occurred
3 ALARM_INT	**Alarm Interrupt Flag** Alarm interrupt flag. Write 1 to clear. 0b - No alarm interrupt 1b - Alarm interrupt has occurred
2	**Fault NMI Flag**

Yuntu Microelectronics

Field	Function
FAULT_INT	Fault non maskable interrupt flag. Write 1 to clear. 0b - No fault non maskable interrupt 1b - Fault non maskable interrupt has occurred

44.3.1.5 FMU CFG_CNT Register

Offset: 10h

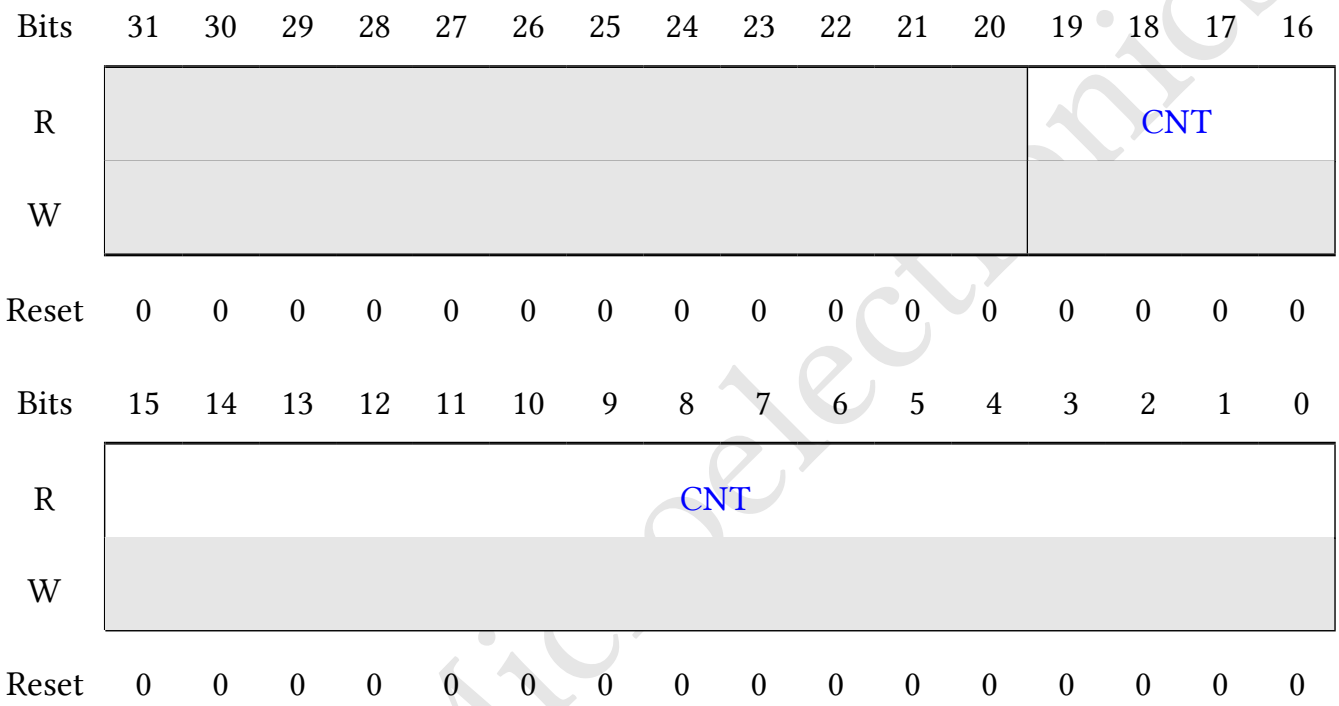


Table 44.8: FMU CFG_CNT Register Description

Field	Function
19 - 0 CNT	**CFG State Counter** Read Only. Read CFG_CNT returns the value of the CFG state counter.

44.3.1.6 FMU FAULT_ST Register

Offset: 14h + n * 4h

Yuntu Microelectronics

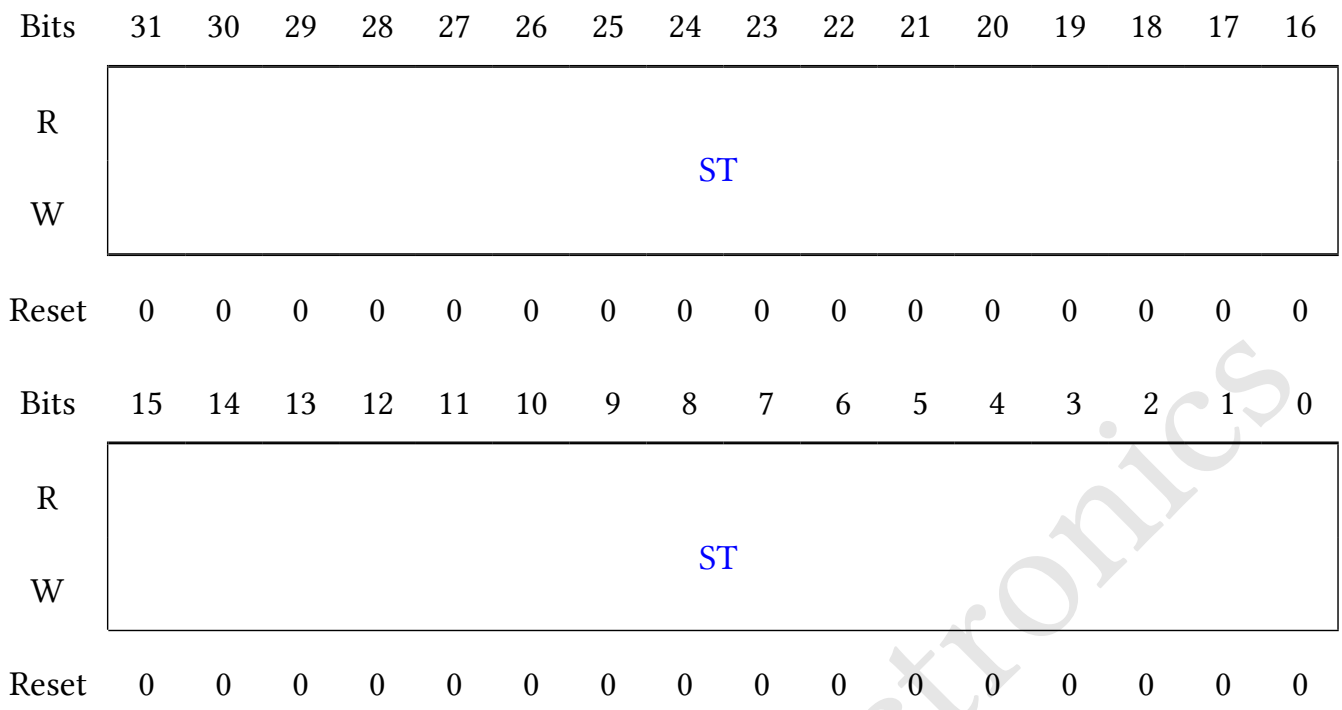


Table 44.9: FMU FAULT_ST Register Description

Field	Function
31 - 0 ST	**FAULT Status of Every FAULT Channel** FAULT_ST[m].ST[n](m=0~1, n=0~31) is the FAULT status of channel numbered by ”32*m+n”. Write 1 to clear. When m=0, n=0~31, When m=1, n=0~3. When FAULT_ST[m].ST[n] is 0b, the channel[32*m+n] has no fault When FAULT_ST[m].ST[n] is 1b, The channel[32*m+n] fault has occurred

Yuntu Microelectronics

44.3.1.7 FMU FAULT_ST_KEY Register

Offset: 34h

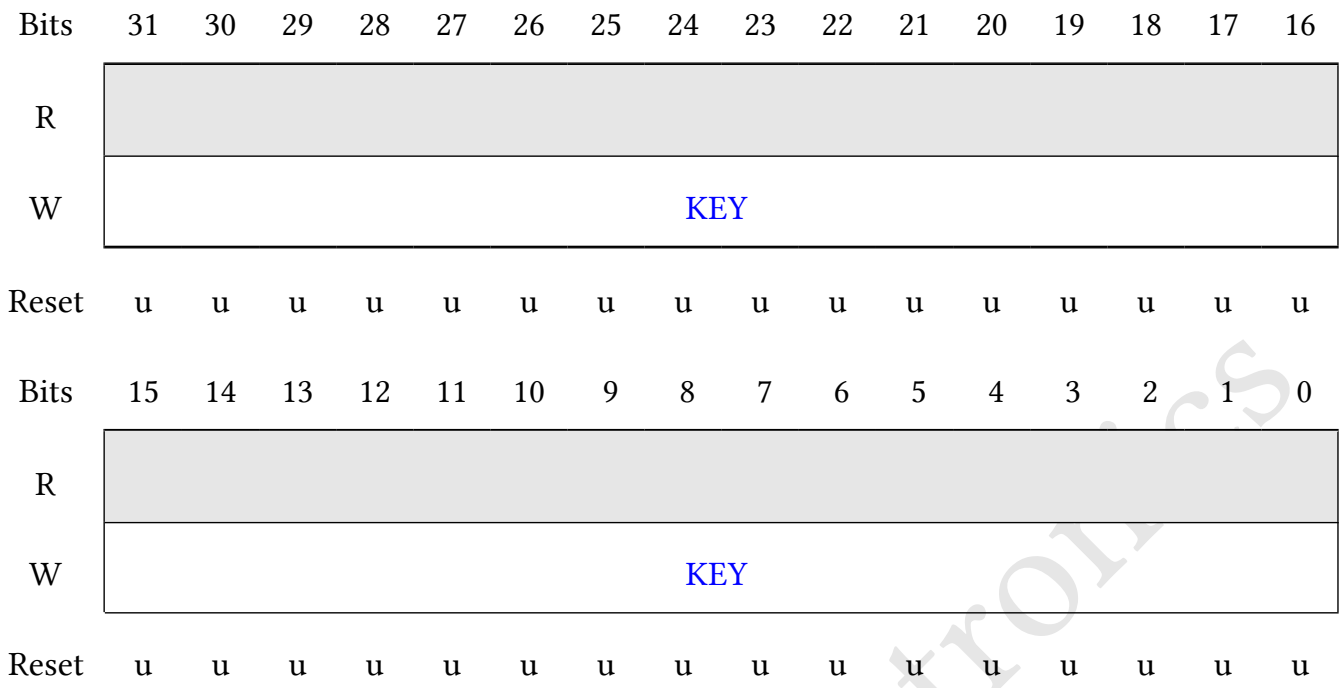


Table 44.10: FMU FAULT_ST_KEY Register Description

Field	Function
31 - 0 KEY	**FAULT_ST Write Protection Key** Write Only. Write 0x52CB_6701 to unlock FAULT_ST write.

