TLE987x

Microcontroller with LIN and BLDC MOSFET driver for automotive applications
BE, BF, A, and UH steps

User manual

About this document

Scope and purpose

This user manual describes the functionality of the superset device of the TLE987x Embedded Power IC family. For the available functionality of a specific TLE987x product, please refer to the respective datasheet. For simplicity, the products are referenced by the collective term TLE987x throughout this manual.

Intended audience

This user manual is intended for embedded hardware and software developers. It provides detailed descriptions about the behavior of the TLE987x functional units and their interactions.

1 TLE987x products

This chapter lists the product variants to which this document applies.

1.1 TLE987x grade 1 derivatives

The following table lists the grade 1 derivatives.

Table 1 Grade 1 derivatives

Product name	Orderable part number (OPN)	Design step	Max. operating frequency [MHz]	Flash size [KB]	RAM size [KB]	Package
TLE9871QXA20	TLE9871QXA20XUMA1	BE	24	36	3	VQFN-48
	TLE9871QXA20XUMA2	BF				
	TLE9871QXA20XUMA3	UH				
TLE9877QXA20	TLE9877QXA20XUMA1	BE	24	64	6	VQFN-48
	TLE9877QXA20XUMA2	BF				
	TLE9877QXA20XUMA3	UH				
TLE9877QXA40	TLE9877QXA40XUMA1	BE	40	64	6	VQFN-48
	TLE9877QXA40XUMA2	BF				
	TLE9877QXA40XUMA3	UH				
TLE9879QXA20	TLE9879QXA20XUMA1	BE	24	128	6	VQFN-48
	TLE9879QXA20XUMA2	BF				
	TLE9879QXA20XUMA3	UH				

Table 1 Grade 1 derivatives (cont'd)

Product name	Orderable part number (OPN)	Design step	Max. operating frequency [MHz]	Flash size [KB]	RAM size [KB]	Package
TLE9879QXA40	TLE9879QXA40XUMA1	BE	40	128	6	VQFN-48
	TLE9879QXA40XUMA2	BF				
	TLE9879QXA40XUMA3	UH				
TLE9879-2QXA40	TLE98792QXA40XUMA1	BF	40	128	6	VQFN-48
	TLE98792QXA40XUMA2	UH				
TLE9872QXA40	TLE9872QXA40XUMA1	A	40	256	8	VQFN-48
TLE9872-2QXA40	TLE98722QXA40XUMA1	A	40	256	8	VQFN-48

1.2 TLE987x grade 0 derivatives

The following table lists the grade 0 derivatives.

Table 2 Grade 0 derivatives

Product name	Orderable part number (OPN)	Design step	Max. operating frequency [MHz]	Flash size [KB]	RAM size [KB]	Package
TLE9873QXW40	TLE9873QXW40XUMA1	BF	40	48	3	VQFN-48
	TLE9873QXW40XUMA2	UH				
TLE9877QXW40	TLE9877QXW40XUMA1	BF	40	64	6	VQFN-48
	TLE9877QXW40XUMA2	UH				
TLE9877QTW40	TLE9877QTW40XUMA1	BF	40	64	6	TQFP-48
TLE9879QXW40	TLE9879QXW40XUMA1	BF	40	128	6	VQFN-48
	TLE9879QXW40XUMA2	UH				
TLE9879QTW40	TLE9879QTW40XUMA1	BF	40	128	6	TQFP-48
TLE9872QTW40	TLE9872QTW40XUMA1	A	40	256	8	TQFP-48

Table of contents

1 TLE987x products . . . 1
1.1 TLE987x grade 1 derivatives . . . 1
1.2 TLE987x grade 0 derivatives . . . 2

2 Overview . . . 16
2.1 Abbreviations . . . 18

3 Modes of operation . . . 21

4 Device register types . . . 24

5 Power management unit (PMU) . . . 25
5.1 Features . . . 25
5.2 Introduction . . . 25
5.2.1 Block diagram . . . 26
5.2.2 PMU modes overview . . . 28
5.3 Power supply generation unit (PGU) . . . 33
5.3.1 Voltage regulator 5.0 V (VDDP) . . . 34
5.3.2 Voltage regulator 1.5 V (VDDC) . . . 35
5.3.3 External voltage regulator 5.0 V (VDDEXT) . . . 36
5.3.4 Low-VS operation . . . 38
5.3.5 PMU register overview . . . 39
5.3.6 Register definition . . . 39
5.3.6.1 Power Supply Generation Register . . . 40
5.3.6.2 VDDEXT Control Register . . . 43
5.4 Power control unit . . . 45
5.4.1 Power control unit – fail-safe scenarios . . . 45
5.4.1.1 Power supervision function of the PMU-PCU . . . 45
5.4.1.2 Watchdog (WDT1) fail safe . . . 45
5.4.1.3 Main regulator fail safe . . . 46
5.4.1.4 VDDEXT failure . . . 46
5.4.1.5 Wake-up from Stop mode with reset fail-safe . . . 46
5.4.2 Register Definition . . . 48
5.4.2.1 PMU System Fail Register . . . 48
5.4.2.2 PMU Wake Fail Register . . . 50
5.5 Wake-up management unit (PMU-WMU) . . . 51
5.5.1 Functional description . . . 51
5.5.2 Register Definition . . . 52
5.5.2.1 PMU Wake-Up Configuration Register . . . 53
5.5.2.2 PMU Wake-Up Status Register . . . 61
5.6 Cyclic management unit (PMU-CMU) . . . 65
5.6.1 Functional description . . . 65
5.6.2 Cyclic Sense mode . . . 65
5.6.2.1 Configuring cyclic sense mode . . . 67
5.6.3 Cyclic Wake mode . . . 68
5.6.4 Register Definition . . . 68
5.6.4.1 Cyclic Wake Configuration Registers . . . 68
5.7 Reset management unit (PMU-RMU) . . . 73
5.7.1 Functional description . . . 73

5.7.2 Register Definition . . . 79
5.7.2.1 Reset Management Unit Registers (RMU) . . . 79
5.8 PMU data storage area . . . 83
5.8.1 Functional description . . . 83
5.8.2 Register definition . . . 83
5.8.2.1 Data Storage Registers . . . 84
5.9 Register access notes . . . 88

6 System control unit – digital modules (SCU-DM) . . . 89
6.1 Features . . . 89
6.2 Introduction . . . 89
6.2.1 Block diagram . . . 90
6.2.2 SCU register overview . . . 91
6.2.2.1 Register map . . . 91
6.3 Clock generation unit . . . 94
6.3.1 Low-precision clock . . . 94
6.3.2 High-precision oscillator circuit (OSC_HP) . . . 94
6.3.2.1 External Input Clock mode . . . 95
6.3.2.2 External Crystal mode . . . 95
6.3.3 Phase-locked loop (PLL) module . . . 96
6.3.3.1 Features . . . 96
6.3.3.2 PLL functional description . . . 97
6.3.3.3 Oscillator watchdog . . . 101
6.3.3.4 PLL VCO lock detection . . . 101
6.3.3.5 Internal oscillator (OSC_PLL) . . . 102
6.3.3.6 Switching PLL parameters . . . 102
6.3.3.7 Oscillator watchdog event or PLL loss-of-lock detection . . . 103
6.3.3.8 Oscillator watchdog event or loss-of-lock recovery . . . 103
6.3.4 Clock control unit . . . 105
6.3.4.1 Clock tree . . . 106
6.3.4.2 Startup control for system clock . . . 107
6.3.5 External clock output . . . 107
6.3.6 CGU registers . . . 108
6.3.6.1 PLL oscillator register . . . 108
6.3.6.2 PLL registers . . . 110
6.3.6.3 System clock control registers . . . 115
6.3.6.4 Analog Peripherals Clock Control registers . . . 116
6.3.6.5 External Clock Control register . . . 122
6.4 Reset control . . . 124
6.4.1 Types of reset . . . 124
6.4.2 Overview . . . 124
6.4.3 Module reset behavior . . . 124
6.4.4 Functional descriptions of reset types . . . 126
6.4.4.1 Power-on and brown-out reset . . . 126
6.4.4.2 Wake-up reset . . . 126
6.4.4.3 Hardware reset . . . 126
6.4.4.4 WDT1 reset . . . 126
6.4.4.5 WDT and soft reset . . . 126

6.4.5 Reset register description . . . 127
6.4.6 Booting scheme . . . 128
6.5 Power management . . . 129
6.5.1 Overview . . . 129
6.5.2 Functional description . . . 130
6.5.2.1 Slow-Down mode . . . 130
6.5.2.2 Stop mode . . . 130
6.5.2.3 Sleep mode . . . 132
6.5.3 Register description . . . 133
6.6 Interrupt management . . . 134
6.6.1 Overview . . . 134
6.6.1.1 External interrupts . . . 134
6.6.1.2 Extended interrupts . . . 135
6.6.2 Interrupt node assignment . . . 135
6.6.3 Interrupt-related registers . . . 137
6.6.3.1 Interrupt event enable control . . . 137
6.6.3.2 DMA interrupt event enable and select control . . . 141
6.6.4 NMI event flag handling . . . 146
6.7 General port control . . . 147
6.7.1 Input pin function selection . . . 147
6.7.2 Port output control . . . 151
6.7.3 GPT12 T3IN and T4IN input pin function selection . . . 158
6.8 Flexible peripheral management . . . 159
6.8.1 Peripheral management registers . . . 159
6.9 Module suspend control . . . 161
6.10 Watchdog timer . . . 163
6.10.1 Functional description . . . 164
6.10.2 Register description . . . 166
6.11 Error detection and correction control for memory . . . 169
6.11.1 Error Detection and Correction Control register . . . 169
6.11.2 Error Detection and Correction Status register . . . 170
6.12 Miscellaneous control . . . 172
6.12.1 Bit-protection register . . . 172
6.12.2 System control and status registers . . . 174
6.13 Register access notes . . . 181

7 System control unit – power modules (SCU-PM) . . . 182
7.1 Features . . . 182
7.2 Introduction . . . 182
7.2.1 Block diagram . . . 182
7.3 Clock Watchdog Unit (CWU) . . . 183
7.3.1 Fail Safe Functionality of Clock Generation Unit (Clock Watchdog) . . . 183
7.3.1.1 Functional Description of Clock Watchdog Module . . . 184
7.3.2 Clock Generation Unit Register . . . 185
7.4 Interrupt Control Unit (ICU) . . . 190
7.4.1 Structure of PREWARN_SUP_NMI . . . 190
7.4.2 Interrupt Control Unit Status Register . . . 191
7.4.2.1 Interrupt Control Unit Status Overview Register . . . 191

7.4.2.2 Interrupt Control Unit - Interrupt Enable Register . . . 208
7.5 Power Control Unit for Power Modules (PCU_PM) . . . 215
7.5.1 VSD-Overvoltage System Shutdown . . . 216
7.5.2 Overtemperature System Shutdown . . . 216
7.5.3 Power Control Unit Register . . . 218
7.6 MISC Control . . . 220
7.6.1 Bus Faults . . . 220
7.6.1.1 Bus Fault Control Register . . . 220
7.7 Register Access Hints . . . 224

8 Arm® Cortex®-M3 core . . . 225
8.1 Features . . . 225
8.2 Introduction . . . 226
8.2.1 Block diagram . . . 226
8.3 Functional Description . . . 227
8.3.1 Registers . . . 227
8.3.1.1 General-Purpose Registers . . . 227
8.3.1.2 Special-Purpose Program Status Registers (xPSR) . . . 228
8.4 Summary of Processor Registers . . . 229
8.5 Instruction Set Summary . . . 269

9 DMA controller . . . 273
9.1 Features . . . 273
9.2 Introduction . . . 274
9.2.1 Block diagram . . . 274
9.3 Functional description . . . 275
9.3.1 DMA mode overview . . . 275
9.3.2 DMA Arbitration Rate . . . 277
9.3.3 Channel Control Data Structure . . . 290
9.4 Register Definition . . . 301

10 Address space organization . . . 318

11 Memory control unit . . . 322
11.1 Features . . . 322
11.2 Introduction . . . 322
11.2.1 Block diagram . . . 322
11.3 NVM module (flash memory) . . . 324
11.3.1 Definitions . . . 325
11.3.1.1 General Definitions . . . 325
11.3.2 Functional Description . . . 327
11.3.2.1 Basic Block Functions . . . 327
11.3.2.2 Memory Cell Array . . . 328
11.3.2.3 SFR Accesses . . . 329
11.3.2.4 Memory Read . . . 329
11.3.2.5 Memory Write . . . 329
11.3.2.6 Timing . . . 330
11.3.2.7 Verify . . . 330
11.3.2.8 Tearing-Safe programming . . . 330
11.3.2.9 Disturb handling . . . 330

11.3.2.10 ECC and EDC . . . 330
11.3.2.11 Resume from disturbed Program/Erase operation . . . 331
11.3.2.12 Code and Data Access through the AHB-Lite Interface . . . 331
11.3.2.13 Prefetch Unit . . . 331
11.4 BootROM Module . . . 332
11.4.1 BootROM addressing . . . 332
11.4.2 BootROM firmware program structure . . . 332
11.5 RAM Module . . . 332
11.5.1 RAM Addressing . . . 333
11.6 Memory Protection Unit (MPU) . . . 333

12 Interrupt system . . . 334
12.1 Features . . . 334
12.2 Introduction . . . 334
12.2.1 Overview . . . 334
12.3 Functional Description . . . 336
12.3.1 Interrupt Node Assignment . . . 336
12.3.1.1 Interrupt Node 0 and 1 - GPT12 Timer Module . . . 336
12.3.1.2 Interrupt Node 2 - Measurement Unit . . . 337
12.3.1.3 Interrupt Node 3 - 10 Bit ADC (ADC1) . . . 338
12.3.1.4 Interrupt Nodes 4/5/6/7 - Capture Compare Unit (CCU6) . . . 339
12.3.1.5 Interrupt Node 8 and 9 - SSC1/SSC2 . . . 340
12.3.1.6 Interrupt Node 10 - UART1/LIN Transceiver . . . 341
12.3.1.7 Interrupt Node 11 - UART2 . . . 342
12.3.1.8 Interrupt Node 12 - External Interrupt 0/MON . . . 343
12.3.1.9 Interrupt Node 13 - External Interrupt 1 . . . 344
12.3.1.10 Interrupt Node 14 - Bridge Driver . . . 345
12.3.1.11 Interrupt Node 15 - DMA Controller . . . 346
12.3.1.12 Non-Maskable Interrupt (NMI) . . . 347
12.3.1.13 Interrupt Flags Overview . . . 348
12.3.2 Interrupt Structure . . . 357
12.3.3 Interrupt Source and Vector . . . 357
12.3.4 Interrupt Priority . . . 359
12.4 Interrupt Handling . . . 361
12.5 Register Definition . . . 362
12.5.1 Interrupt Node Enable Registers . . . 362
12.5.2 External Interrupt Control Registers . . . 364
12.5.3 Interrupt Flag Registers . . . 365
12.6 Interrupt Priority Registers . . . 381

13 Watchdog timer (WDT1) . . . 382
13.1 Features . . . 382
13.2 Introduction . . . 383
13.3 Functional Description . . . 384
13.3.1 Modes of Operation . . . 384
13.3.2 Normal Operation . . . 384
13.4 Register Definition . . . 387

14 GPIO ports and peripheral I/O . . . 390

14.1 Features . . . 390
14.2 Introduction . . . 390
14.2.1 Port 0 and port 1 . . . 390
14.2.2 Port 2 . . . 393
14.3 General Port Register Description . . . 394
14.3.1 Port Data Register . . . 395
14.3.2 Direction Register . . . 395
14.3.3 Open Drain Control Register . . . 396
14.3.4 Pull-Up/Pull-Down Device Register . . . 397
14.3.5 Alternate Input Functions . . . 398
14.3.6 Alternate Output Functions . . . 398
14.3.7 Register Map . . . 399
14.4 TLE987x port module . . . 400
14.4.1 Port 0 . . . 400
14.4.1.1 Overview . . . 400
14.4.1.2 Port 0 functions . . . 401
14.4.1.3 Port 0 Register Description . . . 403
14.4.2 Port 1 . . . 407
14.4.2.1 Overview . . . 407
14.4.2.2 Port 1 functions . . . 407
14.4.2.3 Port 1 Register Description . . . 410
14.4.3 Port 2 . . . 414
14.4.3.1 Overview . . . 414
14.4.3.2 Port 2 functions . . . 414
14.4.3.3 Port 2 Register Description . . . 416
14.5 Alternate Function Map . . . 420

15 General-purpose timer units (GPT12) . . . 421
15.1 Features . . . 421
15.1.1 Features of block GPT1 . . . 421
15.1.2 Features of block GPT2 . . . 421
15.2 Introduction . . . 422
15.2.1 Block diagram of GPT1 . . . 423
15.2.2 Block diagram of GPT2 . . . 424
15.3 Timer Block GPT1 . . . 425
15.3.1 GPT1 Core Timer T3 Control . . . 426
15.3.2 GPT1 Core Timer T3 Operating Modes . . . 428
15.3.3 GPT1 Auxiliary Timers T2/T4 Control . . . 434
15.3.4 GPT1 Auxiliary Timers T2/T4 Operating Modes . . . 435
15.3.5 GPT1 Clock Signal Control . . . 443
15.3.6 Interrupt Control for GPT1 Timers . . . 445
15.3.7 GPT1 Registers . . . 446
15.3.7.1 GPT1 Timer Registers . . . 446
15.3.7.2 GPT1 Timer Control Registers . . . 447
15.3.7.3 GPT1 Timer Interrupt Control Registers . . . 455
15.4 Timer Block GPT2 . . . 456
15.4.1 GPT2 Core Timer T6 Control . . . 457
15.4.2 GPT2 Core Timer T6 Operating Modes . . . 459