Yuntu Microelectronics

44.3.1.8 FMU FAULT_FSM_ST Register

Offset: 38h

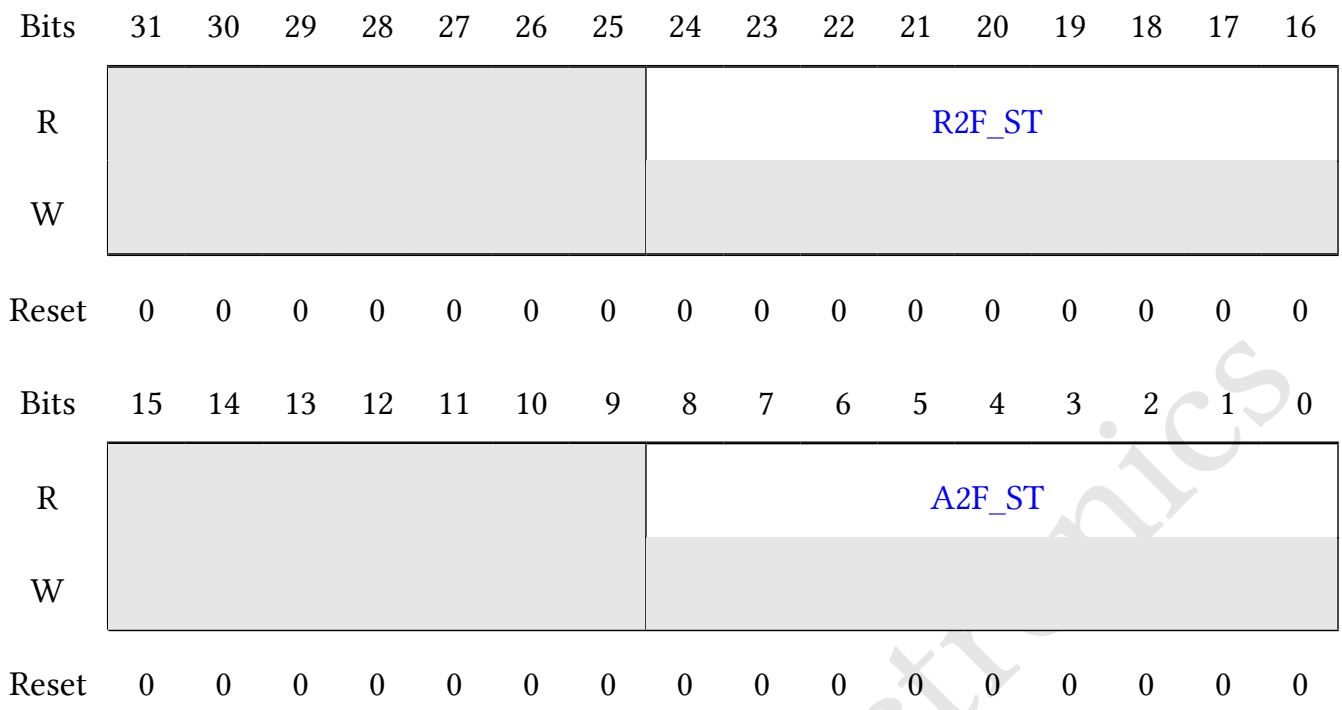


Table 44.11: FMU FAULT_FSM_ST Register Description

Field	Function
24 - 16 R2F_ST	**RUN to FAULT Status** Record the number of FAULT channel which caused FMU FSM turning to FAULT from RUN, fault channel ID is (R2F_ST - 1). Reset by POR. For internal test.
8 - 0 A2F_ST	**AlARM to FAULT Status** Record the number of FAULT channel which caused FMU FSM turning to FAULT from ALARM. Reset by POR. For internal test.

Yuntu Microelectronics

44.3.1.9 FMU ALARM_CNT_TO_TH Register

This register is reset by Power-on Reset.

Offset: 3ch

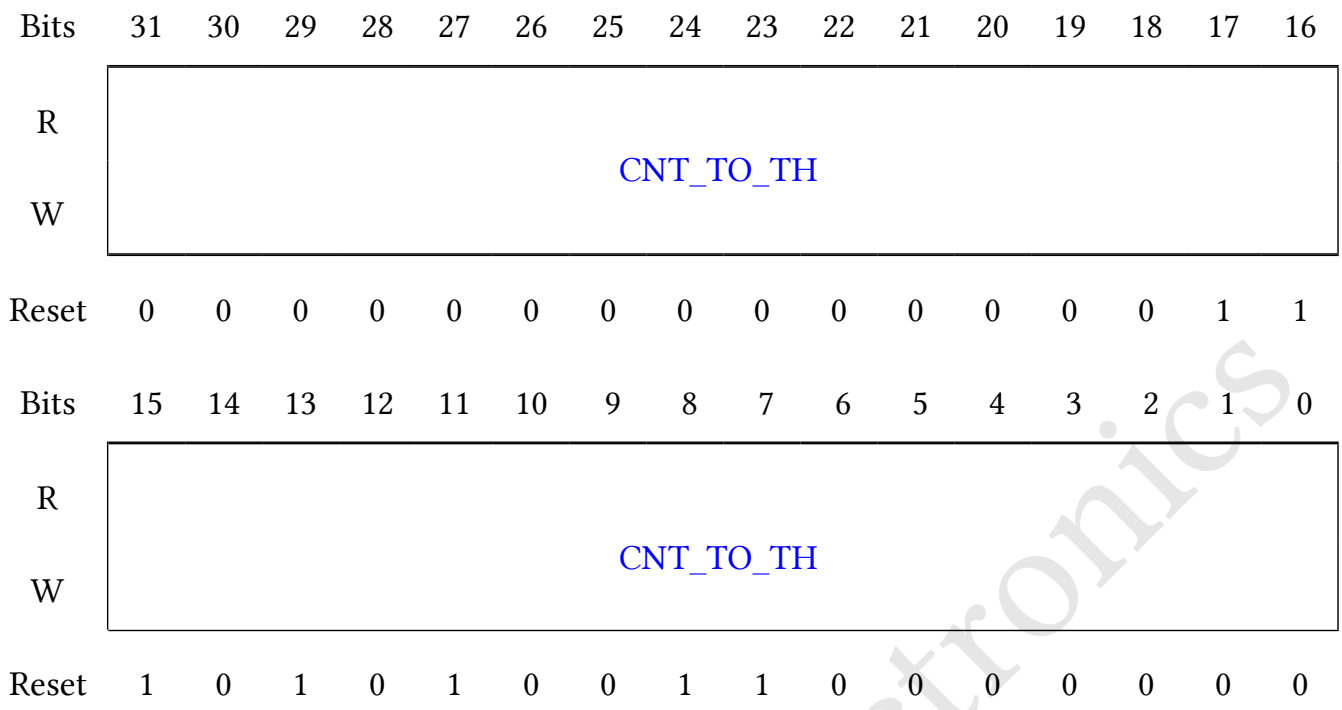


Table 44.12: FMU ALARM_CNT_TO_TH Register Description

Field	Function
31 - 0 CNT_TO_TH	**ALARM Counter Timeout Threshold** When ALARM counter threshold expired, ALARM timeout occurs

Yuntu Microelectronics

44.3.1.10 FMU ALARM_CNT Register

Offset: 40h

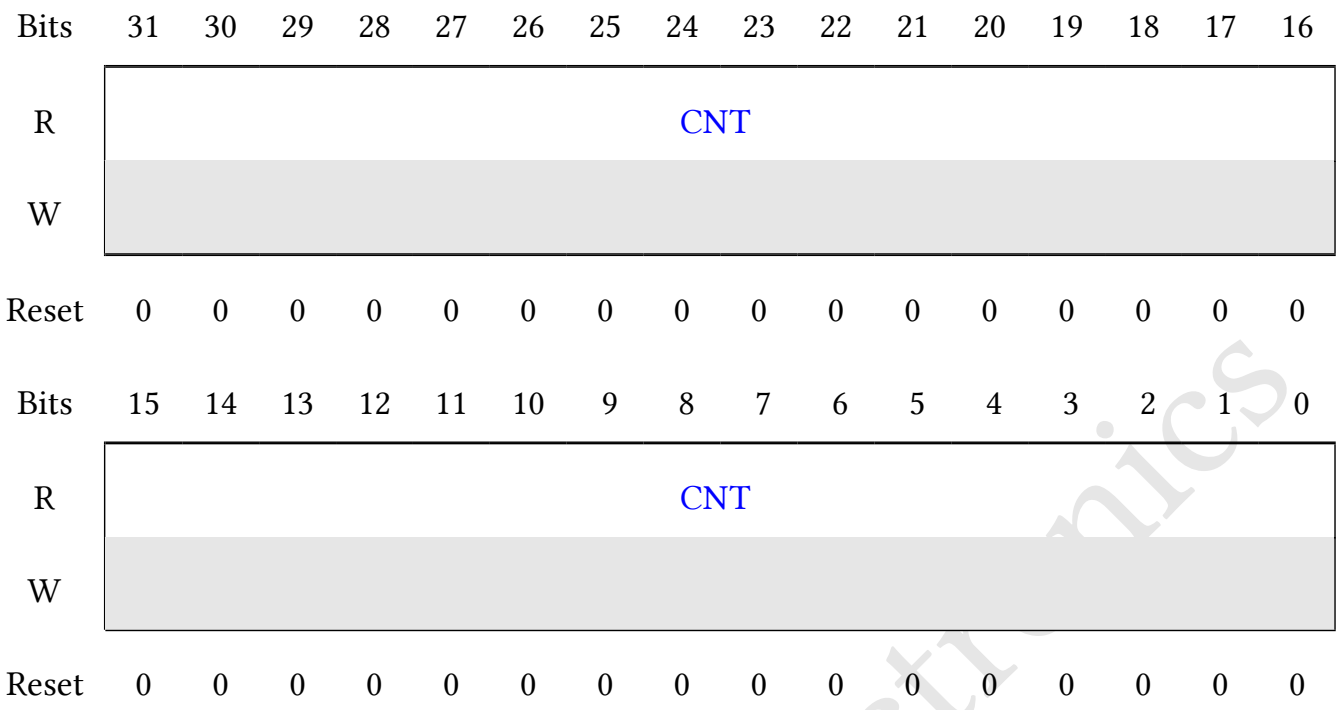


Table 44.13: FMU ALARM_CNT Register Description

Field	Function
31 - 0 CNT	**ALARM Counter** Read Only. Read CNT returns the value of ALARM counter.

44.3.1.11 FMU FOUT_CFG Register

This register is reset by Power-on Reset.

Offset: 48h

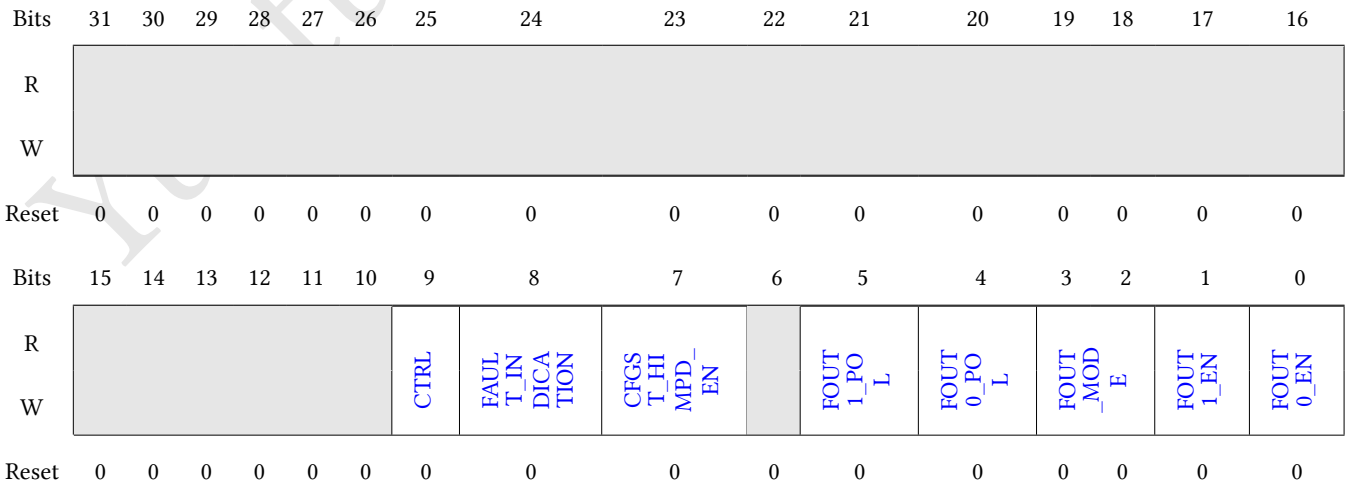


Table 44.14: FMU FOUT_CFG Register Description

Field	Function
9 CTRL	**FMU_ERR_OUT Output Control** Configure FMU_ERR_OUT source. It's reset by POR. 0b - FMU_ERR_OUT is controlled by FMU FSM 1b - FMU_ERR_OUT is controlled by Soft(FOUT_FRC)
8 FAULT_IN DICATION	**FMU_ERR_OUT Output as FAULT When in CFG State** Configure FAULT_INDICATION to select whether FMU_ERR_OUT output as FAULT when in CFG state. It's reset by POR. 0b - FMU_ERR_OUT output as RUN state when in CFG state 1b - FMU_ERR_OUT output as FAULT state when in CFG state
7 CFGST_HI MPD_EN	**CFG State High Impedance Output Enable** Enable FMU_ERR_OUT high impedance output when FMU in CFG state. It's reset by POR. 0b - Disable high impedance output when in CFG state 1b - Enable high impedance output when in CFG state
5 FOUT1_PO L	**FMU_ERR_OUT1 Polarity** Configure FMU_ERR_OUT1 polarity. It's reset by POR. 0b - FMU_ERR_OUT1 is high when in fault state 1b - FMU_ERR_OUT1 is low when in fault state
4 FOUT0_PO L	**FMU_ERR_OUT0 Polarity** Configure FMU_ERR_OUT0 polarity. It's reset by POR. 0b - FMU_ERR_OUT0 is low when in fault state 1b - FMU_ERR_OUT0 is high when in fault state
3 - 2 FOUT_MOD E	**FMU FAULT Output Mode** Select FMU fault output mode. It's reset by POR. 0x1~0x3 are reserved. 00b - Bi-stable mode
1 FOUT1_EN	**FMU_ERR_OUT1 Output Enable** Configure to enable output FMU_ERR_OUT1. It's reset by POR. 0b - FMU_ERR_OUT1 output disable 1b - FMU_ERR_OUT1 output enable
0 FOUT0_EN	**FMU_ERR_OUT0 Output Enable** Configure to enable output FMU_ERR_OUT0. It's reset by POR. 0b - FMU_ERR_OUT0 output disable 1b - FMU_ERR_OUT0 output enable

Yuntu Microelectronics

44.3.1.12 FMU FOUT_MIN_WIN Register

This register is reset by Power-on Reset.

Offset: 4ch

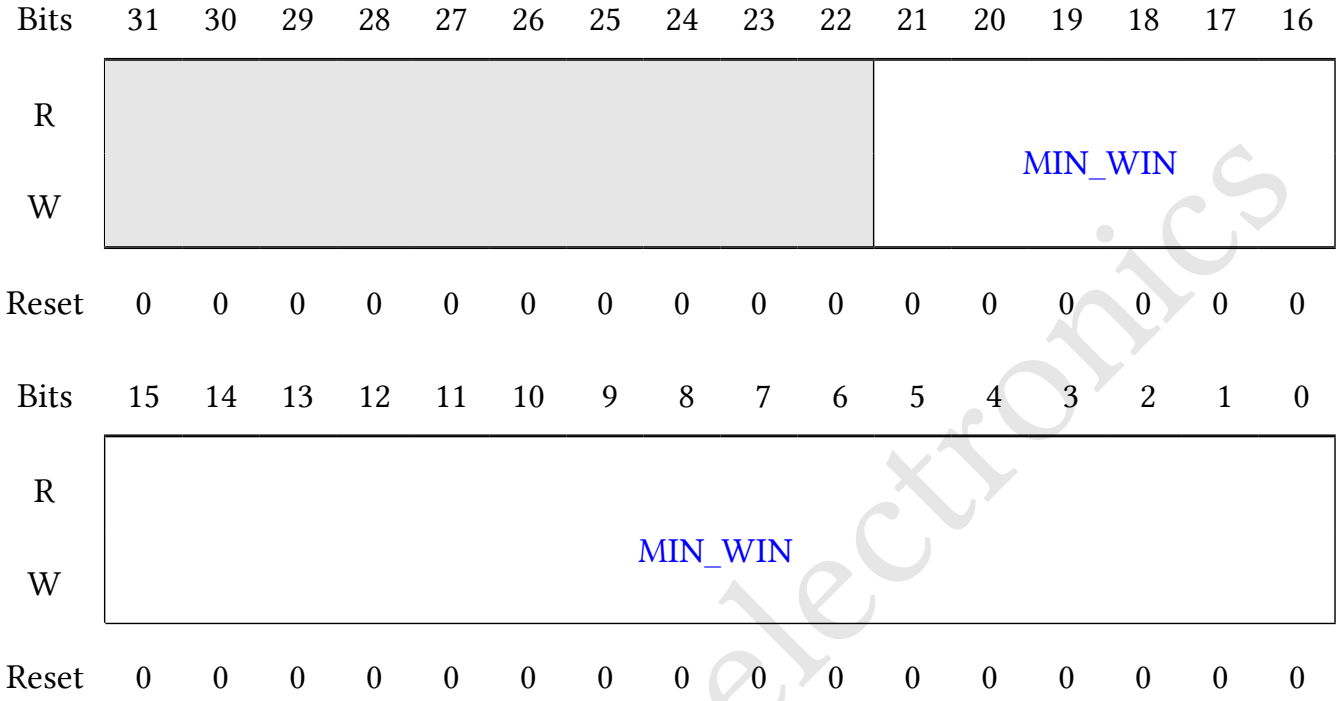


Table 44.15: FMU FOUT_MIN_WIN Register Description

Field	Function
21 - 0 MIN_WIN	**FMU_ERR_OUT Output Minimum Window** Configure the minimum time window of FMU_ERR_OUT output. It's reset by POR. The minimum time is (MIN_WIN * 1024) cycles of functional clock.