15.4.3 GPT2 Auxiliary Timer T5 Control . . . 462
15.4.4 GPT2 Auxiliary Timer T5 Operating Modes . . . 463
15.4.5 GPT2 Register CAPREL Operating Modes . . . 467
15.4.6 GPT2 Clock Signal Control . . . 472
15.4.7 Interrupt Control for GPT2 Timers and CAPREL . . . 474
15.4.8 GPT2 Registers . . . 475
15.4.8.1 GPT2 Timer Registers . . . 475
15.4.8.2 GPT2 Timer Control Registers . . . 476
15.4.8.3 GPT2 Timer and CAPREL Interrupt Control Registers . . . 481
15.5 Miscellaneous GPT12 Registers . . . 482
15.6 Register Map . . . 484
15.7 Implementation of the GPT12 Module . . . 485
15.7.1 Module Connections . . . 485

16 Timer2 and Timer21 . . . 488
16.1 Features . . . 488
16.2 Introduction . . . 488
16.2.1 Timer2 and Timer21 mode overview . . . 489
16.3 Functional Description . . . 490
16.3.1 Auto-Reload Mode . . . 490
16.3.1.1 Up/Down Count Disabled . . . 490
16.3.1.2 Up/Down Count Enabled . . . 491
16.3.2 Capture Mode . . . 493
16.3.3 Count Clock . . . 494
16.4 Module Interfaces . . . 495
16.4.1 Interfaces of the Timer2 and Timer21 . . . 495
16.5 Interrupt Generation . . . 496
16.6 Register Definition . . . 497
16.6.1 Timer2 Registers . . . 497
16.6.1.1 Mode Register . . . 497
16.6.1.2 Control Register . . . 498
16.6.1.3 Timer2 Reload/Capture Register . . . 500
16.6.1.4 Timer2 Count Register . . . 501
16.6.2 Register Map . . . 502

17 Timer3 . . . 503
17.1 Features . . . 503
17.2 Introduction . . . 503
17.3 Functional description . . . 503
17.3.1 Timer3 modes overview . . . 504
17.3.2 Mode 0 . . . 506
17.3.3 Mode 1a . . . 507
17.3.4 Mode 1b . . . 508
17.3.5 Mode 2 . . . 509
17.3.6 Mode 3a . . . 510
17.3.7 Mode 3b . . . 511
17.4 Interrupts . . . 512
17.5 Register Definition . . . 513
17.5.1 Timer3 Control Registers . . . 514

18 Capture/compare unit 6 (CCU6) . . . 522
18.1 Feature set overview . . . 522
18.2 Introduction . . . 523
18.2.1 Block diagram . . . 524
18.3 Operating Timer T12 . . . 525
18.3.1 T12 Overview . . . 526
18.3.2 T12 Counting Scheme . . . 528
18.3.2.1 Clock Selection . . . 528
18.3.2.2 Edge-Aligned / Center-Aligned Mode . . . 529
18.3.2.3 Single-Shot Mode . . . 531
18.3.3 T12 Compare Mode . . . 532
18.3.3.1 Compare Channels . . . 532
18.3.3.2 Channel State Bits . . . 533
18.3.3.3 Hysteresis-Like Control Mode . . . 537
18.3.4 Compare Mode Output Path . . . 538
18.3.4.1 Dead-Time Generation . . . 538
18.3.4.2 State Selection . . . 540
18.3.4.3 Output Modulation and Level Selection . . . 541
18.3.5 T12 Capture Modes . . . 543
18.3.6 T12 Shadow Register Transfer . . . 546
18.3.7 Timer T12 Operating Mode Selection . . . 547
18.4 Operating Timer T13 . . . 548
18.4.1 T13 Overview . . . 548
18.4.2 T13 Counting Scheme . . . 550
18.4.2.1 Clock Selection . . . 550
18.4.2.2 T13 Counting . . . 551
18.4.2.3 Single-Shot Mode . . . 551
18.4.2.4 Synchronization to T12 . . . 552
18.4.3 T13 Compare Mode . . . 553
18.4.4 Compare Mode Output Path . . . 554
18.4.5 T13 Shadow Register Transfer . . . 556
18.5 Trap Handling . . . 557
18.6 Multi-Channel Mode . . . 559
18.7 Hall Sensor Mode . . . 561
18.7.1 Hall Pattern Evaluation . . . 562
18.7.2 Hall Pattern Compare Logic . . . 564
18.7.3 Hall Mode Flags . . . 565
18.7.4 Hall Mode for Brushless DC-Motor Control . . . 566
18.8 General Module Operation . . . 568
18.8.1 Input Selection . . . 568
18.9 Module Interfaces . . . 568
18.9.1 Interfaces of the CCU6 Module . . . 568
18.10 Interrupt Handling . . . 570
18.10.1 Interrupt Structure . . . 570
18.11 Register Definition . . . 572
18.11.1 CCU6 Register Description . . . 572
18.11.1.1 System Registers . . . 574

18.11.2 Timer12 – Related Registers . . . 577
18.11.3 Timer13 – Related Registers . . . 585
18.11.4 Capture/Compare Control Registers . . . 587
18.11.5 Global Modulation Control Registers . . . 597
18.11.6 Multi-Channel Modulation Control Registers . . . 602
18.11.7 Interrupt Control Registers . . . 607
18.11.8 Register Map . . . 617

19 UART1/UART2 . . . 619
19.1 Features . . . 619
19.2 Introduction . . . 619
19.2.1 Block diagram . . . 619
19.3 UART modes . . . 620
19.3.1 Mode 0, 8-Bit Shift Register, Fixed Baud Rate . . . 620
19.3.2 Mode 1, 8-Bit UART, Variable Baud Rate . . . 620
19.3.3 Mode 2, 9-Bit UART, Fixed Baud Rate . . . 622
19.3.4 Mode 3, 9-Bit UART, Variable Baud Rate . . . 622
19.4 Multiprocessor Communication . . . 624
19.5 Baud Rate Generation . . . 625
19.5.1 Baud-rate Generator . . . 625
19.6 LIN Support in UART . . . 627
19.6.1 LIN Protocol . . . 627
19.6.2 LIN Header Transmission . . . 628
19.6.2.1 Automatic Synchronization to the Host . . . 629
19.6.2.2 Initialization of Break/Synch Field Detection Logic . . . 630
19.6.2.3 Baud Rate Range Selection . . . 630
19.6.2.4 LIN Baud Rate Detection . . . 632
19.7 Module Interfaces . . . 634
19.8 Interrupts . . . 637
19.9 Register Definition . . . 638
19.9.1 UART Registers . . . 638
19.9.1.1 UART Control Registers . . . 638
19.9.1.2 Baud-Rate Generator Control and Status Registers . . . 640
19.9.1.3 Baud-Rate Generator Timer/Reload Registers . . . 643
19.9.2 Register Map . . . 646

20 LIN transceiver . . . 647
20.1 Features . . . 647
20.2 Introduction . . . 648
20.2.1 Block diagram . . . 648
20.3 Functional Description . . . 649
20.3.1 LIN Transceiver Modes . . . 649
20.3.2 LIN Transceiver Status for Mode Selection . . . 652
20.3.3 LIN Transceiver Error Handling . . . 652
20.3.4 Slope Modes . . . 653
20.3.5 LIN Transceiver Slope Mode Status . . . 653
20.4 Register Definition . . . 654
20.5 Register Access Hints . . . 657

21 High-speed synchronous serial interface (SSC1/SSC2) 658
21.1 Features 658
21.2 Introduction 659
21.2.1 Block diagram 659
21.3 Functional Description 660
21.3.1 SSC1 and SSC2 Modes Overview 660
21.3.2 Operating Mode Selection 661
21.3.3 Full-Duplex Operation 662
21.3.4 Half-Duplex Operation 664
21.3.5 Continuous Transfers 665
21.3.6 Baud Rate Generation 666
21.3.7 Error Detection Mechanisms 667
21.3.7.1 Port Control 669
21.4 Module Interfaces 669
21.4.1 Interfaces of the SSC Module 669
21.5 Interrupts 672
21.6 Register Definition 673
21.6.1 SSC1 and SSC2 Register Definition 673
21.6.1.1 Port Input Select Register 673
21.6.1.2 Configuration Register 674
21.6.1.3 Baud Rate Timer Reload Register 678
21.6.1.4 Transmitter Buffer Register 678
21.6.1.5 Receiver Buffer Register 679
21.6.2 Register Map 679

22 Measurement unit 680
22.1 Features 680
22.2 Introduction 680
22.2.1 Block diagram 682
22.2.1.1 BEMF comparator block diagram 683
22.2.2 Measurement Unit Register Overview 684
22.3 8-bit - 10 Channel ADC Core (ADC2) 684
22.3.1 8-bit ADC Channel Allocation 685
22.3.2 Transfer Characteristics of ADC2 686
22.3.3 8-bit - 10 Channel Control Registers 686
22.4 ADC Input Selection 687
22.4.1 Functional Description 687
22.4.2 Port 2 ADC Selection Register 687
22.5 VDH Input Voltage Attenuator 689
22.5.1 Functional Description 689
22.5.2 VDH Input Attenuator Control Register 690
22.6 BEMF Comparators 692
22.6.1 Functional Description 692
22.6.1.1 Postprocessing of BEMF Comparator Signals 693
22.6.2 BEMF Comparators Control and Status Register 695
22.7 Central and PMU Regulator Temperature Sensor 699
22.7.1 Functional Description 699
22.7.2 Temperature Sensor Control Register 701

22.8 Supplement Modules . . . 702
22.8.1 Functional Description . . . 702
22.8.1.1 ADC2 Functional Safety Feature . . . 702
22.8.1.2 ADC1 reference voltage (V_{AREF}) . . . 702
22.8.2 Supplement Modules Control and Status Register . . . 703

23 Measurement core module (incl. ADC2) . . . 705
23.1 Features . . . 705
23.2 Introduction . . . 705
23.2.1 Block diagram . . . 705
23.2.2 Measurement core module mode overview . . . 706
23.3 ADC2 - Core (8-bit ADC) . . . 707
23.3.1 Functional Description . . . 707
23.3.2 ADC2 Control Registers . . . 708
23.4 Channel Controller . . . 711
23.4.1 Functional Description . . . 711
23.4.2 Channel Controller Control Registers . . . 713
23.5 Calibration Unit . . . 731
23.5.1 Functional Description . . . 731
23.5.1.1 Method for determining the Calibration Parameters . . . 731
23.5.1.2 Setup of Calibration Unit . . . 731
23.5.2 Calibration Unit Control Registers . . . 733
23.6 IIR-Filter . . . 739
23.6.1 Functional Description . . . 739
23.6.1.1 Step Response . . . 740
23.6.2 IIR Filter Control Registers . . . 741
23.7 Signal Processing . . . 755
23.7.1 Functional Description . . . 755
23.7.2 Post-processing Control Registers . . . 759
23.8 Start-up Behavior after Reset . . . 779
23.9 Post-processing Default Values . . . 779

24 10-bit analog-to-digital converter (ADC1) . . . 781
24.1 Features . . . 781
24.2 Introduction . . . 781
24.2.1 Block diagram . . . 782
24.2.2 ADC1 Modes Overview . . . 783
24.3 ADC1 - Core (10-Bit ADC) . . . 783
24.3.1 Functional Description . . . 783
24.3.2 ADC1 Control and Status Registers . . . 785
24.4 Channel Controller . . . 789
24.4.1 Functional Description . . . 789
24.4.2 Channel Controller Control Registers . . . 791
24.5 Conversion Result Handling . . . 802
24.5.1 Functional Description . . . 802
24.5.2 Result Registers . . . 802
24.6 DMA Requests . . . 815
24.7 Interrupts . . . 815
24.7.1 Interrupt Registers . . . 817

24.8 Module Interfaces . . . 822
24.8.1 ADC1 Hardware Trigger Selection . . . 822

25 14-Bit Sigma Delta ADC (ADC3 / ADC4) . . . 823
25.1 Features . . . 823
25.2 Functional description . . . 823
25.2.1 Input voltage range of SDADC . . . 825
25.2.2 Calculation of SDADC output code . . . 825
25.2.3 Offset Compensation . . . 825
25.3 SDADC Control Register . . . 826

26 High-voltage monitor input . . . 830
26.1 Features . . . 830
26.2 Introduction . . . 830
26.2.1 Block diagram . . . 830
26.3 Functional Description . . . 831
26.4 Register Definition . . . 833
26.4.1 Monitor Input Registers . . . 833

27 Bridge driver (incl. charge pump) . . . 835
27.1 Features . . . 835
27.2 Introduction . . . 835
27.2.1 Block diagram . . . 836
27.2.2 General . . . 836
27.3 Functional Description . . . 837
27.3.1 Slew Rate Control . . . 837
27.3.2 PWM Operation . . . 839
27.3.3 Adjustable Cross-Conduction Protection . . . 840
27.3.4 Supply-Generation of Driver . . . 841
27.3.4.1 Clock Generator of Driver Supply . . . 842
27.3.5 Overcurrent Detection - VDS measurement . . . 842
27.3.6 OFF-State Short Circuit Detection . . . 843
27.3.7 Channel turn-on/turn-off delay measurement . . . 844
27.4 Application Hints . . . 845
27.4.1 Bridge Driver Diagnosis . . . 845
27.4.1.1 Off-Diagnosis . . . 845
27.4.1.2 On-Diagnosis . . . 847
27.4.1.3 Open-Load Detection . . . 849
27.5 SHx pin voltage level . . . 850
27.6 Register Definition . . . 851
27.6.1 Driver Register . . . 852
27.6.2 Driver Trimming Register . . . 865
27.6.3 Charge Pump Control and Status Register . . . 870

28 Current sense amplifier . . . 874
28.1 Features . . . 874
28.2 Introduction . . . 874
28.2.1 Block diagram . . . 875
28.2.2 Recommended offset calibration . . . 876
28.3 Functional Description . . . 877

28.3.1 ADC Code Calculation 877
28.4 Register Definition 878
29 Revision History 880

2 Overview

Summary of Features

- 32-bit Arm®* Cortex®-M3 core
  - Up to 40 MHz clock frequency
  - One clock per machine cycle architecture
- On-chip memory
  - Up to 256 KB flash including
  - 4 KB EEPROM (emulated in flash)
  - 512 byte 100-time programmable memory (100TP)
  - Up to 8 KB RAM
  - Boot ROM for startup firmware and flash routines
- On-chip OSC and PLL for clock generation
  - PLL loss-of-lock detection
- MOSFET driver including charge pump
- 10 general-purpose I/O Ports (GPIO)
- 5 analog inputs, 10-bit A/D Converter (ADC1)
- 2 differential Sigma Delta 14-bit ADC (ADC3/4), product-dependent, see datasheet
- 16-bit timers - GPT12, Timer2, Timer21, and Timer3
- Capture/compare unit for PWM signal generation (CCU6)
- 2 full-duplex serial interfaces (UART) with LIN support (for UART1 only)
- 2 synchronous serial channels (SSC)
- On-chip debug support via 2-wire SWD
- 1 LIN 2.2 transceiver, excluding TLE9871 products
- Bidirectional PWM interface, TLE9871 products
- 1 high-voltage monitoring input
- Single power supply from 5.5 V to 27 V
- Extended power supply voltage range from 3 V to 28 V
- Low-dropout voltage regulators (LDO)
- High-speed operational amplifier for motor current sensing via shunt
- 5 V voltage supply for external loads (e.g., Hall sensor)
- Core logic supply at 1.5 V
- Programmable window watchdog (WDT1) with independent on-chip clock source
- Power-saving modes
  - MCU slow-down mode
  - Sleep mode
  - Stop mode
  - Cyclic wake-up sleep mode
- Power-on and undervoltage/brownout reset generator

* Arm and Cortex are registered trademarks of Arm Limited, UK

- Overtemperature protection
- Short-circuit protection
- Loss of clock detection with fail-safe mode entry for low system power consumption

2.1 Abbreviations

The following acronyms and terms are used within this document. List see in **Table 3**.

Table 3 Acronyms

Acronyms	Name
AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
CCU6	Capture compare unit 6
CGU	Clock generation unit
CLKMU	Clock management unit
CMU	Cyclic management unit
CP	Charge pump for MOSFET driver
CSA	Current-sense amplifier
DMA	Direct memory access
DPP	Data post-processing
ECC	Error correction code
EEPROM	Electrically erasable programmable read only memory
EIM	Exceptional interrupt measurement
ESM	Exceptional sequence measurement
FSM	Finite state machine
GPIO	General-purpose input/output
H-Bridge	Half-bridge
ICU	Interrupt control unit
IEN	Interrupt enable
IIR	Infinite impulse response
LDM	Load instruction
LDO	Low-dropout voltage regulator
LIN	Local interconnect network
LSB	Least significant bit
LTI	Lead tip inspection
MCM	Measurement core module
MCU	Microcontroller unit
MF	Measurement functions
MPU	Memory protection unit
MRST	Master receive, slave transmit
MSB	Most significant bit
MTSR	Master transmit, slave receive
MU	Measurement unit
NMI	Non-maskable interrupt
NVIC	Nested vector interrupt controller

Table 3 **Acronyms** (cont'd)

Acronyms	Name
NVM	Non-volatile memory
OPN	Orderable part number
OSC	Oscillator
OTP	One-time programmable
PBA	Peripheral bridge
PC	Program counter
PCU	Power control unit
PD	Pull-down
PGU	Power supply generation unit
PLL	Phase-locked loop
PMU	Power management unit
POR	Power-on reset
PPB	Private Peripheral Bus
PSW	Program status word
PU	Pull-up
PWM	Pulse-width modulation
RAM	Random-access memory
RCU	Reset control unit
rfu	Reserved for future use
RMU	Reset management unit
ROM	Read-only memory
SCU-DM	System control unit – digital modules
SCU-PM	System control unit – power modules
SFR	Special function register
SOW	Short open window (for WDT)
SPI	Serial Peripheral Interface
SSC	Synchronous serial channel
STM	Store instruction
SWD	Arm® Serial Wire Debug
TCCR	Temperature compensation control register
TMS	Test mode select
TSD	Thermal shut-down
UART	Universal asynchronous receiver-transmitter
VBG	Voltage reference bandgap
VCO	Voltage-controlled oscillator
VPRE	Preregulator
WDT	Watchdog timer in SCU-DM
WDT1	Watchdog timer in SCU-PM

Table 3 **Acronyms** (cont'd)

Acronyms	Name
WMU	Wake-up management unit
100TP	100-time programmable

3 Modes of operation

The TLE987x highly integrated circuit contains analog and digital functional blocks. An embedded 32-bit microcontroller is available for system and interface control. On-chip, low-dropout regulators are provided for internal and external power supply. An internal oscillator provides a cost-effective clock that is particularly well suited for LIN communications. A LIN transceiver is available as a communication interface. Driver stages for a motor bridge or BLDC motor bridge with external MOSFET are integrated, featuring PWM capability, protection features, and a charge pump for operation at low supply voltage. A 10-bit SAR ADC and two differential 14-bit Sigma Delta ADCs[1)] are implemented for high-precision sensor measurement. An 8-bit ADC is used for diagnostic measurements.

The microcontroller unit supervision and system protection (including a reset feature) is complemented by a programmable window watchdog. A cyclic wake-up circuit, supply voltage supervision and integrated temperature sensors are available on-chip.

All relevant modules offer power saving modes in order to support automotive applications connected to terminal 30. A wake-up from power-save mode is possible via a LIN bus message, via the monitoring input, or using a programmable time period (cyclic wake-up).

The TLE987x has several operation modes mainly to support low power consumption requirements.

Reset mode

The Reset mode is a transition mode used, e.g., during power-up of the device after a power-on reset, or after wake-up from Sleep mode. In this mode, the on-chip power supplies are enabled and all other modules are initialized. Once the core supply VDDC is stable, the device enters Active mode. If the watchdog timer WDT1 fails more than four times, the device performs a fail-safe transition to Sleep mode.

Active mode

In Active mode, all modules are activated and the TLE987x is fully operational.

Stop mode

Stop mode is one of two major low-power modes. The transition to the low-power modes is performed by setting the corresponding bits in the mode control register. In Stop mode, the embedded microcontroller is still powered, allowing for shorter wake-up response times. Wake-up from this mode is possible through LIN bus activity, by using the high-voltage monitoring pin, or through the corresponding 5 V GPIOs.

Stop mode with cyclic wake-up

The Cyclic Wake-Up mode is a special operating mode of the Stop mode. The transition to the Cyclic Wake-Up mode is performed by first setting the corresponding bits in the mode control register, followed by the Stop Mode command. In addition to the cyclic wake-up behavior (wake-up after a programmable time period), asynchronous wake events via the activated sources (LIN and/or MON) are available, as in normal Stop mode.

Sleep mode

The Sleep mode is a low-power mode. The transition to the low-power mode is performed by setting the corresponding bits in the MCU mode control register or in case of failure (see below). In Sleep mode the embedded microcontroller power supply is deactivated, allowing for the lowest system power consumption. A wake-up from this mode is possible by LIN bus activity, the High Voltage Monitor Input pin, or through cyclic wake-up.

1) SDADC is not available in all products of the TLE987x family. See the datasheet for details.

Sleep mode in case of failure

Sleep mode is activated after 5 consecutive watchdog failures or in case of supply failure (5 times). In this case, MON is enabled as the wake source and cyclic wake-up is activated with 1 s of wait time.

Sleep mode with cyclic wake-up

The Cyclic Wake-Up mode is a special operating mode of the Sleep mode. The transition to Cyclic Wake-Up mode is performed by first setting the corresponding bits in the mode control register followed by the Sleep mode command. In addition to the cyclic wake-up behavior (wake-up after a programmable time period), asynchronous wake events via the activated sources (LIN and/or MON) are available, as in normal Sleep mode.

When using Sleep mode with cyclic wake-up, the voltage regulator is switched off and started again with the wake. A limited number of registers is buffered during sleep, and can be used by software, e.g., for counting sleep/wake cycles.

MCU Slow Down mode

In MCU Slow Down mode the MCU frequency is reduced to save power during operation. LIN communication is still possible. LS MOSFET can be activated.

Wake-up source prioritization

All wake-up sources have the same priority. In order to handle the asynchronous nature of the wake-up sources, the first wake-up signal will initiate the wake-up sequence. Nevertheless, all wake-up sources are latched in order to provide all wake-up events to the application software. The software can clear the wake-up source flags. This is to ensure that no wake-up event is lost.

As the default wake-up source, the MON input is activated after power-on reset only. Additionally, the device is in Cyclic Wake-Up mode with the configurable dead time setting.

The following table shows the possible power mode configurations including the Stop mode.

Table 4 Power mode configurations

Module/function	Active mode	Stop mode	Sleep mode	Comment
VDDEXT	ON/OFF	ON (no dynamic load)/OFF	OFF	–
Bridge Driver	ON/OFF	OFF	OFF	
LIN TRx	ON/OFF	Wake-up only/OFF	Wake-up only/OFF	–
VS sense	ON/OFF Brownout detection	Brownout detection	POR on VS	Brownout detection performed in PCU
GPIO 5V (wake-up)	n.a.	Disabled/static	OFF	–
GPIO 5V (active)	ON	ON	OFF	–
WDT1	ON	OFF	OFF	–
CYCLIC WAKE	n.a.	Cyclic wake-up/ cyclic sense/OFF	Cyclic wake-up/OFF	–
Measurement	ON[1)]	OFF	OFF	–
MCU	ON/slow-down/STOP	STOP[2)]	OFF	–
CLOCK GEN (MC)	ON	OFF	OFF	–
LP_CLK (18 MHz)	ON	OFF	OFF	WDT1
LP_CLK2 (100 kHz)	ON/OFF	ON/OFF	ON/OFF	For cyclic wake-up

1) May not be switched off due to safety reasons.
2) MC PLL clock disabled, MC supply reduced to $V_{DDCOUT_Stop_Red}$.

Wake-up levels and transitions

The wake-up can be triggered by rising, falling, or both signal edges for the monitor input, GPIOs, by LIN, or by cyclic wake-up.

4 Device register types

The following register types are used within this device. List see in **Table 5**.

In the table below, “set” means writing 1 to the bit and “clear” means writing 0.

Table 5 Register Types

Type	Can be modified by			Description
	Hardware	Firmware	Software	
r	yes	no	no	Read-only flag
rh	yes	no	no	Read-only flag which is modified by hardware
rhc	yes	no	yes	Read-only flag which can be cleared by a read operation
rw	no	no	yes	Bit can be read or written
rwp	yes	yes	no	Protected bit; read operations are always possible
rwp2	yes	yes	yes	Protected bit; read operations are always possible; the protection can be removed by writing the SCU password
rwh	yes	no	yes	Bit can be written by hardware and software; hardware has priority; read operations are always possible
rwh1	yes	no	yes	Bit can be set by software and is cleared by hardware; hardware has priority; read operations are always possible
rwc	yes	no	yes	Bit can be written by hardware and software; writing any value to the register clears the bit; read operations are always possible
rwd	no	no	yes	Bit can be read and written by software; writing is delayed for synchronization purposes; read operations are always possible
rwhir	yes	no	yes	Bit can be written by hardware and software; hardware has only priority to clear the bit; read operations are always possible
w	yes	yes	yes	Bit can be written by hardware or software; this bit can only be set by software; it is cleared by hardware; read operations return the default value (see individual bit).
wh	yes	no	yes	Bit can be written by hardware and software; hardware has priority; read operations return the default value (see individual bit).
wi	yes	yes	yes	Bit can be written by hardware or software; this bit can only be set by software; it is cleared by hardware; read operations return the default value (see individual bit).

5 Power management unit (PMU)

5.1 Features

- System mode control (startup, sleep, stop and active)
- Power management (cyclic wake-up)
- Control of system voltage regulators with diagnosis (overload, short, overvoltage)
- Fail-safe mode detection and operation in case of system errors (watchdog fail)
- Wake-up sources configuration and management (LIN, MON, GPIOs)
- System error logging

5.2 Introduction

The power management unit is responsible for generating all required voltage supplies for the embedded MCU (VDDC, VDDP) and the external supply (VDDEXT). The power management unit is designed to ensure fail-safe behavior of the system IC by controlling all system modes, including the corresponding transitions. Additionally, the PMU provides well-defined sequences for the system mode transitions and generates hierarchical reset priorities. The reset priorities control the reset behavior of all system functions, especially the reset behavior of the embedded MCU. All these functions are controlled by a state machine. The system master function of the PMU uses an independent logic supply and system clock. For this reason, the PMU has an "Internal logic supply and system clock" module which works independently of the MCU clock.

5.2.1 Block diagram

The following figure shows the structure of the power management unit. **Table 6** describes the submodules in more detail.

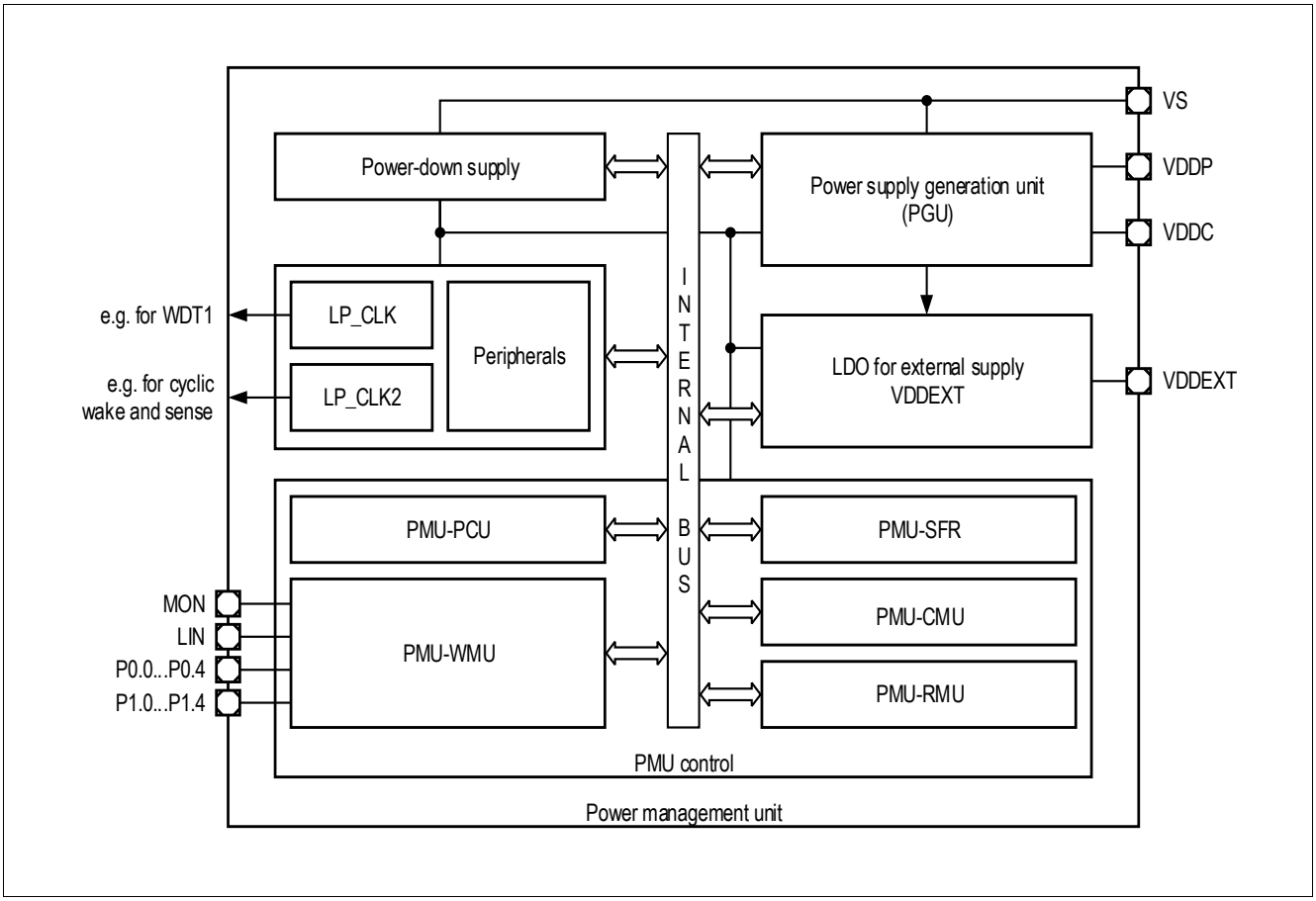


Figure 1 Power management unit block diagram

Table 6 Description of PMU submodules

Module name	Modules	Functions
Power-down supply	Independent supply voltage generation for PMU.	This supply is dedicated to the PMU to ensure an operation independently of generated power supplies (VDDP, VDDC).
LP_CLK (18 MHz)	• Clock source for all PMU submodules. • Backup clock source for the system. • Clock source for WDT1.	This ultra-low-power oscillator generates the clock for the PMU. This clock is also used as the backup clock for the system in case of PLL clock failures and as an independent clock source for WDT1.
LP_CLK2 (100 kHz)	Clock source for PMU.	This ultra-low-power oscillator generates the clock for the PMU in Stop mode and in the cyclic modes.
Peripherals	Peripheral blocks of PMU.	These blocks include the analog peripherals to ensure a stable and fail-safe PMU startup and operation (bandgap, bias).

Table 6 Description of PMU submodules (cont'd)

Module name	Modules	Functions
Power supply generation unit (PGU)	Voltage regulators for VDDP and VDDC.	This block includes the voltage regulators for the pad supply (VDDP) and the core supply (VDDC).
VDDEXT	Voltage regulator for VDDEXT to supply external modules (e.g., sensors).	This voltage regulator is a dedicated supply for external modules and can also be used for cyclic sense operations (e.g., with hall sensor).
PMU-SFR	All extended special function registers that are relevant to the PMU.	This module contains all registers needed to control and monitor the PMU.
PMU-PCU	Power control unit of the PMU.	This block is responsible for controlling all power-related actions within the PGU module. It also contains all regulator-related diagnostics such as undervoltage and overvoltage detection as well as overcurrent and short-circuit diagnostics.
PMU-WMU	Wake-up management unit of the PMU.	This block is responsible for controlling all actions related to wake-up within the PMU module.
PMU-CMU	Cyclic management unit of the PMU.	This block is responsible for controlling all actions in cyclic mode.
PMU-RMU	Reset management unit of the PMU.	This block generates resets triggered by the PMU, such as undervoltage or short-circuit reset, and passes all resets to the relevant modules and their registers.

5.2.2 PMU modes overview

The following state diagram shows the available modes of the device.