44.3.1.13 FMU FOUT_FRC Register

Offset: 50h

Yuntu Microelectronics

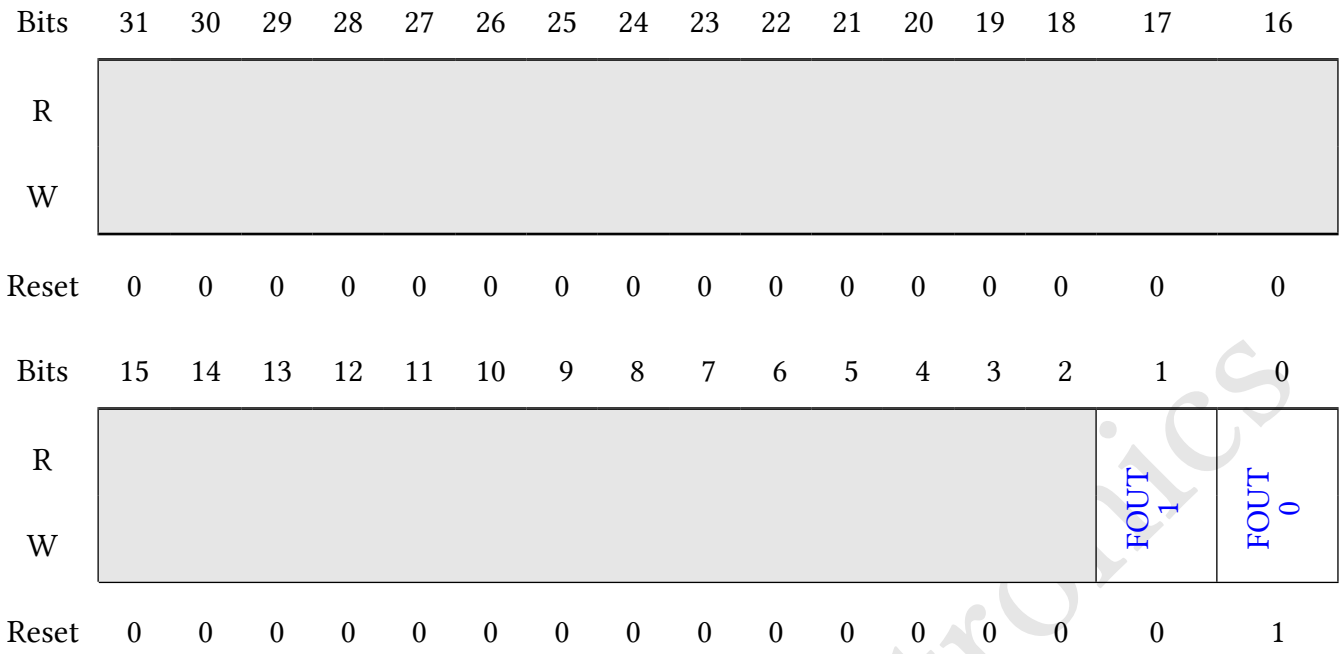


Table 44.16: FMU FOUT_FRC Register Description

Field	Function
1 FOUT1	**FMU_ERR_OUT1 Control** Control FMU_ERR_OUT1 output 0b - FMU_ERR_OUT1 output low 1b - FMU_ERR_OUT1 output high
0 FOUT0	**FMU_ERR_OUT0 Control** Control FMU_ERR_OUT0 output 0b - FMU_ERR_OUT0 output low 1b - FMU_ERR_OUT0 output high

Yuntu Microelectronics

44.3.1.14 FMU FOUT_CNT Register

Offset: 54h

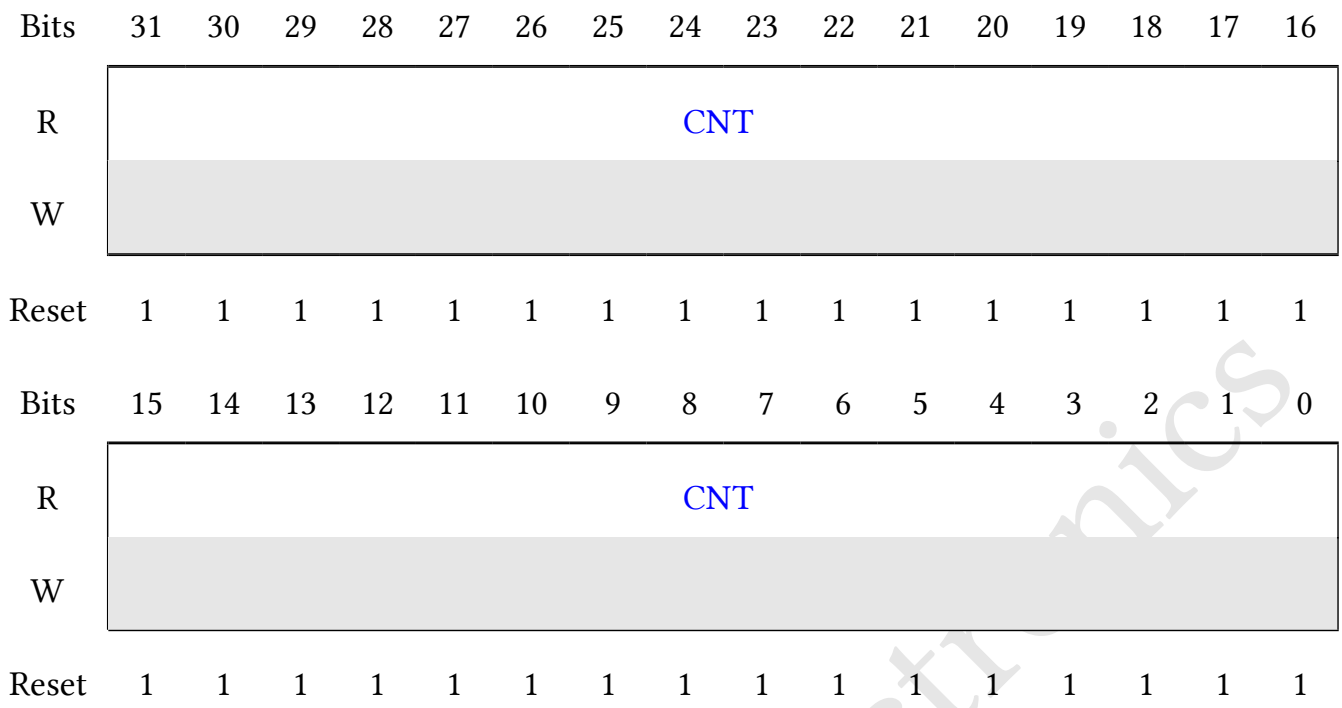


Table 44.17: FMU FOUT_CNT Register Description

Field	Function
31 - 0 CNT	**FMU_ERR_OUT Window Counter** Read FOUT_CNT returns the value of FMU_ERR_OUT window counter

Yuntu Microelectronics

44.3.1.15 FMU FFTR_CTRL Register

This register is reset by Power-on Reset.

Offset: 58h

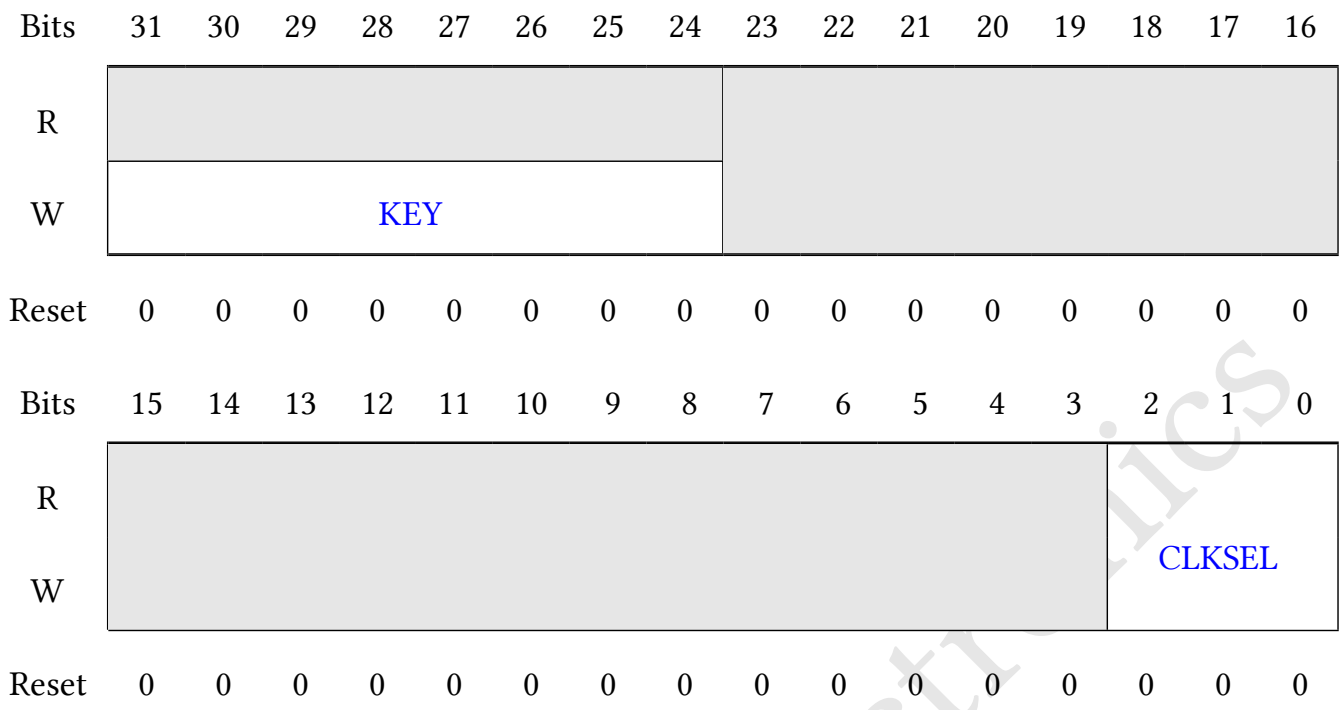


Table 44.18: FMU FFTR_CTRL Register Description

Field	Function
31 - 24 KEY	**CLKSEL Write Protection Key** CLKSEL can only be configured if KEY is written 0xA5 at the same time.
2 - 0 CLKSEL	**FMU Fail To React Detect Clock Select** Select the clock source of FMU fail to react detect function. It's reset by POR and should be configured when FMU module is static. 0x5~0x7 are reserved, no clock source selected. 000b - SIRC_DIV4(SIRC clock divided by 4) 001b - FIRC 010b - SXOSC 011b - FXOSC 100b - PLL

Yuntu Microelectronics

44.3.1.16 FMU FFTR_CNT_TO_TH Register

This register is reset by Power-on Reset.