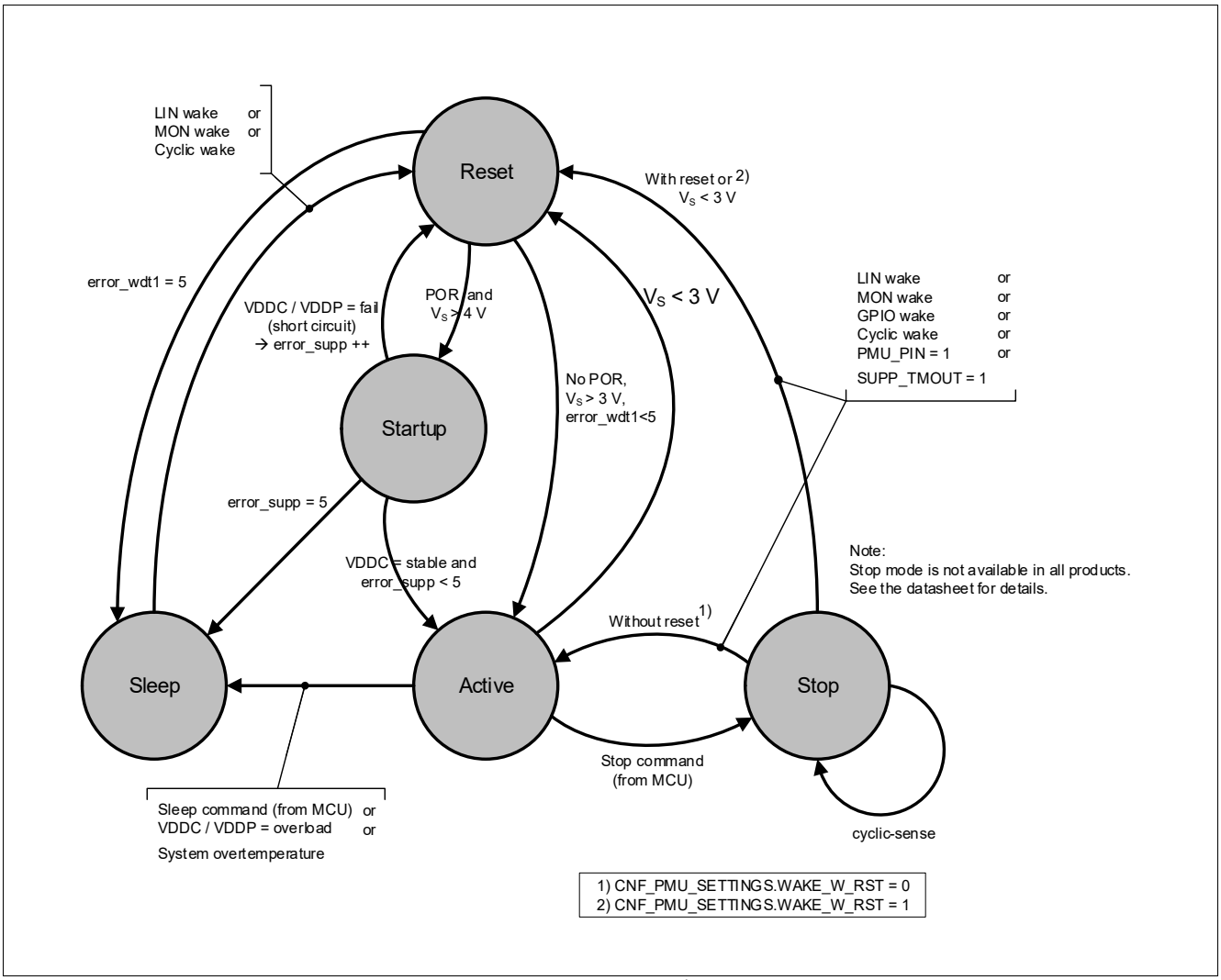


Figure 2 Power management unit system modes, LIN[1)]

1) Product dependent. See the datasheet for details.

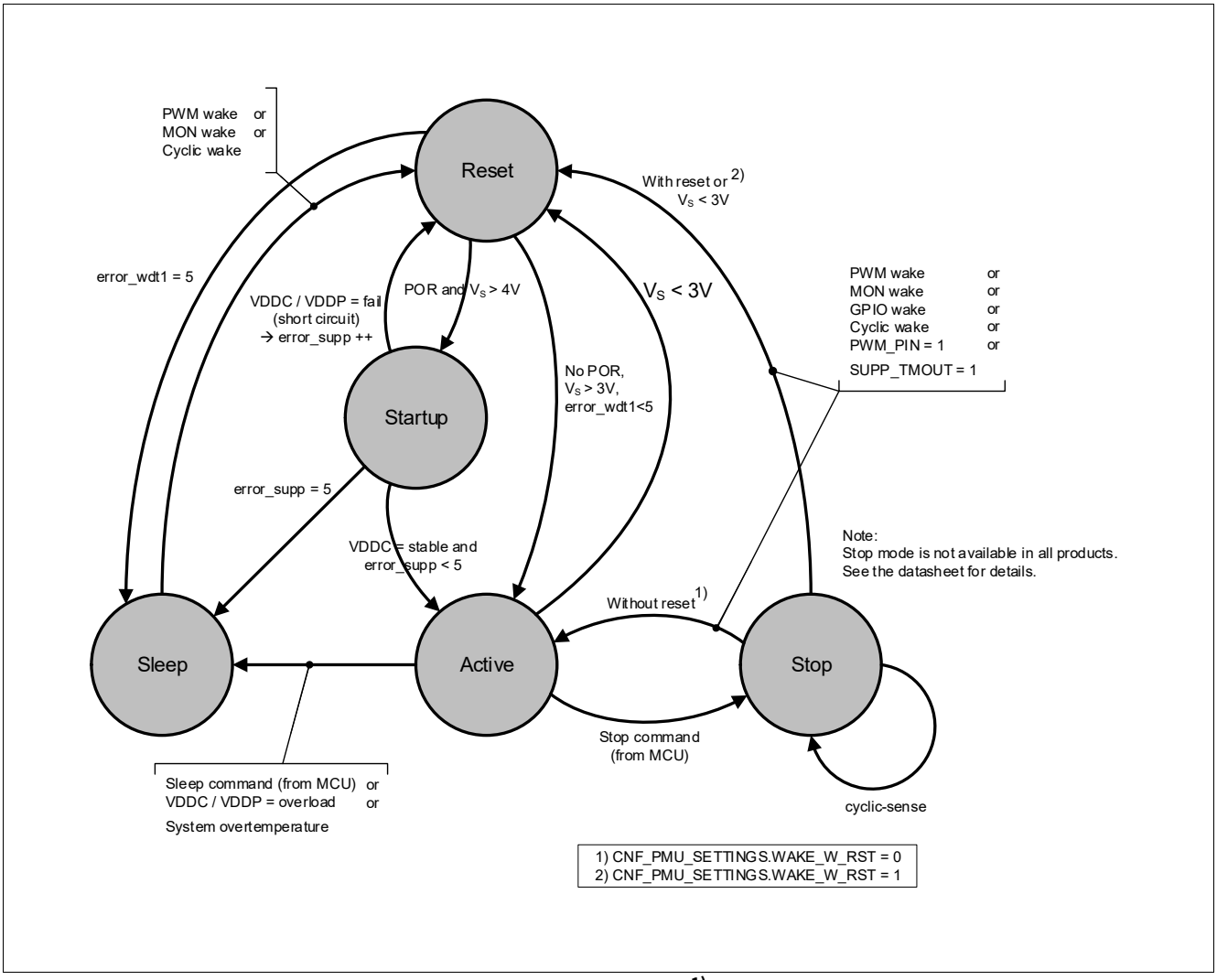


Figure 3 Power management unit system modes, PWM[1)]

Note: The error_supp and error_wdt1 values are not user-readable. Both are reset to 0 after a power-on reset (POR) or after Sleep mode is exited.

Active mode

In Active mode, the power management unit releases the reset of the embedded MCU, and the application software takes control of the system. Now the PMU is responsible for supplying and supervising the embedded system. The supervision functionality of the PMU monitors the output voltages and currents of the generated supplies and the status information of the system watchdog (WDT1).

Sleep mode

The Sleep mode is the power-saving mode with the lowest power consumption. In this mode, the PMU resets all system functions and switches off all voltage supplies which are generated in the PMU (VDDP, VDDC, VDDEXT). The only submodules of the PMU which stay active are the ones responsible for controlling the wake-up procedure of the system. **Figure 4** shows the Sleep mode entry procedure.

1) Product-dependent. See the datasheet for details.

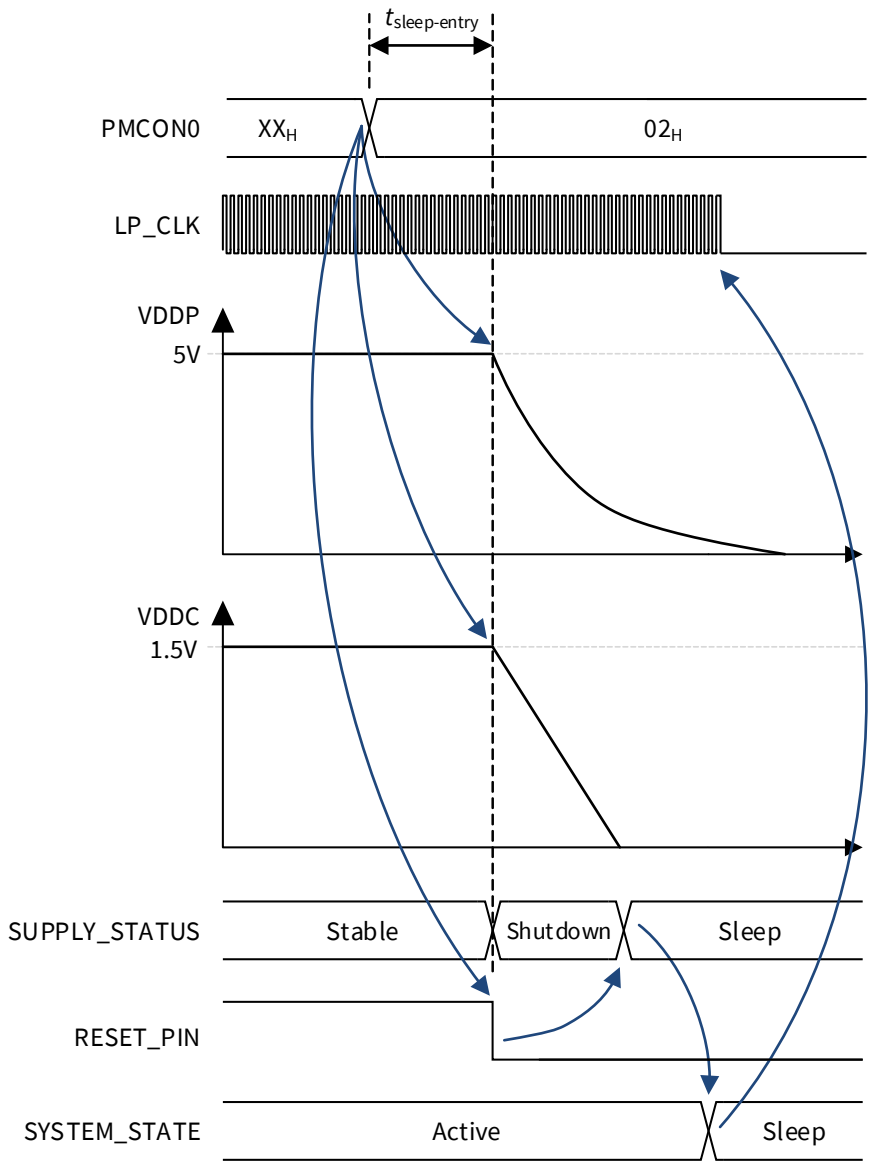


Figure 4 Sleep mode entry timing

The arrows in the diagram above indicate causal connections between the various steps and not necessarily voltage-based connections.

The Sleep mode is terminated by a LIN pulse or a corresponding (rising edge or falling edge) activity at the MON input. The PMU processes the wake-up information in an independent finite-state machine (FSM) which starts the PMU internal system clock to process the startup sequences in a synchronous way. A successful startup sequence enters the Startup mode automatically. **Figure 5** illustrates waking via LIN.

The Sleep mode can also be terminated by synchronous wake-up events. If this is desired, the PMU must be configured by setting the corresponding special function registers (SFRs). A synchronous wake-up can be configured using the Cyclic Wake mode. If these synchronous wake-up events are configured, the power consumption of the PMU increases in Sleep mode. The increased current consumption is caused by an oscillator which generates the needed time base (typically 100 kHz).

The procedure for waking up from Sleep mode via MON pins (instead of LIN) follows the same sequence as shown in **Figure 5**.

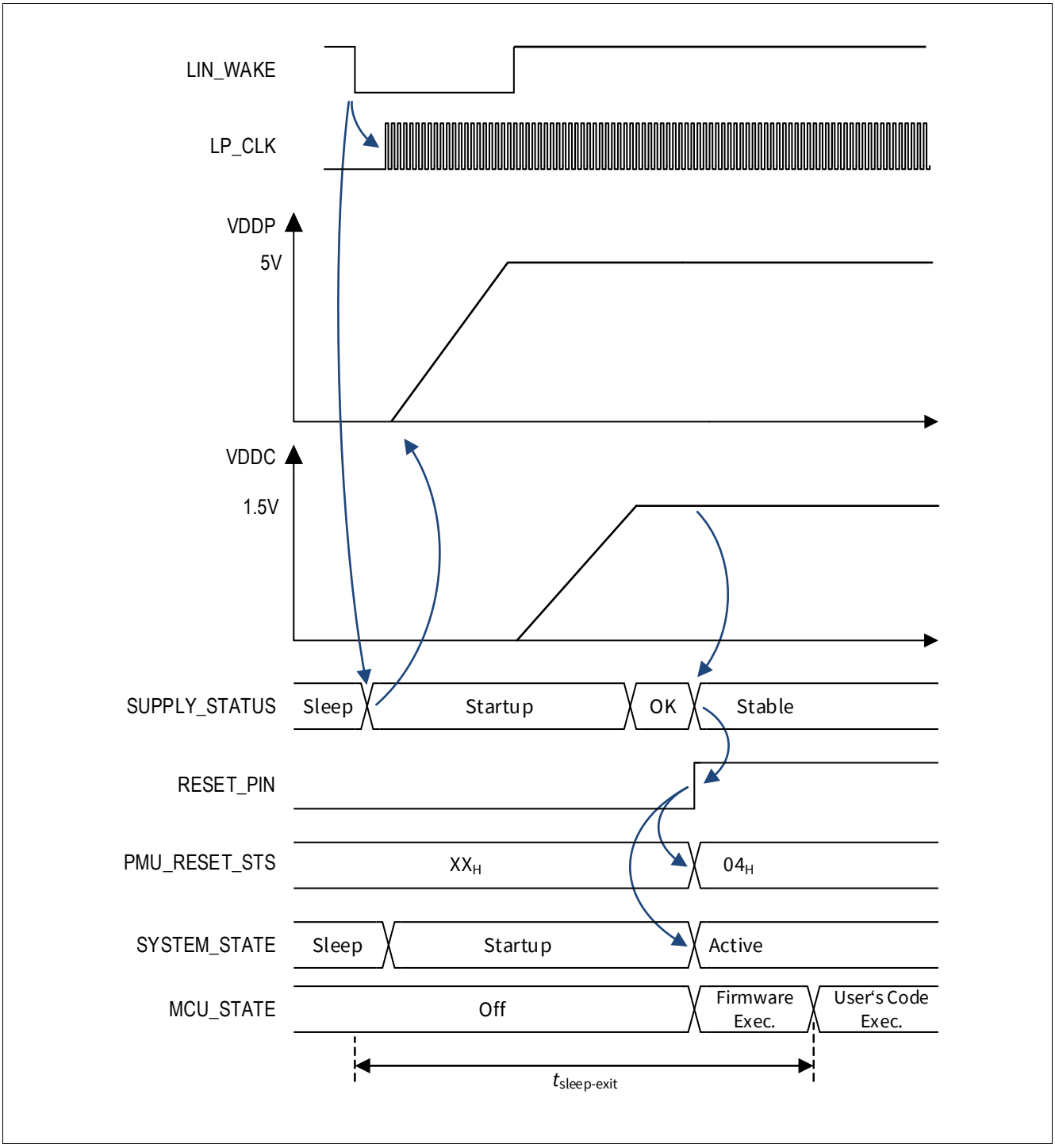


Figure 5 Sleep mode LIN wake-up timing

The arrows in the diagram above indicate causal connections between the various steps and not necessarily voltage-based connections.

Stop mode

The objective of the Stop mode is to provide a low-power mode in which the data retention feature for the embedded MCU and the special function registers (SFRs) is supported. In Stop mode, the core supply voltage VDDC switches from V_{DDCOUT} to $V_{DDCOUT_Stop_Red}$ with the objective to reduce leakage current. In Stop mode, the voltage regulation on load transients is limited. The corresponding limitation is given by the external buffer capacitor at the VDDC/VDDP pin. The figure below shows the Stop mode entry sequence.

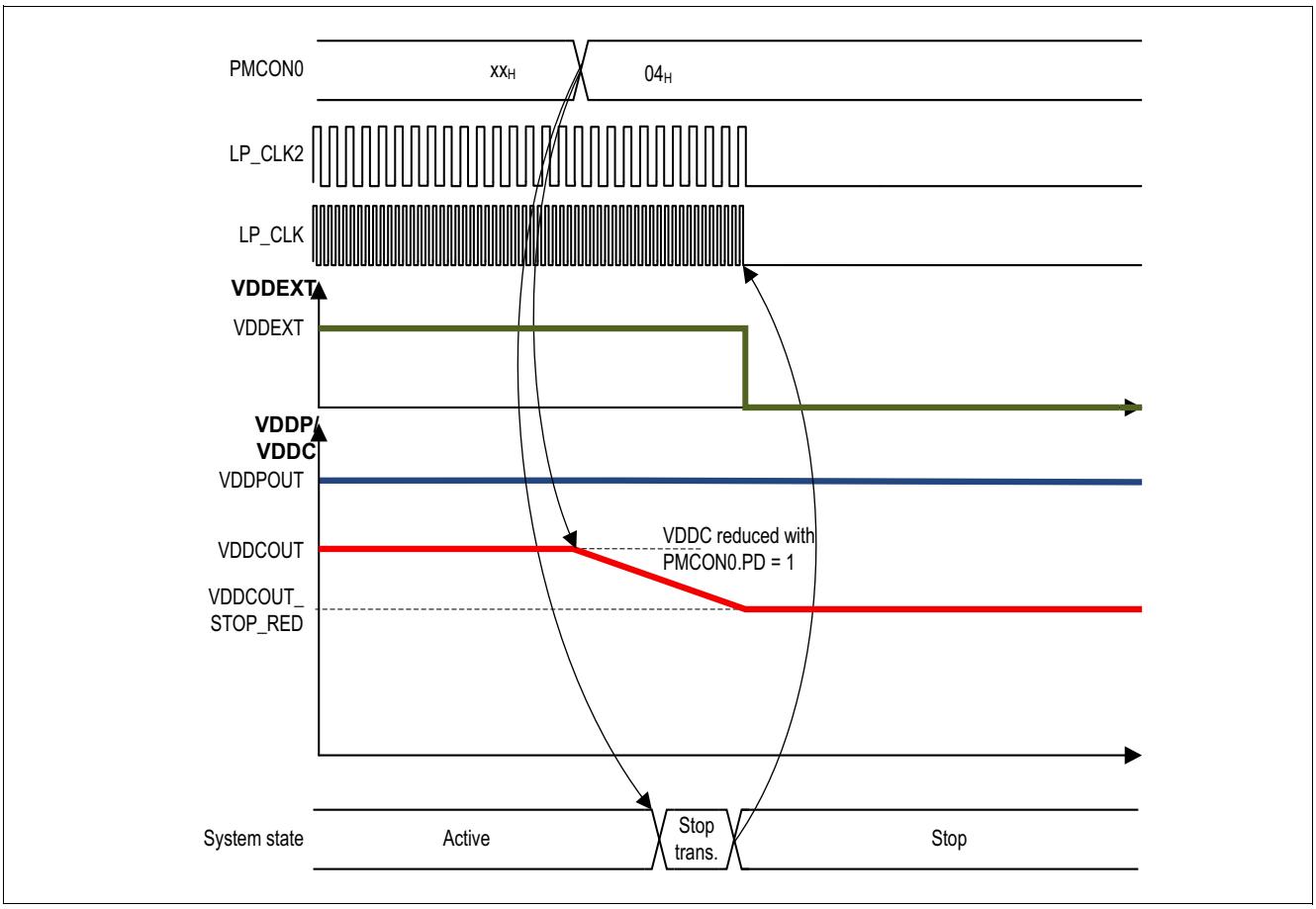


Figure 6 Stop mode entry timing

The arrows in the diagram above indicate causal connections between the various steps and not necessarily voltage-based connections.

The asynchronous wake-up works using a LIN message or an event (rising edge or falling edge) at the MON input. Stop mode is terminated by an event at one of the GPIO pins. The wake-up configuration of every MON and GPIO input is stored in the corresponding SFR. The configuration for the high-voltage input (MON) is used exiting Stop mode and exiting Sleep mode (same SFR). The Stop mode is terminated by one of the synchronous wake-up features, which include the MCU's Cyclic Sense mode and cyclic wake-up (waking up after a set interval). Both of these wake-up procedures work similarly to exiting Sleep mode. In Cyclic Sense mode, both the MON input and the GPIOs can be evaluated, and a transition causes the termination of the Stop mode. The sensing period for MON inputs and GPIOs is generated with the same time base (typically 100 kHz). The sensing period is set in the **CNF_CYC_SENSE** register. To bias the external load of the GPIOs, the supply voltage VDDEXT may switch on for the sensing time. Only during this sensing time does the PMU evaluate the corresponding GPIO. When a valid wake-up signal is received, the PMU goes to Active mode and the application software takes control over the system. If no valid wake-up information is available, then the external supply VDDEXT switches off until the configured sensing period starts again.

5.3 Power supply generation unit (PGU)

As shown in the diagram below, the power supply generation unit consists of the following modules:

Submodules of the PGU:

- **Power-down supply:** Independently generates analog supply voltage for the power control unit, the VDDP regulator, and the VDDC regulator.
- **VPRE:** Analog supply voltage preregulator to reduce the power dissipation for the following regulator stages.
- **VDDP:** 5 V digital voltage regulator used for internal modules and all GPIOs.
- **VDDC:** 1.5 V digital voltage regulator used for internal microcontroller modules and core logic.
- **PMU-PCU:** The power control unit supervises and controls the 5-V and 1.5-V regulators.

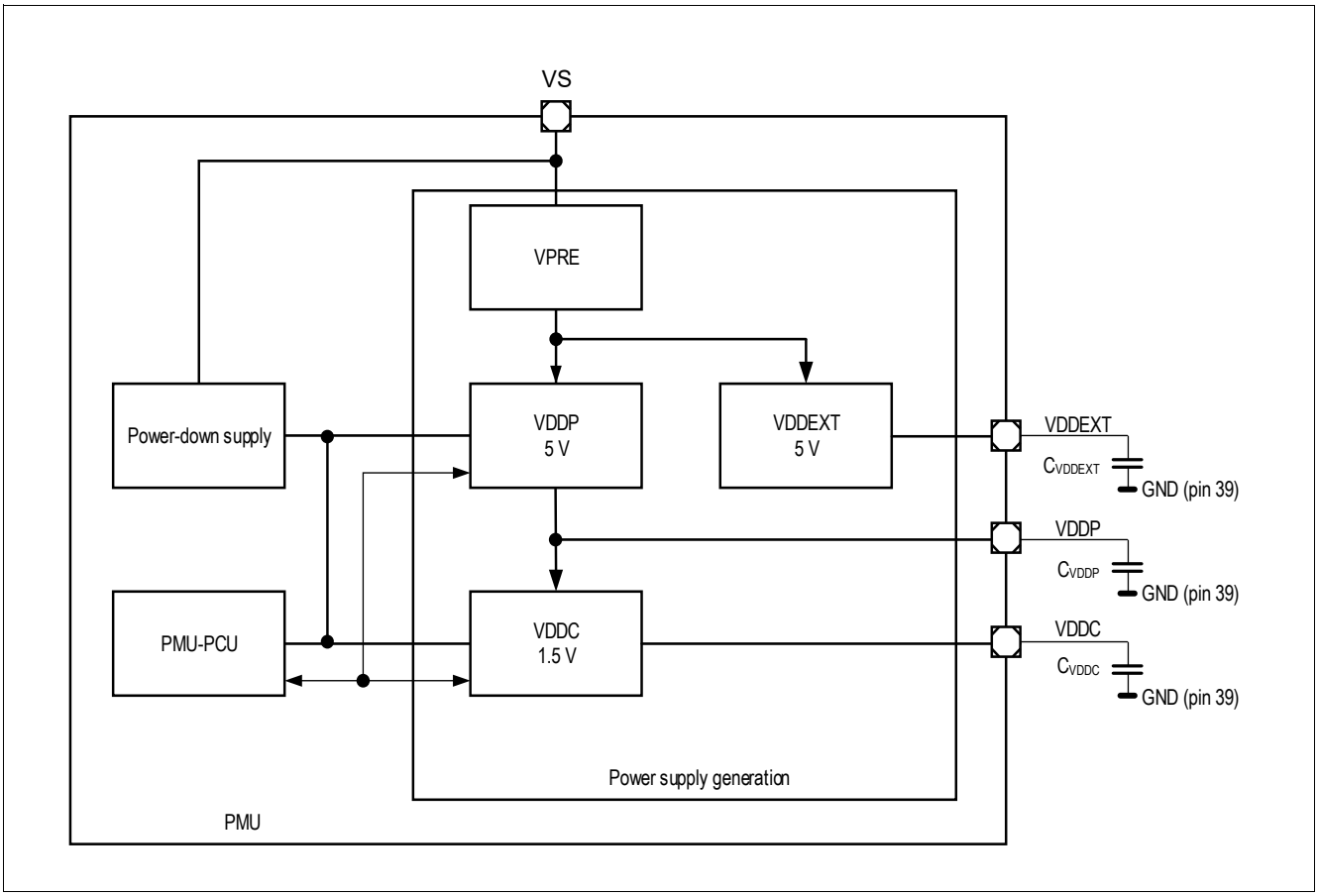


Figure 7 Power supply generation block diagram

5.3.1 Voltage regulator 5.0 V (VDDP)

This module represents the 5 V voltage regulator, which provides the pad supply for the parallel port pins and other 5 V analog functions (e.g. LIN transceiver).

Features

- 5 V low-drop voltage regulator
- Overcurrent monitoring and shutdown with MCU signaling (interrupt)
- Overvoltage monitoring with MCU signaling (interrupt)
- Undervoltage monitoring with MCU signaling (interrupt)
- Undervoltage monitoring with reset (undervoltage reset, V_{DDPUV})
- Preregulator for the VDDC regulator
- GPIO supply
- Pull-down current source at the output for Sleep mode only (typ. 5 mA)

The output capacitor C_{VDDP} is mandatory to ensure proper regulator functionality.

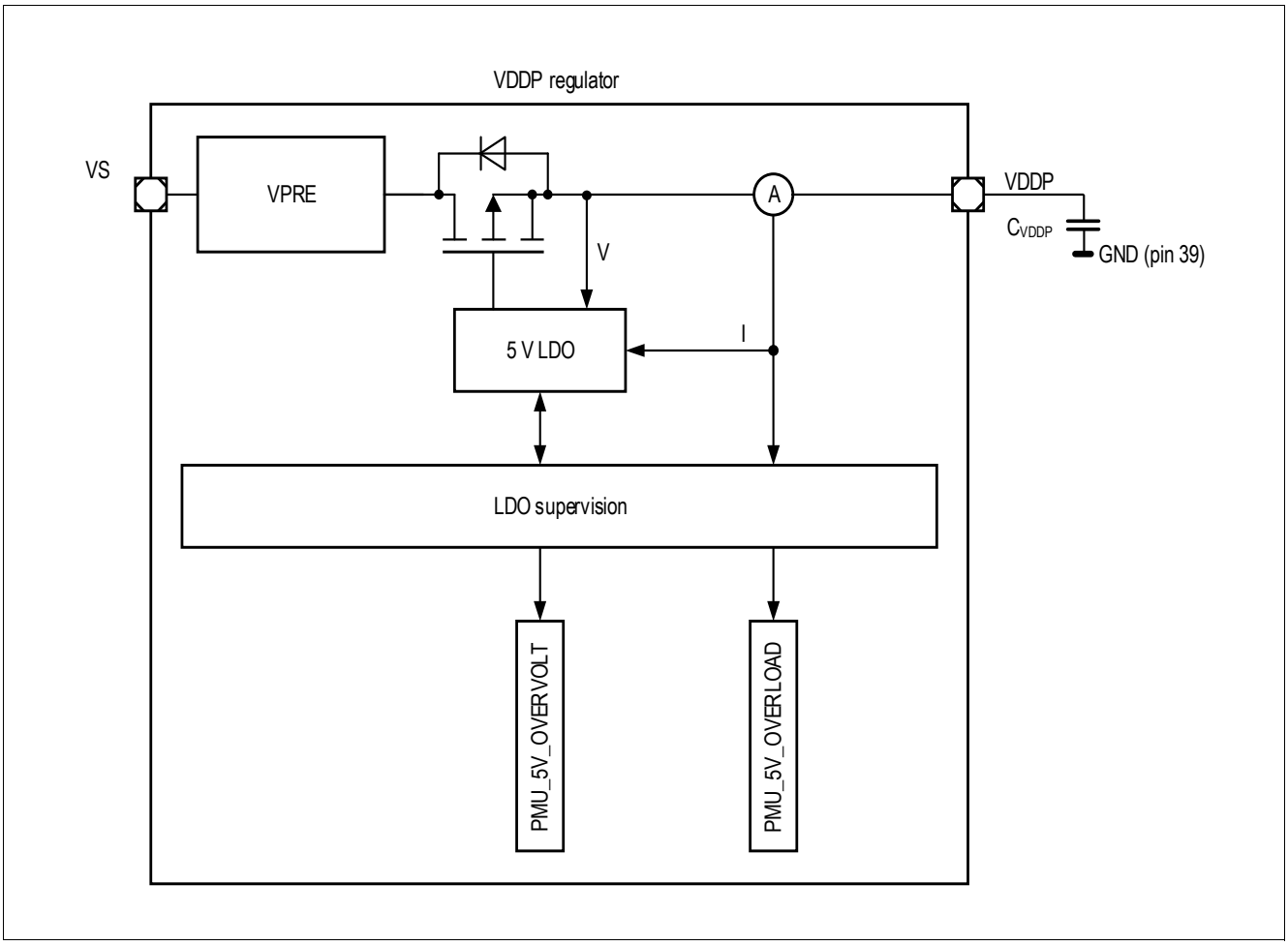


Figure 8 Module block diagram of the VDDP voltage regulator

5.3.2 Voltage regulator 1.5 V (VDDC)

This module represents the 1.5 V voltage regulator, which provides the supply for the microcontroller core, the digital peripherals, and other internal analog 1.5 V functions (e.g., ADC2) of the chip. To further reduce the current consumption of the MCU during Stop mode the output voltage can be lowered to $V_{DDCOUT_Stop_Red}$.

Features

- 1.5 V low-drop voltage regulator
- Overcurrent monitoring and shutdown with MCU signaling (interrupt)
- Overvoltage monitoring with MCU signaling (interrupt)
- Undervoltage monitoring with MCU signaling (interrupt)
- Undervoltage monitoring with reset
- Pull-down current source at the output for Sleep mode only (typ. 100 µA)

The output capacitor C_{VDDC} is mandatory to ensure a proper regulator functionality.

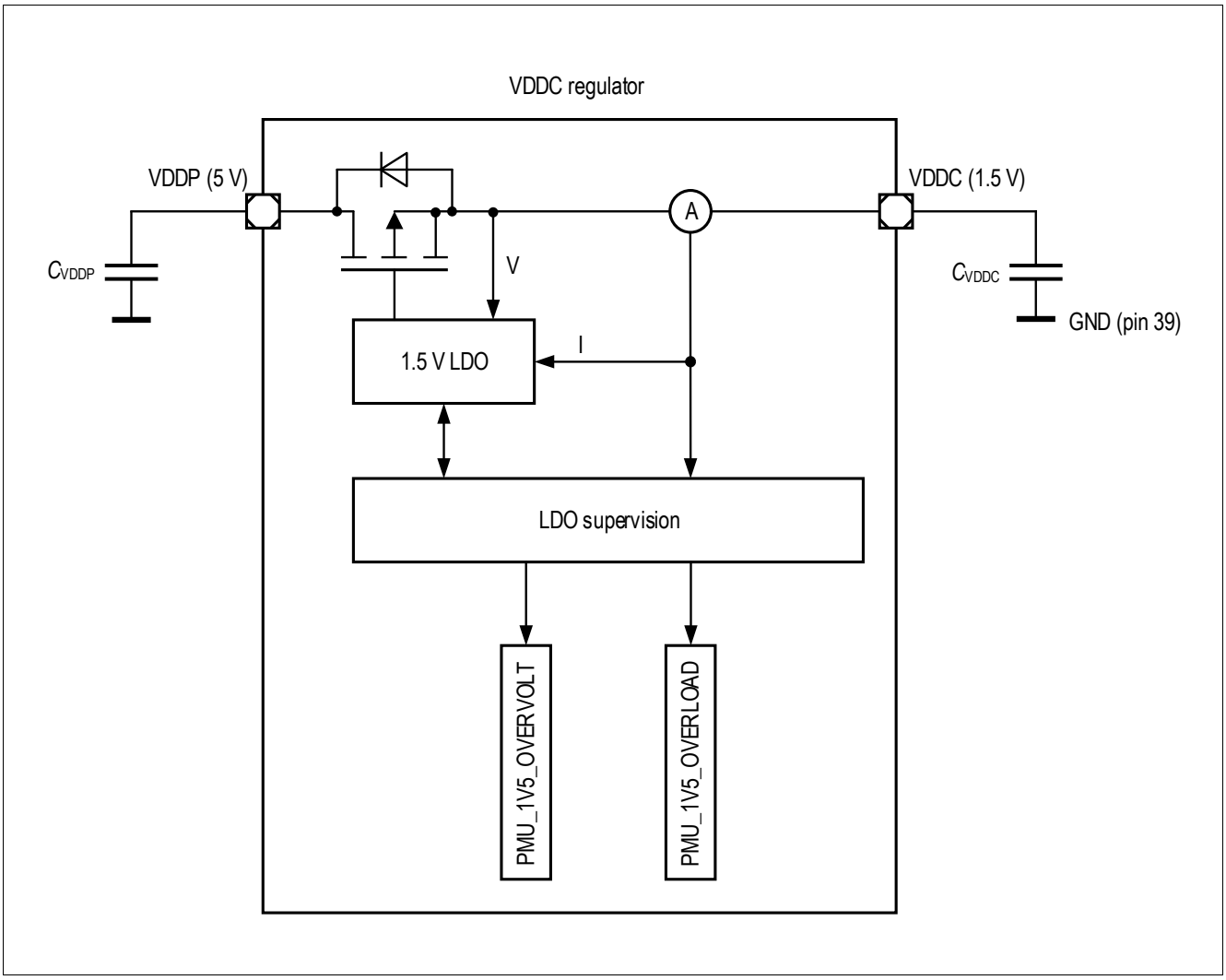


Figure 9 Module block diagram of the VDDC voltage regulator

5.3.3 External voltage regulator 5.0 V (VDDEXT)

This module represents the 5 V voltage regulator, which serves as a supply for external circuits. It can be used, e.g., to supply an external sensor, LEDs, or potentiometers. VDDEXT can be used as reference for SDADC (ADC3/4).[1)]

Features

- Switchable +5 V, low-drop voltage regulator
- Switch-on overcurrent blanking time in order to drive small capacitive loads
- Overcurrent monitoring and shutdown with MCU signaling (interrupt)
- Overvoltage monitoring with MCU signaling (interrupt)
- Undervoltage monitoring with MCU signaling (interrupt)
- Pull-down current source at the output for Sleep mode only (typ. 100 µA)
- Cyclic sense option together with GPIOs

The output capacitor C_{VDDEXT} is mandatory to ensure a proper regulator functionality.