Offset: 5ch

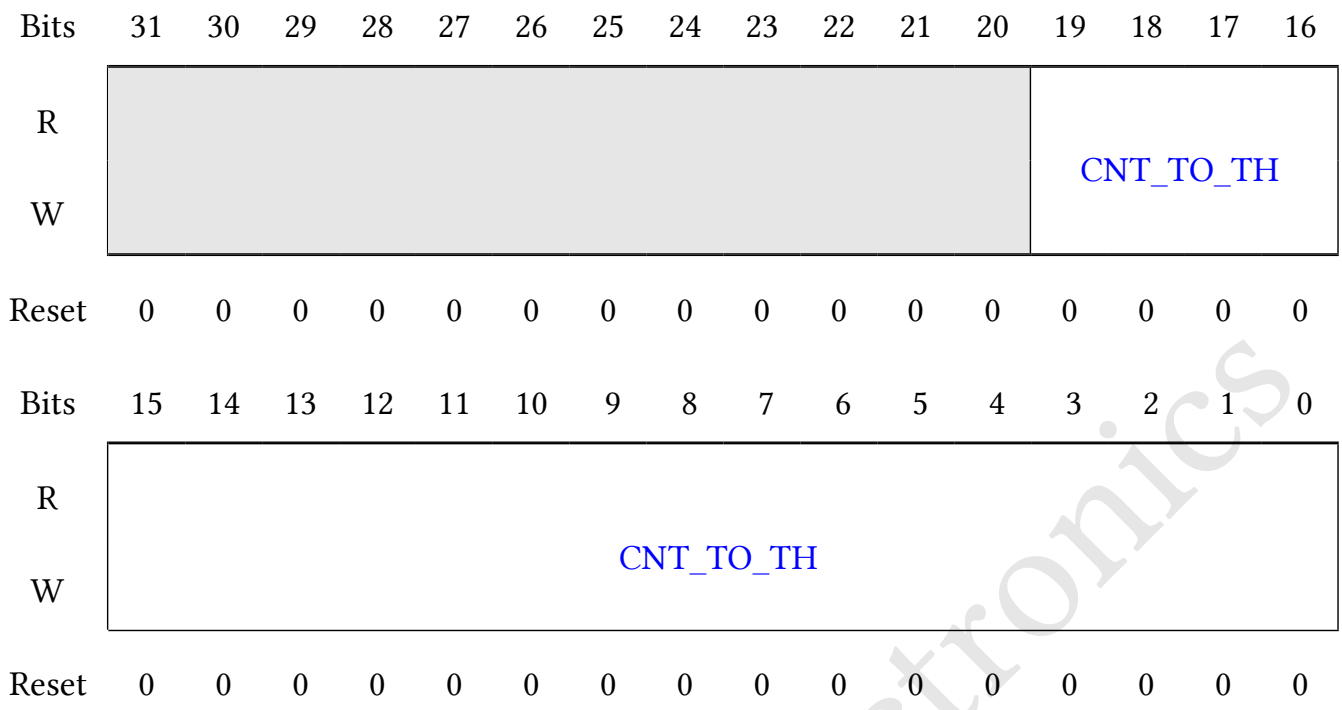


Table 44.19: FMU FFTR_CNT_TO_TH Register Description

Field	Function
19 - 0 CNT_TO_TH	**FMU Fail To React Counter Timeout Threshold** Configure FMU fail to react counter timeout threshold. The counter threshold is ((CNT_TO_TH + 1) * 4096) cycles of FFTR clock. It's reset by POR.

Yuntu Microelectronics

44.3.1.17 FMU FFTR_CNT Register

Offset: 60h

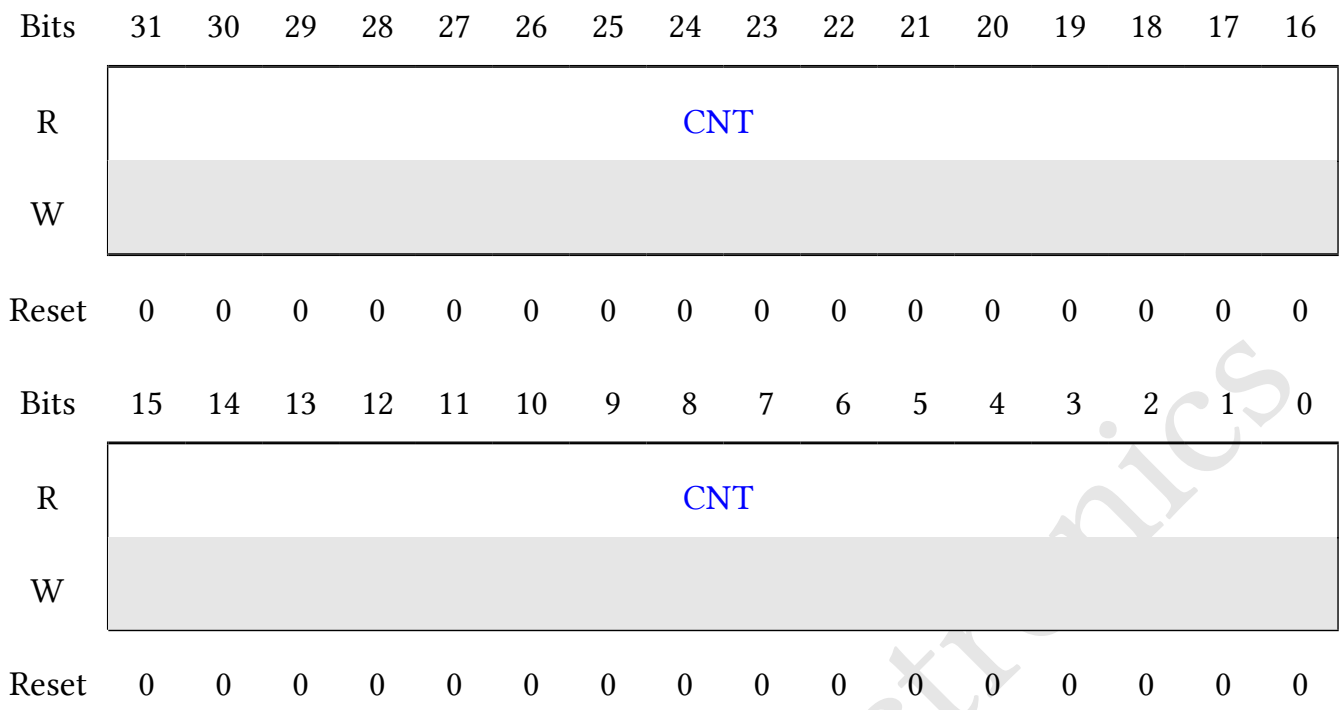


Table 44.20: FMU FFTR_CNT Register Description

Field	Function
31 - 0 CNT	**FMU Fail To React Counter** Read FFTR_CNT returns the value of FMU fail to react counter