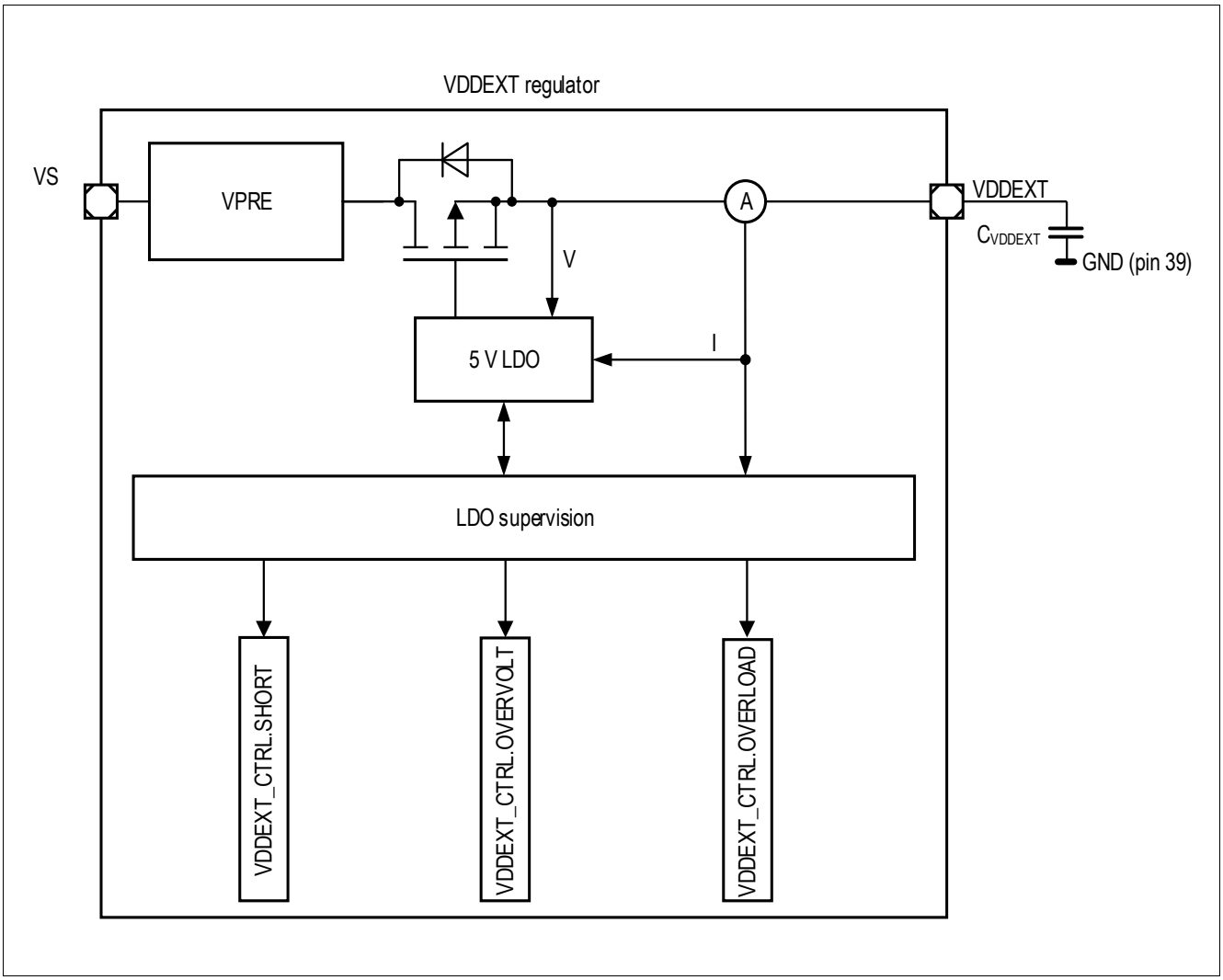


Figure 10 Module block diagram of the external voltage regulator

1) SDADC support is not available in all products of the TLE987x family. See the datasheet for details.

VDDEXT provides three diagnostic features:

- VDDEXT overvoltage detection
- VDDEXT overload detection
- VDDEXT short-circuit detection

VDDEXT overvoltage detection recognizes a reverse supply of the VDDEXT above $V_{VDDEXTOV}$ (5.4 V). Refer to datasheet for the exact value. The **VDDEXT_CTRL**.OVERVOLT flag indicates the presence of a overvoltage condition on VDDEXT.

VDDEXT overload detection is triggered if the current drawn out of VDDEXT exceeds $I_{VDDEXTOC}$ (50 mA). Refer to the datasheet for the exact value. The **VDDEXT_CTRL**.OVERLOAD flag is set if the overload condition is present.

VDDEXT short-circuit detection is triggered by the undervoltage threshold. If the VDDEXT output voltage drops below the undervoltage threshold of $V_{VDDEXTUV}$ (2.8 V; refer to the datasheet for the exact value) because the voltage being drawn out of VDDEXT is too high, the **VDDEXT_CTRL**.SHORT flag is set and the VDDEXT is switched off. In order to turn it on again, VDDEXT has to be enabled by writing 1 to **VDDEXT_CTRL**.ENABLE.

Note: If the VS supply is below V_{S_AM}, VDDEXT operates in low-drop-out mode and the output voltage of VDDEXT follows the VS voltage with a drop of V_S - V_{DDEXT} (P_2.3.4, P2.3.14; refer to the datasheet for the exact value). If the undervoltage threshold of VDDEXT is reached, a short-circuit is signalled, even though it is not caused by a short but by an undervoltage. However, the physical evaluation behind both is the same.

All three diagnostic signals cause an NMI if **VDDEXT_CTRL**.FAIL_EN is enabled. The shared **NMICON**.NMISUP flag enables the corresponding NMI to signal the VDDEXT issue detected.

5.3.4 Low-VS operation

The integrated VDDP regulator enters dropout operation as the VS pin voltage is dropping below the minimum supply voltage. As a consequence the regulator will enter dropout and can no longer maintain its output voltage within the regulation limits.

The MCU subsystem remains fully functional down to the minimum extended supply voltage range.
Care should be taken while operating following peripherals under low-supply conditions:

- Bridge driver
- Charge pump
- GPIOs
- Transceiver interface
- VDDEXT regulator

The following figure illustrates the operation under low-supply conditions:

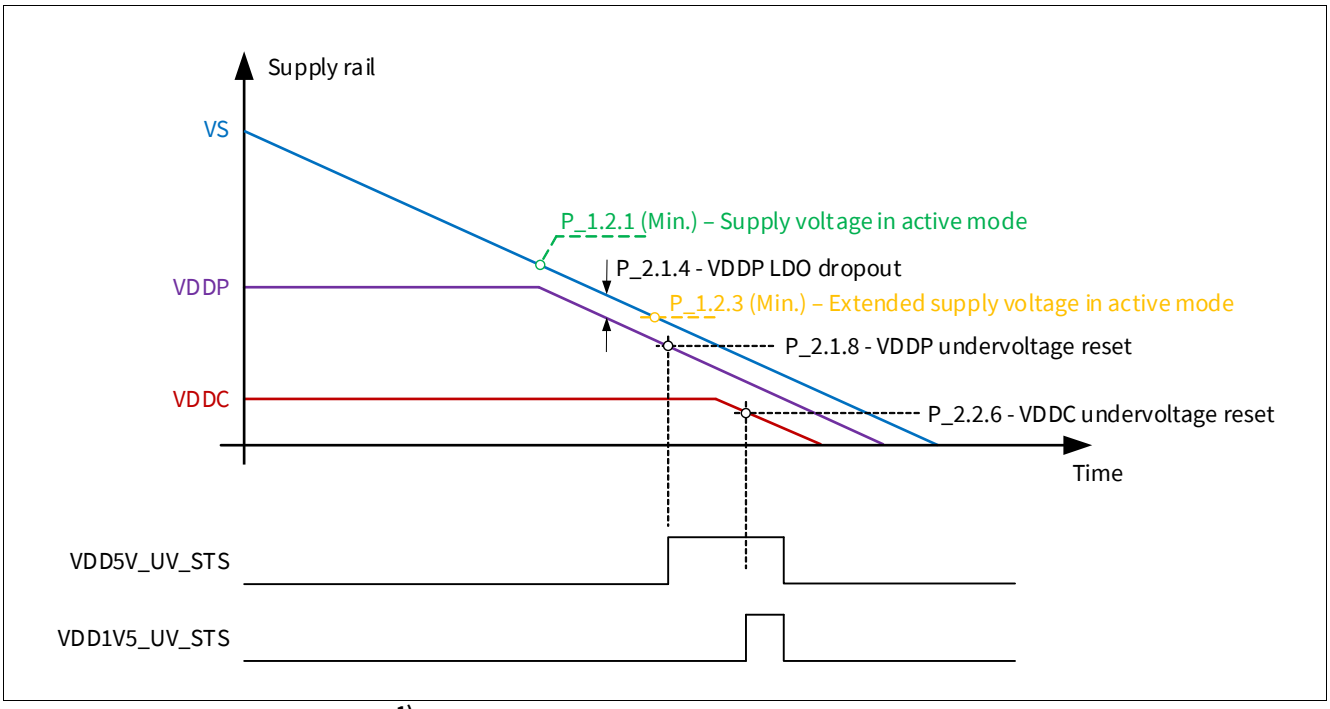


Figure 11 Low-VS operation[1)]

1) P_1.2.1, P_2.1.4, P_1.2.3, P_2.1.8, P_2.2.6: See the datasheet for details.

5.3.5 PMU register overview

Table 7 Register Address Space for PMU Registers

Module	Base Address	End Address	Note
PMU	50004000_H	$50004FFF_H$	Power management unit registers

The registers are addressed wordwise.

5.3.6 Register definition

Table 8 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Power Supply Generation Register			
PMU_SUPPLY_STS	Voltage Reg Status Register	004_H	$00xx\ 00xx_B$
VDDEXT Control Register			
VDDEXT_CTRL	VDDEXT Control Register	008_H	$xxxx\ 0000_B$

The registers are addressed wordwise.

5.3.6.1 Power Supply Generation Register

The power supply generation register contains status information for the VDDP and VDDC voltage regulators and settings for triggering interrupts based on the status.

Voltage Reg Status Register

The PMU_SUPPLY_STS register indicates the overvoltage and overload conditions of VDDP and VDDC. To use this information as interrupt sources, it must be selected explicitly in this register. The register is partly reset by RESET_TYPE_0 and partly reset by RESET_TYPE_3.

PMU_SUPPLY_STS
Voltage Reg Status Register **(004_H)** **Reset Value: $00xx\ 00xx_B$**

7	6	5	4	3	2	1	0
RES	**PMU_5V_FAIL_EN**	**PMU_5V_OVERLOAD**	**PMU_5V_OVERVOLT**	**RES**	**PMU_1V5_FAIL_EN**	**PMU_1V5_OVERLOAD**	**PMU_1V5_OVERVOLT**
r	rw	r	r	r	rw	r	r

Field	Bits	Type	Description
RES	7	r	**Reserved** Always read as 0
PMU_5V_FAIL_EN	6	rw	**Enabling of VDDP status information as interrupt source, RESET_TYPE_3** 0_B No interrupts are generated based on the VDDP status 1_B Interrupts are generated
PMU_5V_OVERLOAD	5	r	**Overload at VDDP regulator, RESET_TYPE_0** *Note: If this flag is set and an additional filter time of typically 290 µs passes, the system will be put into Sleep mode* 0_B No overload detected 1_B Overload detected
PMU_5V_OVERVOLT	4	r	**Overvoltage at VDDP regulator, RESET_TYPE_0** 0_B No overvoltage detected 1_B Overvoltage detected
RES	3	r	**Reserved** Always read as 0
PMU_1V5_FAIL_EN	2	rw	**Enabling of VDDC status information as interrupt source, RESET_TYPE_3** 0_B No interrupts are generated 1_B Interrupts are generated

Field	Bits	Type	Description
PMU_1V5_OVERLOAD	1	r	**Overload at VDDC regulator, RESET_TYPE_0** *Note: If this flag is set and an additional filter time of typically 290 µs passes, the system will be put into Sleep mode* 0_B No overload detected 1_B Overload detected
PMU_1V5_OVERVOLT	0	r	**Overvoltage at VDDC regulator, RESET_TYPE_0** 0_B No overvoltage detected 1_B Overvoltage detected