Yuntu Microelectronics

44.3.1.18 FMU FAULT_INJ Register

Offset: 64h

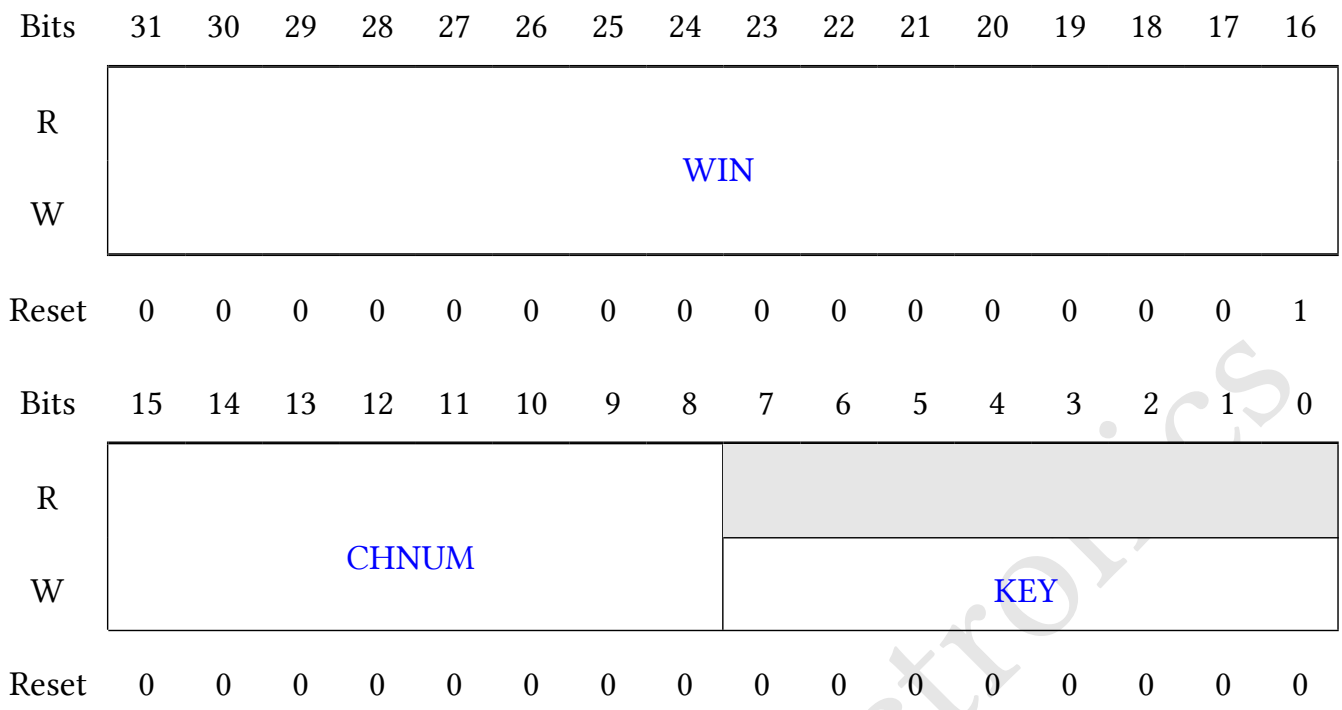


Table 44.21: FMU FAULT_INJ Register Description

Field	Function
31 - 16 WIN	**Fault Injection Window** Configure fault injection time window
15 - 8 CHNUM	**Fault Injection Channel Number** Configure the channel number to inject fault.
7 - 0 KEY	**KEY Protection for Fault Injection** Fault injection assert when KEY is written 0x69.

Yuntu Microelectronics

44.3.1.19 FMU FIN_CTRL Register

This register is reset by Power-on Reset.

Offset: 68h

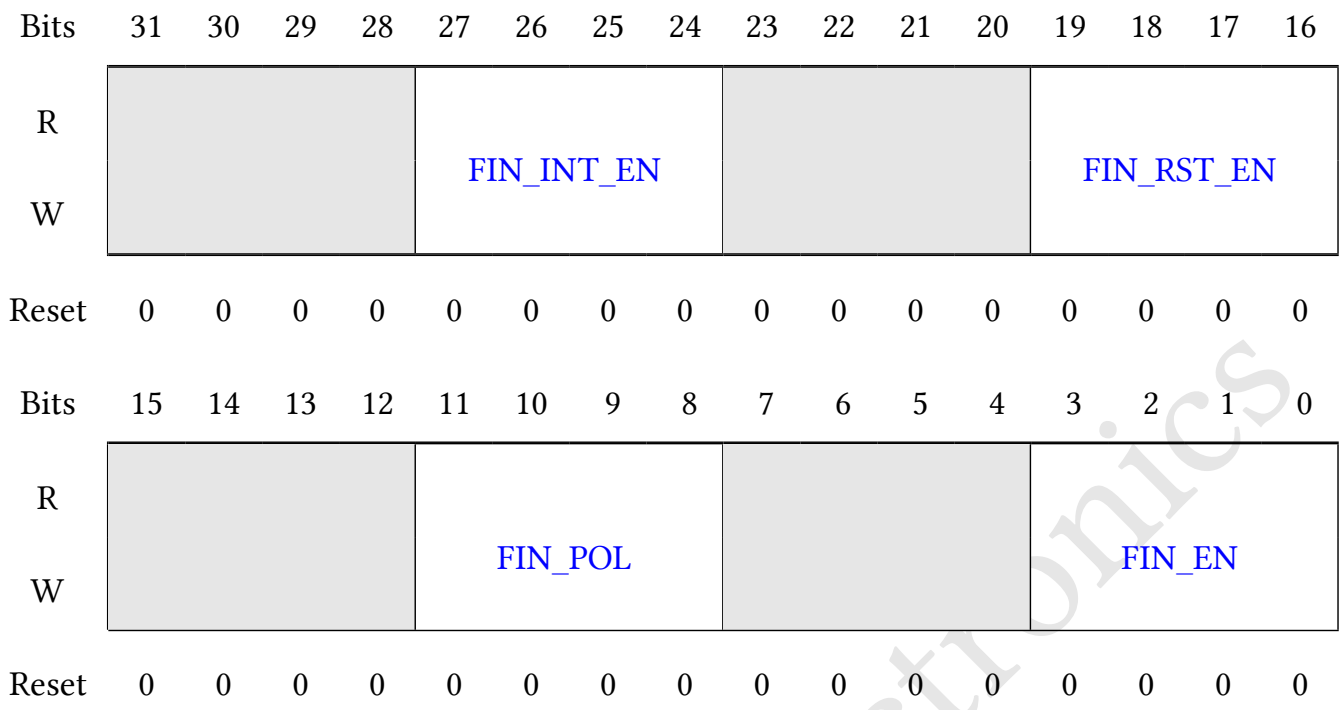


Table 44.22: FMU FIN_CTRL Register Description

Field	Function
27 - 24 FIN_INT_EN	**FMU_ERR_IN Fault Generate Interrupt Enable** Enable FMU_ERR_IN fault to generate interrupt. FIN_INT_EN[n] enables the correspond FMU_ERR_INn to generate interrupt. FIN_INT_EN is reset by POR. When FIN_INT_EN[n] is 0b, FMU_ERR_INn fault not generate interrupt. When FIN_INT_EN[n] is 1b, FMU_ERR_INn fault generate interrupt.
19 - 16 FIN_RST_EN	**FMU_ERR_IN Fault Generate Reset Enable** Enable FMU_ERR_IN fault to generate reset. FIN_RST_EN[n] enables the correspond FMU_ERR_INn to generate reset. FIN_RST_EN is reset by POR. When FIN_RST_EN[n] is 0b, FMU_ERR_INn fault not generate reset. When FIN_RST_EN[n] is 1b, FMU_ERR_INn fault generate reset.
11 - 8 FIN_POL	**FMU_ERR_IN Polarity** Select FMU_ERR_IN Polarity. FIN_POL[n] select the polarity of the correspond FMU_ERR_INn. It's reset by POR. When FIN_POL[n] is 0b, FMU_ERR_INn input high will generate fault. When FIN_POL[n] is 1b, FMU_ERR_INn input low will generate fault.
3 - 0 FIN_EN	**FMU_ERR_IN Enable** Enable FMU_ERR_IN input. Set FIN_EN[n] to 1, the correspond FMU_ERR_INn input is enabled. It's reset by POR. When FIN_EN[n] is 0b, FMU_ERR_INn is disable. When FIN_EN[n] is 1b, FMU_ERR_INn is enable.

Yuntu Microelectronics

44.3.1.20 FMU FCH_CTRL Register

Offset: 100h + n * 4h

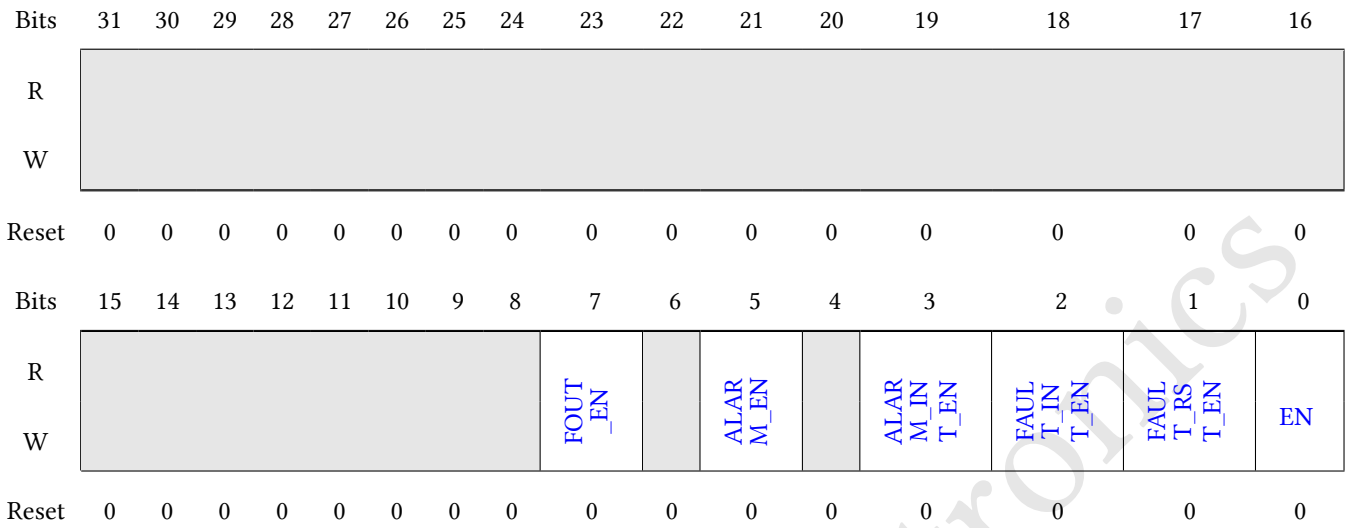


Table 44.23: FMU FCH_CTRL Register Description

Field	Function
7 FOUT_EN	**FMU_ERR_OUT Enable** Enable FMU_ERR_OUT when the fault occurs. It's reset by POR. 0b - FMU_ERR_OUT disabled when the fault occurs 1b - FMU_ERR_OUT enabled when the fault occurs
5 ALARM_EN	**ALARM State Entry Enable** Enable the FMU FSM enter ALARM state when the fault comes. It's reset by POR. 0b - The FMU FSM won't enter ALARM state when the fault comes 1b - The FMU FSM enter ALARM state when the fault comes
3 ALARM_INT_EN	**ALARM Generate Interrupt Enable** Enable alarm to generate interrupt. It's reset by POR. 0b - The alarm not generate interrupt. 1b - The alarm generates interrupt.
2 FAULT_INT_EN	**Fault Generate Interrupt Enable** Enable fault to generate interrupt. It's reset by POR. 0b - The fault not generate NMI interrupt. 1b - The fault generates NMI interrupt.
1 FAULT_RST_EN	**Fault Generate Reset Enable** Enable fault to generate reset. It's reset by POR. 0b - The fault not generate reset. 1b - The fault generate reset.

Yuntu Microelectronics

Field	Function
0 EN	**Fault Channel Enable** Enable the fault channel. It's reset by POR. 0b - The fault channel is disable 1b - The fault channel is enable

44.4 Function

44.4.1 FMU Reset Source

FMU has 2 reset sources, module reset from IPC and power on reset.

Table 44.24: FMU Reset Source Table

Reset Name	Usage	Reset Source
Module Reset	Reset FMU FSM, FFTR Counter, CFG Timeout Counter	From IPC_FMU reset
Power On Reset	FMU Configuration Registers, detailed information please refer to Register descriptions	PMC Power on Reset

Any chip reset will reset FMU FSM, FFTR Counter, CFG Counter. FMU fault function configurations will only be reset by power on reset. So when chip reset (not power on reset) occurs, the configurations are kept, FMU can go on working after reset deasserted. Of course, if needed, user can configure FMU again after chip reset for robustness.

44.4.2 FMU FAULT Inputs

44.4.2.1 Fault Inputs On-chip

Every fault source on-chip has respective enable control in register FCHn_CTRL. When an enabled fault comes, the relative status bit in FAULT_STn is set. User can get the fault source information from FAULT_STn register. Through FCHn_CTRL, user can configure a fault to be a alarm source or a fault source, and can also sets how the FMU reacts to the fault(such as generate a normal interrupt, Non-maskable interrupt, fault indication output or generate a reset), it depends on the user's assessment of the severity of the fault to the system.

44.4.2.2 Fault Inputs from PAD(FMU_ERR_IN)

FMU supports fault inputs through FMU_ERR_IN I/O function. Every FMU_ERR_IN has respective enable/polarity control in FIN_CTRL register. When an enable fault FMU_ERR_INn occurs, the relative FINn_FAULT_ST bit in ST is set. User can get the FMU_ERR_INn information from ST register. Through FIN_CTRL, user can set the FMU reacts to FMU_ERR_INn fault by generating a Non-maskable interrupt or a reset. By default, FMU_ERR_IN fault will output fault indication if Bi-stable mode enabled.

44.4.2.3 Fault Recovery

Yuntu Microelectronics

44.4.2.3.1 Software Recovery

For software recovery fault, user can clear the status bit in FAULT_ST by writing 1 to the bit you want to clear.

44.4.2.3.2 Hardware Recovery

The fault indication is a level-sensitive signal that remains asserted until the fault cause is deasserted. The status bit in FAULT_ST is cleared when the fault signal goes to deasserted and write 1 to the relative bit in FAULT_ST you want to clear.

44.4.2.4 Disable FAULT in Debug Mode

If DBGDIS bit in CTRL register is set, all the fault inputs are disabled.

44.4.3 FMU FSM

FMU FSM has 4 states including NORMAL, CFG, ALARM and FAULT states.

44.4.3.1 NORMAL

After reset released, FMU FSM turns to NORMAL state. FMU has no enabled faults occurring in this state.

44.4.3.2 CFG

After FMU FSM set to CFG state, user can modify the configuration of FMU. A subset of the FMU registers, dedicated to FMU configuration, including global configuration, reactions to fault, timeout, fault masking, can be accessed in write mode only in the CFG state.

The CFG to NORMAL state transition is executed by SW or automatically following a timeout condition of the CFG counter. If CFG counter is timeout, the configuration registers that are writable only in the CFG state are reset to default values. And a CFG timeout interrupt occurs if enabled. The incoming faults, occurring during the CFG state are latched in FAULT_ST registers. Note that after CFG state exit, it's suggested read FAULT_ST registers to check all pending faults that occurred during the CONFIG state and deal with the faults before FMU works at new configurations.

These registers can only be configured in CFG state.

- ALARM_CNT_TO_TH
- FOUT_CFG
- FIN_CTRL
- FOUT_MIN_WIN
- FFTR_CNT_TO_TH
- FCH_CTRLn

44.4.3.3 ALARM

FMU FSM turns into the ALARM state when a fault with alarm enabled occurs. Transition to the ALARM state goes along with an alarm interrupt, if enabled.
This fault may be recovered within a programmable timeout period, before it generates a transition to the FAULT state. The timeout is reinitialized if FMU enters NORMAL state. The timeout restarts following the recovery from the FAULT state.

Yuntu Microelectronics

44.4.3.4 FAULT

FMU FSM turns to FAULT state when one of the below conditions occur:

- A fault with alarm disabled has occurred
- A FMU_ERR_IN fault has occurred
- ALARM state timeout

Transitions to FAULT state go along with generation of NMI(Non-maskable Interrupt) or reset. User can also configure FMU to output fault state indication to external through IO function FMU_ERR_OUT.

After moving to FAULT state, if there is either a previous pending fault or a new fault for which NMI is enabled, NMI generation takes place. Multiple faults can occur at the same time. FSM can't exit from FAULT state until all faults with alarm disabled are recovered.

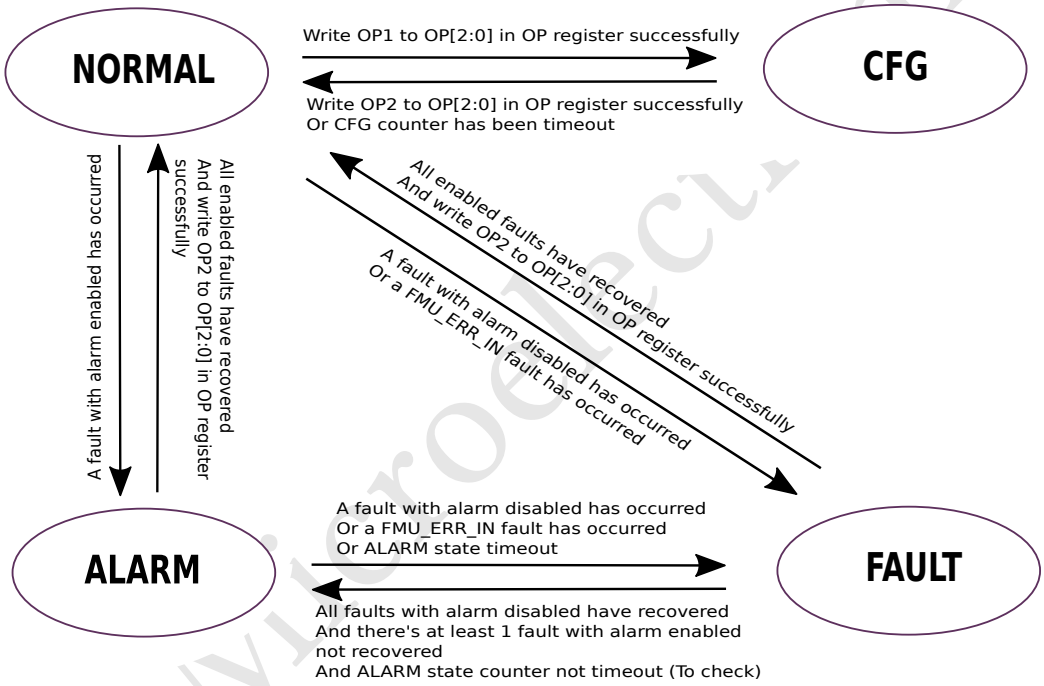


Figure 44.2: FSM Diagram of FMU

44.4.3.5 FMU Fault Indication Output

When in FAULT state, FMU can output fault indication to off-chip device through FMU_ERR_OUT[1:0]. FMU_ERR_OUT can output according to the respective settings. FMU_ERR_OUT can also be used together according to Bi-Stable protocol. Note that the reset reaction has higher priority than the output of fault indication. If a fault with its reaction configured to reset occurs, chip will be reset and FMU_ERR_OUT can't output. FMU_ERR_OUT can indicate three FMU states as below.

- NONFAULTY: The fault output counter is not running or FMU FSM is NORMAL/ALARM state.
- FAULTY: The fault output counter is running or FMU FSM is FAULT state.
- CFG: The fault output counter is not running or FMU FSM is CFG state.

44.4.3.5.1 Bi-Stable Protocol

The table below shows the Bi-Stable Protocol.

Table 44.25: Bi-Stable Protocol Table

Indication States	FMU_ERR_OUT[1:0] (FOUT0_POL=0,FOUT1_POL=0)	FMU_ERR_OUT[1:0] (FOUT0_POL=1,FOUT1_POL=1)
NONFAULTY	01b	10b
FAULTY	10b	01b
CFG(CFGST_HIMPD_EN=1)	High-Impedence	High-Impedence
CFG(FAULT_INDICATION=0)	01b	10b
CFG(FAULT_INDICATION=1)	10b	01b

44.4.3.5.2 Fault Output Counter

In Bi-Stable mode, the fault output counter starts when all of the following are true:

- A fault's FMU reaction is configured to fault indication output. Note FMU_ERR_INn doesn't need this configuration.
- The fault output counter is not already running.
- FMU FSM enters the FAULT state.

When the fault output counter is already running and a new fault occurs:

- If FMU is in the CFG state: FMU does not restart the counter.
- If FMU is in the NORMAL or ALARM or FAULT state: FMU restarts the counter if the fault's alarm is disabled or the fault is from FMU_ERR_IN.