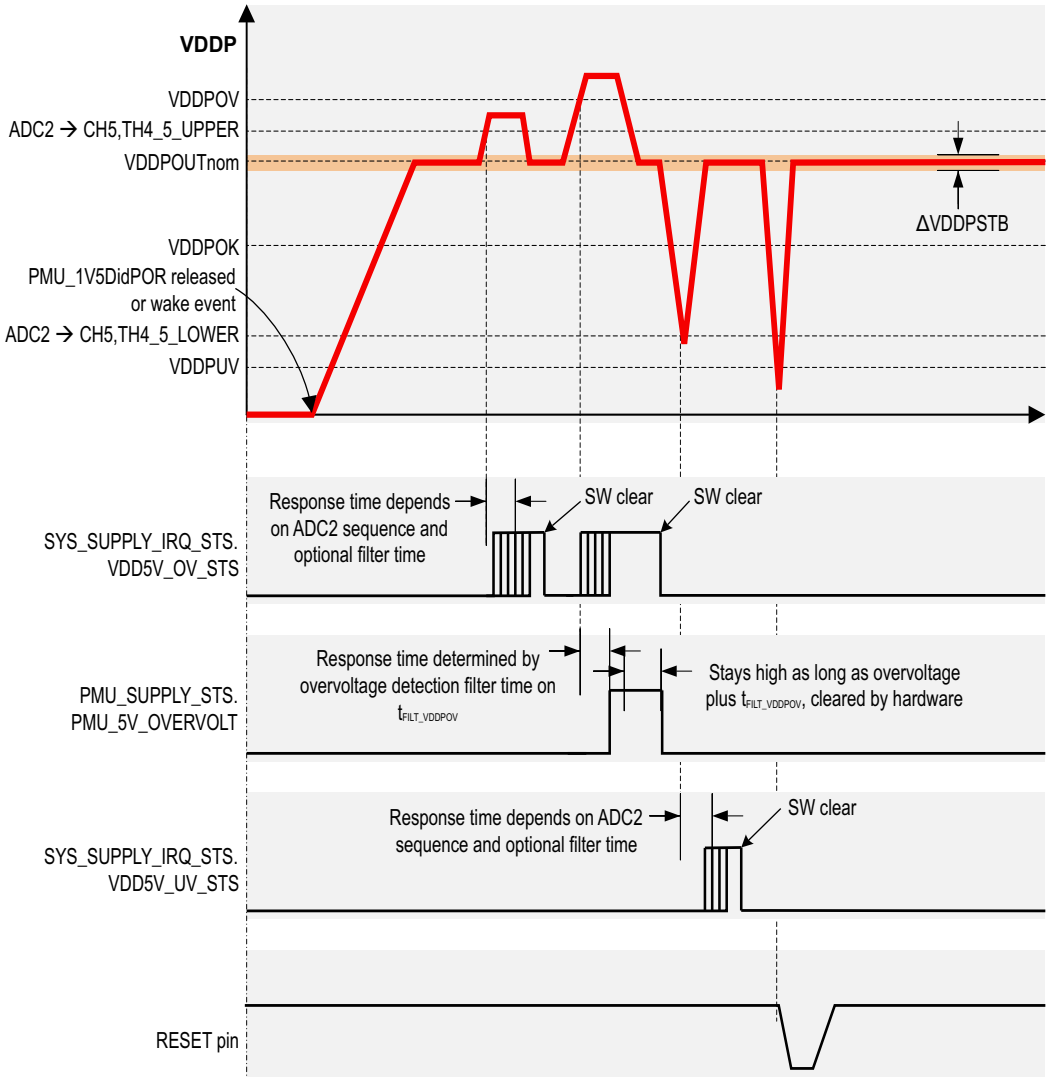


Figure 12 VDDP

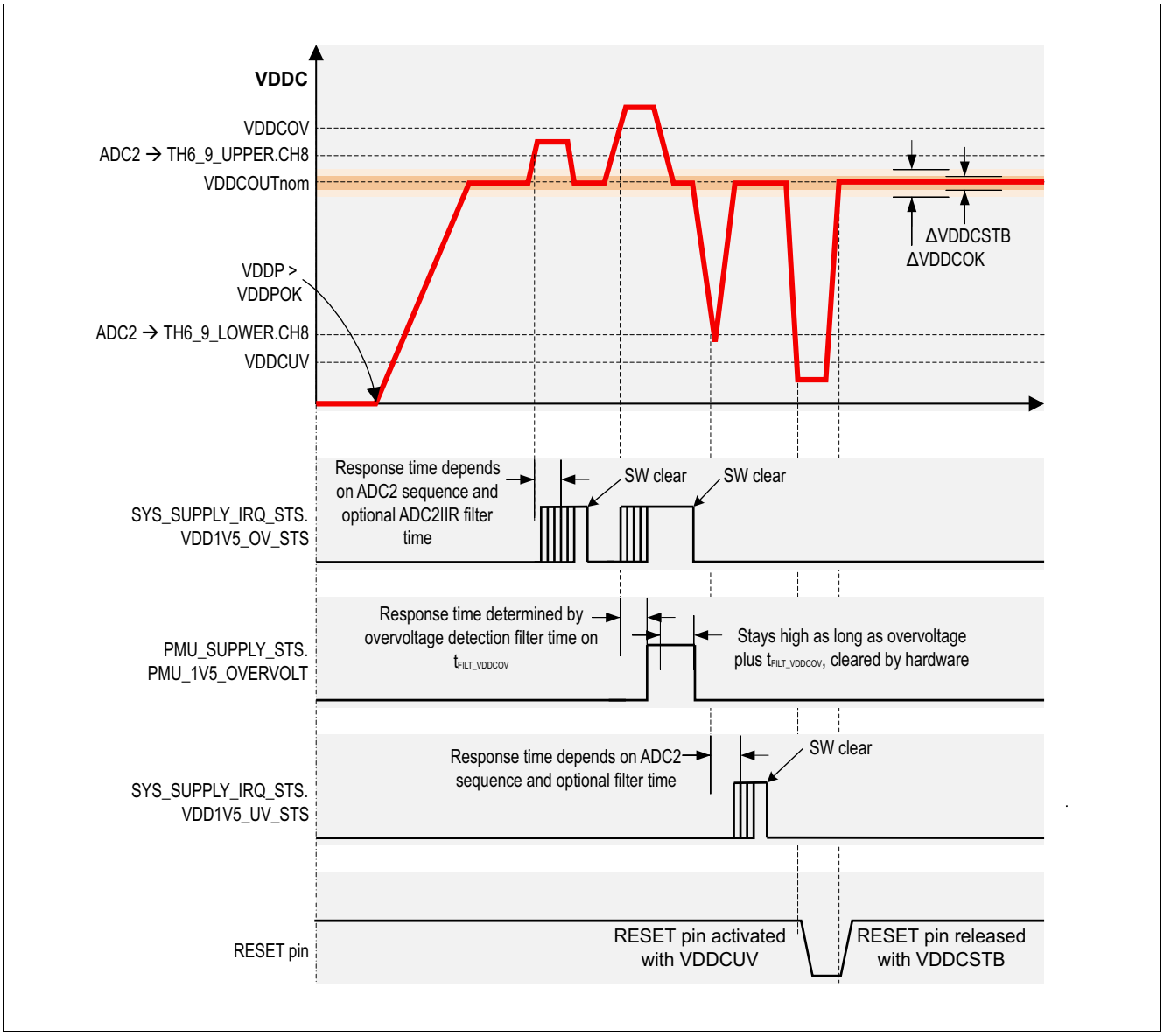


Figure 13 VDDC

5.3.6.2 VDDEXT Control Register

VDDEXT can be fully controlled by this register, including all diagnostic functions. Two over- and undervoltage ranges are available which serve as a supply prewarning. Both cases are indicated in the bits STABLE and OK.

VDDEXT Control

The register is partly reset by RESET_TYPE_3 and partly reset by RESET_TYPE_0.

The SHORT status flag triggers a shutdown of VDDEXT. To re-enable VDDEXT, the SHORT flag needs to be cleared.

VDDEXT_CTRL
VDDEXT Control **(008_H)** **Reset Value: xxxx 0000_B**

7	6	5	4	3	2	1	0
STABLE	**OK**	**OVERLOAD**	**OVERVOLT**	**SHORT**	**FAIL_EN**	**CYC_EN**	**ENABLE**
r	r	r	r	rwh	rw	rw	rw

Field	Bits	Type	Description
STABLE	7	r	**VDDEXT Supply works inside its specified range 1, RESET_TYPE_0** 1_B VDDEXT voltage inside of specified range 0_B VDDEXT voltage outside of specified range
OK	6	r	**VDDEXT Supply works inside its specified range 2 (see datasheet, $V_{VDDEXTOK}$), RESET_TYPE_0** 1_B VDDEXT in low-drop mode 0_B VDDEXT not in low-drop mode
OVERLOAD	5	r	**VDDEXT Supply Overload, RESET_TYPE_0** *Note: This flag is automatically cleared if the error condition is removed.* 0_B VDDEXT not in overload condition 1_B VDDEXT in overload condition
OVERVOLT	4	r	**VDDEXT Supply Overvoltage, RESET_TYPE_0** *Note: This flag is automatically cleared if the error condition is removed.* 0_B VDDEXT not in overvoltage condition 1_B VDDEXT in overvoltage condition

Field	Bits	Type	Description
SHORT	3	rwh	**VDDEXT Supply Shorted Output, RESET_TYPE_3** *Note: If this flag is set, the VDDEXT regulator is automatically switched off. To re-enable it, the software must clear this flag. The SHORT flag has RESET_TYPE_0.* 0_B VDDEXT no short circuit detected 1_B VDDEXT short circuit detected
FAIL_EN	2	rw	**Enabling of VDDEXT Supply status information as interrupt source, RESET_TYPE_3** 0_B VDDEXT fail interrupts are disabled 1_B VDDEXT fail Interrupts are enabled
CYC_EN	1	rw	**VDDEXT Supply for Cyclic Sense Enable, RESET_TYPE_3** *Note: To use VDDEXT supply for cyclic sense, both the CYC_EN and ENABLE bits must be set.* 0_B VDDEXT for cyclic sense disabled 1_B VDDEXT for cyclic sense enabled
ENABLE	0	rw	**VDDEXT Supply Enable, RESET_TYPE_3** 0_B VDDEXT supply disabled 1_B VDDEXT supply enabled

5.4 Power control unit

The power control unit controls the system's power supply generation unit (PGU). It offers important fail-safe features, which are described in **Chapter 5.4.1**.

5.4.1 Power control unit – fail-safe scenarios

The PMU-PCU handles several different failure scenarios, listed below and described in the following chapters:

- Fail-safe mode (Sleep mode) in case of power failure.
- Fail-safe mode (Sleep mode) in case of watchdog service failure.
- Fail-safe mode (Sleep mode) in case of overcurrent on voltage regulators VDDP or VDDC.
- Two-level monitoring (prewarning and reset) of voltage regulator output voltages (VDDP, VDDC, VDDEXT).
- Wake-up from Stop mode with cyclic sense in case of VDDEXT regulator failures.
- Wake-up from Stop mode in case of hardware reset on RESET pin.

5.4.1.1 Power supervision function of the PMU-PCU

The power supervision feature of the PMU-PCU mainly monitors the voltage regulators VDDP and VDDC. When a voltage regulator malfunctions, the PMU-PCU restarts both voltage regulators (VDDP and VDDC). Each time this happens, the error counter "error_supp" is incremented. If the counter reaches 5, the PMU-PCU supervision function puts the device into Sleep mode. In this case, the device can still be woken up by LIN and MON input.

After a wake-up, if the PMU can be successfully restarted and code execution is possible, you can determine which failure occurred by checking the corresponding **SYS_FAIL_STS** register. In this case bit SUPP_TMOUT is set.

If there is a short circuit at the VDDC or VDDP voltage regulator during startup, the reset of the embedded MCU is triggered and the system goes into Startup mode. The error counter "error_supp" is incremented. After this, the PMU-PCU itself tries to go into Active mode again using the power-on sequence. If the short circuit still exists, the procedure is repeated. This procedure will be executed, as mentioned above, no more than five times. After reaching the value 5, the PMU-PCU sends the system into Sleep mode.

If the startup after wake-up from Sleep mode is successful, you can determine the failure by reading the SUPP_SHORT flag in the **SYS_FAIL_STS** register.

5.4.1.2 Watchdog (WDT1) fail safe

The PMU-PCU monitors the failure information of the system watchdog (WDT1). If the watchdog is not serviced or serviced in a wrong way (in the following called "watchdog not serviced") the MCU is reset and the "error_wdt" error-counter is incremented. The PMU itself stays in the Active mode, and after the reset the application software takes over control of the system. If the software does not service the system watchdog, the described procedure starts again. After the watchdog has not been serviced five times during one Active mode period, the PMU sends the embedded system to Sleep mode. The PMU detects the transition to Sleep mode as a safety fallback, and the Sleep mode can be terminated in two ways: Either by a LIN wake or a rising or falling edge at a MON pin, or by a cyclic wake which is triggered after a sleep time of 1 s. The error counter is reset when the system is sent to Sleep mode or Stop mode by a software command.

If the system can be successfully restarted, the cause of failure can be again checked by reading the **SYS_FAIL_STS** register. The WDT1_SEQ_FAIL bit indicates the described failure.

5.4.1.3 Main regulator fail safe

If one of the voltage regulators needs to deliver too much current, the supply voltage can become unstable. In this case, the VDDC and VDDP overcurrent detection ensures that the system enters Sleep mode.

When the overcurrent condition is gone, a wake-up can be triggered. The system then starts up and works properly. Afterwards, the failure flags PMU_1V5_OVL and PMU_5V_OVL can be checked.

5.4.1.4 VDDEXT failure

If VDDEXT is used in combination with the GPIOs as a supply, for example, for the switches, several error cases are possible: Overvoltage, overload, and short circuit to GND. These error cases may lead to the generation of false wake-up events or to missed wake-up events. To avoid these scenarios, errors on the VDDEXT voltage regulator would automatically wake the system up from Stop mode. The errors are signalled in the **WAKE_STS_FAIL** register.

5.4.1.5 Wake-up from Stop mode with reset fail-safe

One fail-safe measure to wake up the embedded system from Stop mode can be executed by triggering a hardware reset. If there is a reset request on the reset pin, the PMU transitions to Active mode. Simultaneously, the embedded system is reset by forcing the bidirectional reset pin. The reset pin goes high again when the PMU releases the MCU reset. This event is indicated in the reset status register as a hard reset together with a wake-up reset. In case of a fail condition at one of the voltage regulators, the PMU also transitions to Active mode. After this transition, the PMU starts the supply fail-safe procedure which is described in the **Active mode** section. The described sequence is illustrated in **Figure 14**.

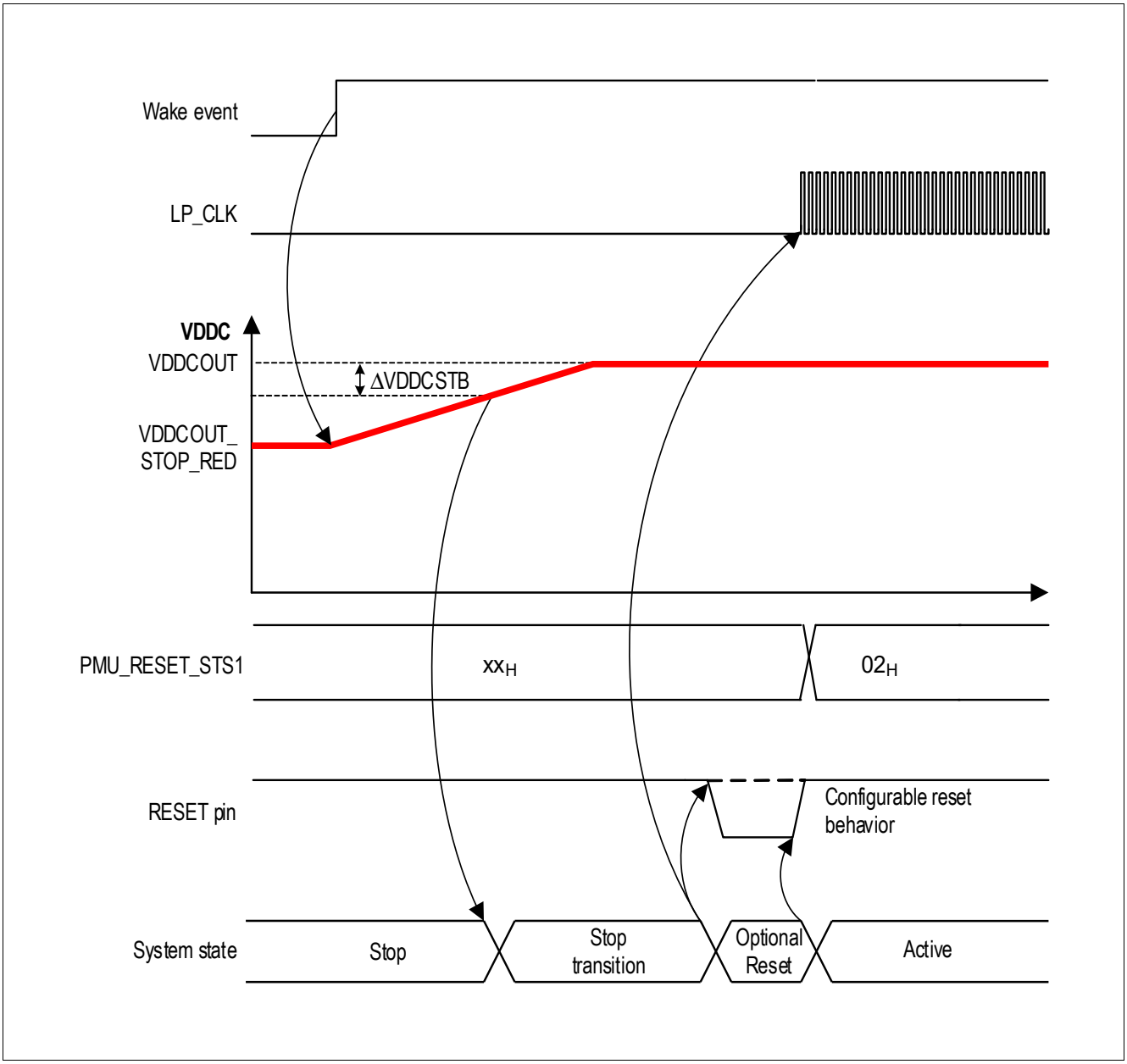


Figure 14 Stop mode exit timing

The arrows in the diagram above indicate causal connections between the various steps and not necessarily voltage-based connections.

5.4.2 Register Definition

Table 9 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
PMU System Fail Register			
SYS_FAIL_STS	System Fail Status Register	070_H	$0000\ 0000_H$
PMU Wake Fail Register			
WAKE_STS_FAIL	Wake Status Fail Register	080_H	$0000\ 0000_B$

The registers are addressed wordwise.

5.4.2.1 PMU System Fail Register

This register contains status information for the PMU peripherals.

System Fail Status Register

The register is reset by RESET_TYPE_0.

*Note: The SYS_FAIL_STS register is also cleared when **PMU_RESET_STS1**.SYS_FAIL is cleared.*

SYS_FAIL_STS
System Fail Status Register **(070_H)** **Reset Value: $0000\ 0000_H$**

7	6	5	4	3	2	1	0
RES	**WDT1_SEQ_FAIL**	**SYS_OT**	**RES**	**PMU_5V_OVL**	**PMU_1V5_OVL**	**SUPP_TMOUT**	**SUPP_SHORT**
r	rwh	rwh	r	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
RES	7	r	**Reserved** Always read as 0
WDT1_SEQ_FAIL	6	rwh	**External Watchdog (WDT1) Sequential Fail** Indicates that watchdog has not been serviced five times 0_B **No Fail**, System working properly 1_B **Sequential Watchdog Fail**, five consecutive watchdog fails
SYS_OT	5	rwh	**System Overtemperature Indication Flag** Indicates system overtemperature condition 0_B **No Overtemperature**, System temperature OK 1_B **Overtemperature**, System overtemperature
RES	4	r	**Reserved** Always read as 0
PMU_5V_OVL	3	rwh	**VDDP Overload Flag** Indicates overload condition at VDDP 0_B **No Overload**, VDDP load OK 1_B **Overload**, VDDP overload

Field	Bits	Type	Description
PMU_1V5_OVL	2	rwh	**VDDC Overload Flag** Indicates overload condition at VDDC 0_B **No Overload**, VDDC load OK 1_B **Overload**, Hall VDDC overload
SUPP_TMOUT	1	rwh	**Supply Time Out** Indicates the status of the main supply (VDDP and VDDC) after a certain time of power-on reset 0_B **Main Supply ok**, VDDP or VDDC are within the expected range 1_B **Main Supply fail**, VDDP or VDDC do not have a stable operating point
SUPP_SHORT	0	rwh	**Supply Short** Indicates the status of the main supply (VDDP and VDDC) after a certain time of power-on reset 0_B **Main Supply ok**, VDDP or VDDC are within the expected range 1_B **Main Supply short**, VDDP or VDDC do not have a stable operating point

5.4.2.2 PMU Wake Fail Register

This register contains status information for VDDEXT failures.

Wake Status Fail Register

This register is reset by RESET_TYPE_0.