FMU stops and initializes the fault output counter when all of the following are true:

- The fault output counter has expired(refer to FOUT_MIN_WIN of FOUT_CFG).
- All faults that caused FMU to enter or remain in the FAULT state have recovered, and FMU FSM returns to a NON-FAULT state.

44.4.3.6 Interrupt Generation

FMU generates interrupt sources in the following table to NVIC.

Table 44.26: FMU Interrupt Source Table

Interrupt Name	Descriptions	Clock domain
CFG Timeout Interrupt	Occurs when CFG counter timeout	FMU bus clock
ALARM Interrupt	Occurs when FSM transition to ALARM state, and ALARM_INT_EN bit is set to 1	FMU func clock
FAULT Interrupt(Nonmaskable)	Occurs when FSM transition to FAULT state, and FAULT_INT_EN bit is set to 1	FMU func clock

44.4.3.7 FMU Reset Generation

If FMU FSM moves to FAULT state and the fault's FMU reaction has been configured to reset, FMU will generate a reset to MCU.

Yuntu Microelectronics

44.4.4 FFTR(FMU Fail To React)

FFTR is used to detect the event when FMU fails to react when it should generate interrupts to NVIC, output fault indication or reset MCU. FFTR monitors the fault input and the FMU reaction signals and generates a FFTR reset if FMU has no reaction.

44.4.4.1 FFTR Counter

FFTR contains an independent counter driven by FFTR_CLK. The counter is not started if one of the conditions below meets:

- fault configured to be disabled(EN bit of FCH_CTRLn is 0)
- fault enabled but FMU reaction disabled(any of ALARM_INT_EN, FAULT_INT_EN, FAULT_RST_EN, FOUT_EN is not set).
- FMU FSM is CFG state

If FMU has fault enabled and FMU reaction enabled, the FFTR counter will be started when a fault occurs. The FFTR counter will be reset by module reset. And it will be stopped and cleared if one of the conditions below occurs:

- there's an ALARM interrupt
- there's a FAULT interrupt
- FMU_ERR_OUT toggles

44.4.4.2 FFTR Reset Generation

After a enabled fault occurs, FFTR counter starts running, if there's no FMU reaction within a preset time window(refer to FFTR_CNT_TO_TH of FFTR_CTRL), FFTR reset will be issued.

44.5 Glossary

FFTR	FMU Fail To React
NMI	Non-maskable Interrupt
FSM	Finite state machine

Yuntu Microelectronics

Part X

Appendix

Yuntu Microelectronics

Appendix A

A.1 Revision History

Table A.1: Reference Manual Revision History

Rev.No.	Release Date
1.0	2023/9/26
1.1	2024/2/27
1.2	2024/6/28
1.3	2026/4/21

Yuntu Microelectronics

A.2 Release Notes

A.2.1 Introduction Changes

- No substantial content changes.

A.2.2 Memory Map and Memories Changes

A.2.2.1 Memory Map Changes

- No substantial content changes.

A.2.2.2 Embedded FLASH Module (EFM) Changes

- Updated the function description of EFM STS Register bit 15 (BOOT_INFO).

A.2.2.3 Read-Only Memory (ROM) Changes

- Removed some non-existent addresses in subsection of Location of BVT.

A.2.2.4 On-Chip RAM (OCRAM) Changes

- No substantial content changes.

A.2.2.5 Register File (REGFILE) Changes

- No substantial content changes.

Yuntu Microelectronics

A.2.3 Signal multiplexing Changes

A.2.3.1 Pinmux Changes

- No substantial content changes.

A.2.3.2 General Purpose Input/Output (GPIO) Changes

- No substantial content changes.

A.2.3.3 Port Control (PCTRL) Changes

- No substantial content changes.

A.2.4 Core and System Changes

A.2.4.1 CM7 Overview Changes

- Corrected the number of hardware breakpoint comparators and watchpoint comparators in section of JTAG Controller Configuration.

A.2.4.2 Direct Memory Access (DMA) Changes

- Updated the offset of registers in subsection of DMA Memory Map.
- Opened the bit 17 (PAUSE) in subsubsection of DMA CTRL Register.

A.2.4.3 Chip Integration Module (CIM) Changes

- No substantial content changes.

A.2.4.4 Trigger Multiplexer Module (TMU) Changes

- Removed SPI trigger output in section of Trigger Interconnectivity Table.

Yuntu Microelectronics

A.2.5 Clock,Reset and Power Management Changes

A.2.5.1 System Clock Unit (SCU) Changes

- Updated the function description of bit field GMSEL in subsubsection of SCU FXOSC_CTRL Register.

A.2.5.2 IP Control (IPC) Changes

- No substantial content changes.

A.2.5.3 Reset Control Unit (RCU) Changes

- No substantial content changes.

A.2.5.4 Power Control Unit (PCU) Changes

- No substantial content changes.

A.2.5.5 Power Mode Changes

- No substantial content changes.

A.2.5.6 Wake-up Unit (WKU) Changes

- No substantial content changes.

Yuntu Microelectronics

A.2.6 Communication Interfaces Changes

A.2.6.1 Flexible Controller Area Network (FlexCAN) Changes

- Updated the offset and register name of WMB in subsection of CAN Memory Map.
- Updated the function description about DMA Enable in bit field of 15 in subsubsection of CAN MCR Register.
- Corrected the function description of bit field ERFEN in subsubsection of CAN ERFCR Register,
- Updated the document description in section of Functional Description, Initialization/Application Information, FlexCAN RAM Usage and Initialization.

A.2.6.2 Local Interconnect Network (LIN) Changes

- Updated the offset of registers in subsection of LINFlexD Memory Map.

A.2.6.3 Serial Port Interface (SPI) Changes

- Updated the table description in subsection of Siganl Description.

A.2.6.4 Quad Serial Peripheral Interface (QSPI) Changes

- Updated two reserved registers to QSPI SFA1TA Register and QSPI SFB1TA Register in subsection of QSPI Memory Map.

A.2.6.5 Inter-Integrated Circuit (I2C) Changes

- No substantial content changes.

A.2.6.6 Single Edge Nibble Transmission (SENT) Changes

- No substantial content changes.

Yuntu Microelectronics

A.2.6.7 Serial Audio Interface (SAI) Changes

- Corrected the function description of bit field FPACK in subsubsection of SAI TFCR Register.
- Removed the bit field BCI in subsubsection of SAI TCFG0 Register.
- Removed the bit field BCI in subsubsection of SAI RCFG0 Register.

A.2.7 Timer Changes

A.2.7.1 Timer (TMR) Changes

- No substantial content changes.

A.2.7.2 Periodic Timer (pTMR) Changes

- Updated the offset of registers in subsection of pTMR Memory Map.

A.2.7.3 Low Power Timer (lpTMR) Changes

- No substantial content changes.

A.2.7.4 Enhanced Timer (eTMR) Changes

- Corrected the table about hall sensor information in subsection of Instantiation Information.
- Updated the offset of registers in subsection of EMU Memory Map.

A.2.7.5 Multiple Pulse Width Modulation (MPWM) Changes

- Updated the offset of registers in subsection of MPWM Memory Map.
- Corrected the channel number description in subsubsection of MPWM STS1 Register.

A.2.7.6 Real Time Clock (RTC) Changes

- No substantial content changes.

Yuntu Microelectronics

A.2.7.7 Programmable Trigger Unit (PTU) Changes

- Updated the offset of registers in subsection of PTU Memory Map.

A.2.8 Analog Changes

A.2.8.1 Analog-to-Digital Converter (ADC) Changes

- Corrected the function description of ADC CFG0 register bit 22 (LPEN).

A.2.8.2 Analog Comparator (ACMP) Changes

- No substantial content changes.

A.2.8.3 Temperature Sensor (TEMPSENSOR) Changes

- No substantial content changes.

A.2.9 Security, Integrity and Safety Changes

A.2.9.1 Programmable Cyclic Redundancy Check (PCRC) Changes

- Updated the offset of registers in subsection of EMU Memory Map.

A.2.9.2 Hardware Cryptography Unit (HCU) Changes

- No substantial content changes.

A.2.9.3 Ture Random Number Generator (TRNG) Changes

- No substantial content changes.

Yuntu Microelectronics

A.2.9.4 Watch Dog (WDG) Changes

- No substantial content changes.

A.2.9.5 ECC Management Unit (EMU) Changes

- Updated the offset of registers in subsection of EMU Memory Map.
- Corrected the descrption in subsection of Error Injection.

A.2.9.6 Interrupt Monitor (INTM) Changes

- Updated the offset of registers in subsection of INTM Memory Map.

A.2.9.7 Peripheral Protection Unit (PPU) Changes

- No substantial content changes.

A.2.9.8 Fault Management Unit (FMU) Changes

- Updated both FMU FAULT_ST0 Register and FMU FAULT_ST1 Register to one FMU FAULT_ST Register.

Yuntu Microelectronics

Copyright and Contact

Information in this document is provided solely to enable system and software implementers to use Yuntu Microelectronics products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. Yuntu Microelectronics reserves the right to change the detail specifications as may be required to permit improvements in the design of its products.

Yuntu Microelectronics makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Yuntu Microelectronics assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Yuntu Microelectronics data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Yuntu Microelectronics does not convey any license under its patent rights nor the rights of others. Yuntu Microelectronics sells products pursuant to standard terms and conditions of sale, which can be found at the following address: www.ytmicro.com

While Yuntu Microelectronics has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer’s applications and products, and Yuntu Microelectronics accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Yuntu Microelectronics reserves the right to change the production detail specifications as may be required to permit improvements in the design of its products.

©2020 - 2026 Jiangsu Yuntu Microelectronics Co., LTD.

How to reach us:
Home Page: www.ytmicro.com

Yuntu Microelectronics