WAKE_STS_FAIL
Wake Status Fail Register **(080_H)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES					**VDDEXTSHORT**	**RES**	**SUPPFAIL**
		r			rwh	r	rwh

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Always read as 0
VDDEXTSHORT	2	rwh	**Stop-Exit due to short circuit at the VDDEXT Supply** 0_B No short circuit 1_B Short circuit
RES	1	r	**Reserved** Always read as 0
SUPPFAIL	0	rwh	**Stop-Exit due to overvoltage at the VDDEXT Supply** 0_B No overvoltage 1_B Module suspend enabled

5.5 Wake-up management unit (PMU-WMU)

5.5.1 Functional description

The wake-up management unit (PMU-WMU) mainly handles the wake-up events on LIN, HV monitoring inputs (MON), hardware reset, and all GPIOs belonging to port 0 and port 1. These wake scenarios are possible:

- **Wake-up over port 0 and port 1 pins:** Port 0 and port 1 pins can be configured for wake-up events triggered by rising or falling edges. This can be used to wake up the device from normal Stop mode and Stop mode with the cyclic sense option. To bias the GPIOs, VDDEXT can be used as the voltage source. The wake-up feature from Sleep mode in combination with GPIOs is not possible.
- **Wake-up over hardware reset pin:** It can be used to wake up the device from Stop mode. It cannot be used to wake it up from Sleep mode.
- **Wake-up over MON pins:** The MON pins can be configured for wake-up events triggered by rising or falling edges. This can be used to wake up the device from Stop mode or Sleep mode.
- **LIN:** LIN is a normal wake-up source and cannot be configured.
- **Wake-up on VDDEXT fail from Stop mode:** This wake-up is performed automatically in case of VDDEXT failures, as described in **Power control unit – fail-safe scenarios**.

Note:

1. *The pins of port 2 cannot trigger a wake-up.*
2. *None of the GPIOs is supplied in Sleep mode. Therefore, they cannot be used to wake up the device.*

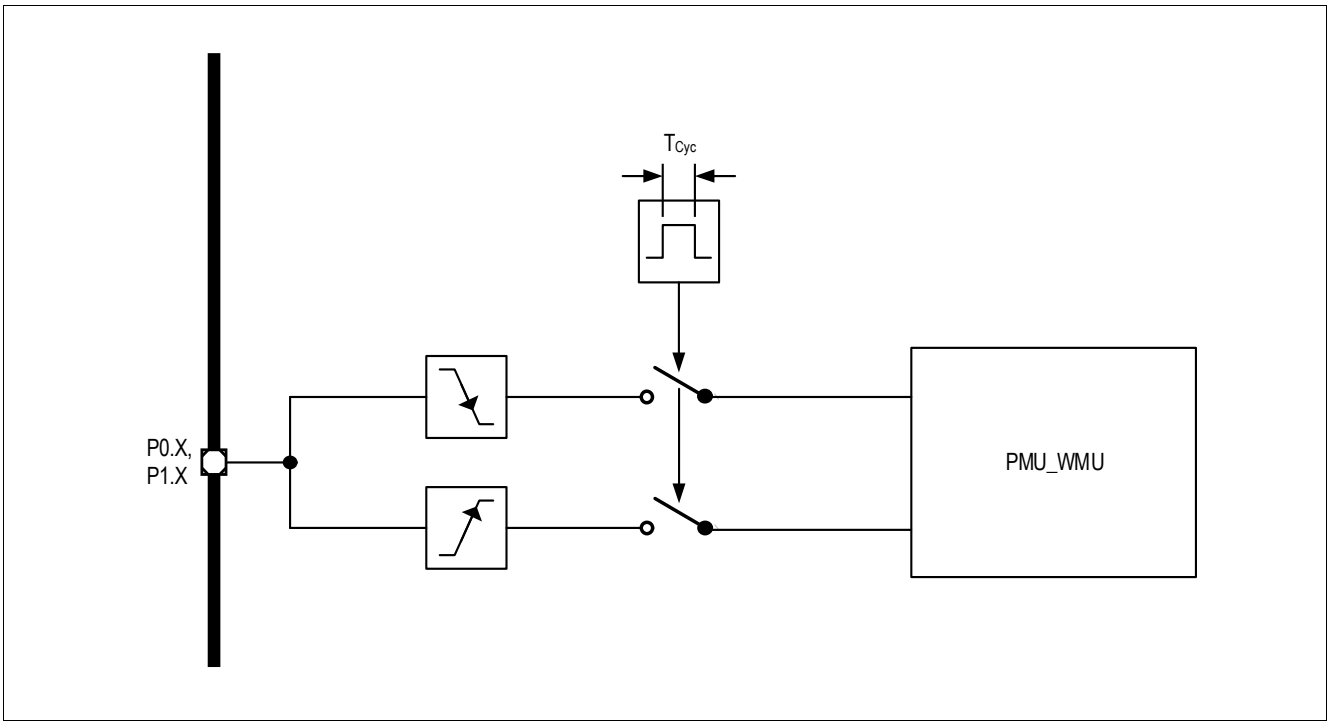


Figure 15 Block diagram of the wake-up management unit with VDDEXT

5.5.2 Register Definition

These registers control the wake-up functions of all GPIOs and provide wake-up status information.

The PMU-WMU can be fully controlled by the registers listed below.

Table 10 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
PMU Wake-Up Configuration Register			
WAKE_CONF_GPIO0_RISE	Wake Configuration GPIO Port 0 Rising Edge Register	$0D8_H$	$0000\ 0000_B$
WAKE_CONF_GPIO0_FALL	Wake Configuration GPIO Port 0 Falling Edge Register	$0DC_H$	$0000\ 0000_B$
WAKE_CONF_GPIO0_CYC	Wake Port 0 Cycle Enabled Register	$0E0_H$	$0000\ 0000_B$
WAKE_CONF_GPIO1_RISE	Wake Configuration GPIO Port 1 Rising Edge Register	$0E4_H$	$0000\ 0000_B$
WAKE_CONF_GPIO1_FALL	Wake Configuration GPIO Port 1 Falling Edge Register	$0E8_H$	$0000\ 0000_B$
WAKE_CONF_GPIO1_CYC	Wake Port 1 Cycle Enabled Register	$0EC_H$	$0000\ 0000_B$
CNF_WAKE_FILTER	PMU Wake-Up Timing Register	$0AC_H$	$0000\ 0000_B$
LIN_WAKE_EN	LIN Wake Enable	050_H	$0000\ 0000_B$
PMU Wake-Up Status Register			
WAKE_STATUS	Main Wake Status Register	000_H	$00xx\ xxxx_B$
WAKE_STS_MON	Wake Source MON Input Register	084_H	$0000\ 0000_B$
WAKE_STS_GPIO0	Wake Status GPIO 0 Register	088_H	$0000\ 0000_B$
WAKE_STS_GPIO1	Wake Status GPIO 1 Register	$08C_H$	$0000\ 0000_B$

The registers are addressed wordwise.

5.5.2.1 PMU Wake-Up Configuration Register

This register controls the wake-up functions of the GPIOs.

Wake Configuration GPIO Port 0 Rising Edge Register

The register is reset by RESET_TYPE_3.

WAKE_CONF_GPIO0_RISE
Wake Configuration GPIO Port 0 Rising Edge Register ($0D8_H$) **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
	RES		**GPIO0_RI_4**	**GPIO0_RI_3**	**GPIO0_RI_2**	**GPIO0_RI_1**	**GPIO0_RI_0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO0_RI_4	4	rw	**Port 0_4 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_RI_3	3	rw	**Port 0_3 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_RI_2	2	rw	**Port 0_2 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_RI_1	1	rw	**Port 0_1 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_RI_0	0	rw	**Port 0_0 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled

Wake Configuration GPIO Port 0 Falling Edge Register

The register is reset by RESET_TYPE_3.

WAKE_CONF_GPIO0_FALL
Wake Configuration GPIO Port 0 Falling Edge Register ($0DC_H$) **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO0_FA_4**	**GPIO0_FA_3**	**GPIO0_FA_2**	**GPIO0_FA_1**	**GPIO0_FA_0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO0_FA_4	4	rw	**Port 0_4 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_FA_3	3	rw	**Port 0_3 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_FA_2	2	rw	**Port 0_2 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_FA_1	1	rw	**Port 0_1 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO0_FA_0	0	rw	**Port 0_0 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled

Wake Port 0 Cycle Enabled Register

The register is reset by RESET_TYPE_3.

WAKE_CONF_GPIO0_CYC
Wake Port 0 Cycle Enabled Register **($0E0_H$)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO0_CYC_4**	**GPIO0_CYC_3**	**GPIO0_CYC_2**	**GPIO0_CYC_1**	**GPIO0_CYC_0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO0_CYC_4	4	rw	**GPIO0_4 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO0_CYC_3	3	rw	**GPIO0_3 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO0_CYC_2	2	rw	**GPIO0_2 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO0_CYC_1	1	rw	**GPIO0_1 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO0_CYC_0	0	rw	**GPIO0_0 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled

Wake Configuration GPIO Port 1 Rising Edge Register

The register is reset by RESET_TYPE_3.

WAKE_CONF_GPIO1_RISE
Wake Configuration GPIO Port 1 Rising Edge Register ($0E4_H$) **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO1_RI_4**	**GPIO1_RI_3**	**GPIO1_RI_2**	**GPIO1_RI_1**	**GPIO1_RI_0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO1_RI_4	4	rw	**Port 1_4 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_RI_3	3	rw	**Port 1_3 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_RI_2	2	rw	**Port 1_2 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_RI_1	1	rw	**Port 1_1 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_RI_0	0	rw	**Port 1_0 Wake-Up on Rising Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled

Wake Configuration GPIO Port 1 Falling Edge Register

The register is reset by RESET_TYPE_3.

WAKE_CONF_GPIO1_FALL
Wake Configuration GPIO Port 1 Falling Edge Register ($0E8_H$) **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO1_FA_4**	**GPIO1_FA_3**	**GPIO1_FA_2**	**GPIO1_FA_1**	**GPIO1_FA_0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO1_FA_4	4	rw	**Port 1_4 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_FA_3	3	rw	**Port 1_3 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_FA_2	2	rw	**Port 1_2 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_FA_1	1	rw	**Port 1_1 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled
GPIO1_FA_0	0	rw	**Port 1_0 Wake-Up on Falling Edge enable** 1_B **ENABLE**, wake-up enabled 0_B **DISABLE**, wake-up disabled

Wake Port 1 Cycle Enabled Register

The register is reset by RESET_TYPE_3.

WAKE_CONF_GPIO1_CYC
Wake Port 1 Cycle Enabled Register **($0EC_H$)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO1_CYC_4**	**GPIO1_CYC_3**	**GPIO1_CYC_2**	**GPIO1_CYC_1**	**GPIO1_CYC_0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO1_CYC_4	4	rw	**GPIO1_4 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO1_CYC_3	3	rw	**GPIO1_3 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO1_CYC_2	2	rw	**GPIO1_2 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO1_CYC_1	1	rw	**GPIO1_1 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled
GPIO1_CYC_0	0	rw	**GPIO1_0 input for cycle sense enable** 1_B **ENABLE**, input for cycle sense enabled 0_B **DISABLE**, input for cycle sense disabled

PMU Wake-Up Timing Register

These registers are for wake-up control of all wake-up capable general purpose inputs outputs.

The register is reset by RESET_TYPE_2.

CNF_WAKE_FILTER
PMU Wake-Up Timing Register **($0AC_H$)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES				**CNF_GPIO_FT**		**CNF_MON_FT**	**CNF_LIN_FT**
r				rw		rw	rw

Field	Bits	Type	Description
RES	7:4	r	**Reserved** Always read as 0.
CNF_GPIO_FT	3:2	rw	**Wake-Up Filter time for General Purpose IO** Specifies the filter time for the wake-up 00_B **10_us**, 10 µs filter time 01_B **20_us**, 20 µs filter time 10_B **40_us**, 40 µs filter time 11_B **5_us**, 5 µs filter time
CNF_MON_FT	1	rw	**Wake-Up Filter time for Monitoring Inputs** Specifies the filter time for the wake-up 0_B **20_us**, 20 µs filter time 1_B **40_us**, 40 µs filter time
CNF_LIN_FT	0	rw	**Wake-Up Filter time for LIN WAKE** Specifies the filter time for the wake-up 0_B **30_us**, 30 µs filter time 1_B **50_us**, 50 µs filter time

LIN Wake Enable

The register is reset by RESET_TYPE_2.

LIN_WAKE_EN
LIN Wake Enable **(050_H)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
LIN_EN	**RES**						
rw				r			

Field	Bits	Type	Description
LIN_EN	7	rw	**Lin Wake enable** 0_B **Disable**, 1_B **Enable**,
RES	6:0	r	**Reserved** Always read as 0.

5.5.2.2 PMU Wake-Up Status Register

Main wake status register

The register is reset by RESET_TYPE_0.

WAKE_STATUS
Main wake status register **(000_H)** **Reset Value: $00xx\ xxxx_B$**

7	6	5	4	3	2	1	0
RES		FAIL	CYC_WAKE	GPIO1	GPIO0	MON_WAKE	LIN_WAKE
r		r	rhc	r	r	r	rhc

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Always read as 0
FAIL	5	r	**Wake-Up after VDDEXT Fail** 0_B No wake-up occurred 1_B Wake-up occurred
CYC_WAKE	4	rhc	**Wake-Up caused by Cyclic Wake** *Note: This flag is cleared by read operations.* 0_B No wake-up occurred 1_B Wake-up occurred
GPIO1	3	r	**Wake-Up via GPIO1 which is a logical OR combination of all Wake_STS_GPIO1 bits** 0_B No wake-up occurred 1_B Wake-up occurred
GPIO0	2	r	**Wake-Up via GPIO0 which is a logical OR combination of all Wake_STS_GPIO0 bits** 0_B No wake-up occurred 1_B Wake-up occurred
MON_WAKE	1	r	**Wake-Up via MON** 0_B No wake-up occurred 1_B Wake-up occurred
LIN_WAKE	0	rhc	**Wake-Up via LIN- Message** *Note: This flag is cleared by read operations.* 0_B No wake-up occurred 1_B Wake-up occurred

Wake Source MON Input Register

The register is reset by RESET_TYPE_0.

WAKE_STS_MON
Wake Source MON Input Register **(084_H)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**RES**				**WAKE_STS**
r			r				rhc

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
RES	4:1	r	**Reserved** Always read as 0
WAKE_STS	0	rhc	**Status of MON** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,

Wake Status GPIO 0 Register

The register is reset by RESET_TYPE_0.

WAKE_STS_GPIO0
Wake Status GPIO 0 Register **(088_H)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO0_STS_4**	**GPIO0_STS_3**	**GPIO0_STS_2**	**GPIO0_STS_1**	**GPIO0_STS_0**
r			rhc	rhc	rhc	rhc	rhc

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO0_STS_4	4	rhc	**Status of GPIO0_4** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO0_STS_3	3	rhc	**Status of GPIO0_3** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO0_STS_2	2	rhc	**Status of GPIO0_2** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO0_STS_1	1	rhc	**Status of GPIO0_1** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO0_STS_0	0	rhc	**Status of GPIO0_0** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,

Wake Status GPIO 1 Register

The register is reset by RESET_TYPE_0.

WAKE_STS_GPIO1
Wake Status GPIO 1 Register **($08C_H$)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES			**GPIO1_STS_4**	**GPIO1_STS_3**	**GPIO1_STS_2**	**GPIO1_STS_1**	**GPIO1_STS_0**
r			rhc	rhc	rhc	rhc	rhc

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Always read as 0
GPIO1_STS_4	4	rhc	**Wake GPIO1_4** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO1_STS_3	3	rhc	**Wake GPIO1_3** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO1_STS_2	2	rhc	**Wake GPIO1_2** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO1_STS_1	1	rhc	**Wake GPIO1_1** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,
GPIO1_STS_0	0	rhc	**Wake GPIO1_0** *Note: This flag is cleared by read operations.* 0_B **No wake-up detected**, 1_B **Wake-up detected**,

5.6 Cyclic management unit (PMU-CMU)

5.6.1 Functional description

The cyclic management unit controls the timing sequence in cyclic sense or cyclic wake operation. The unit operates with the LP_CLK2 clock.

5.6.2 Cyclic Sense mode

To select a dedicated GPIO0.x or GPIO1.x pin for cyclic sense mode, the corresponding GPIO0_CYC_x or GPIO1_CYC_x bit must be set in the **WAKE_CONF_GPIO0_CYC** **or** **WAKE_CONF_GPIO1_CYC** register, respectively.

In this configuration, the wake-up information of the configured GPIO0.x or GPIO1.x pin is only accepted during the sensing time – the time when the VDDEXT supply (internal VDDEXT_CYC_ON gating signal) is on (see **Figure 16**). The sensing time where the enable signal is active, will be set in the **CNF_CYC_SENSE** and **CNF_CYC_SAMPLE_DEL**-SFR register. The bits of the **CNF_CYC_SENSE** register specify the dead time (T_{Dead}). The **CNF_CYC_SAMPLE_DEL** register specifies the sample delay. When the sample delay has elapsed, the selected wake source is evaluated by the wake-up management unit. If the status of the wake-up source has not changed since the last sampling, the PMU enters the power-down period again, the length of which is specified as the dead time.

If the status of the wake-up source, sampled during the sensing time, has changed from the previous cycle, the device wakes up.

After a valid wake-up event, the start-up sequence is similar to the asynchronous wake-up, and the system enters the Startup mode automatically. If the PMU detects a wake-up during Cyclic Sense, the enable signal of the voltage source (VDDEXT) stays active as long the application software does not disable this signal.

Figure 16 illustrates the principle of the cyclic sense mode. The VDDEXT supply is used as the voltage source and a GPIO pin as the wake-up source. This can be applied to all GPIOs from port 0 and port 1.

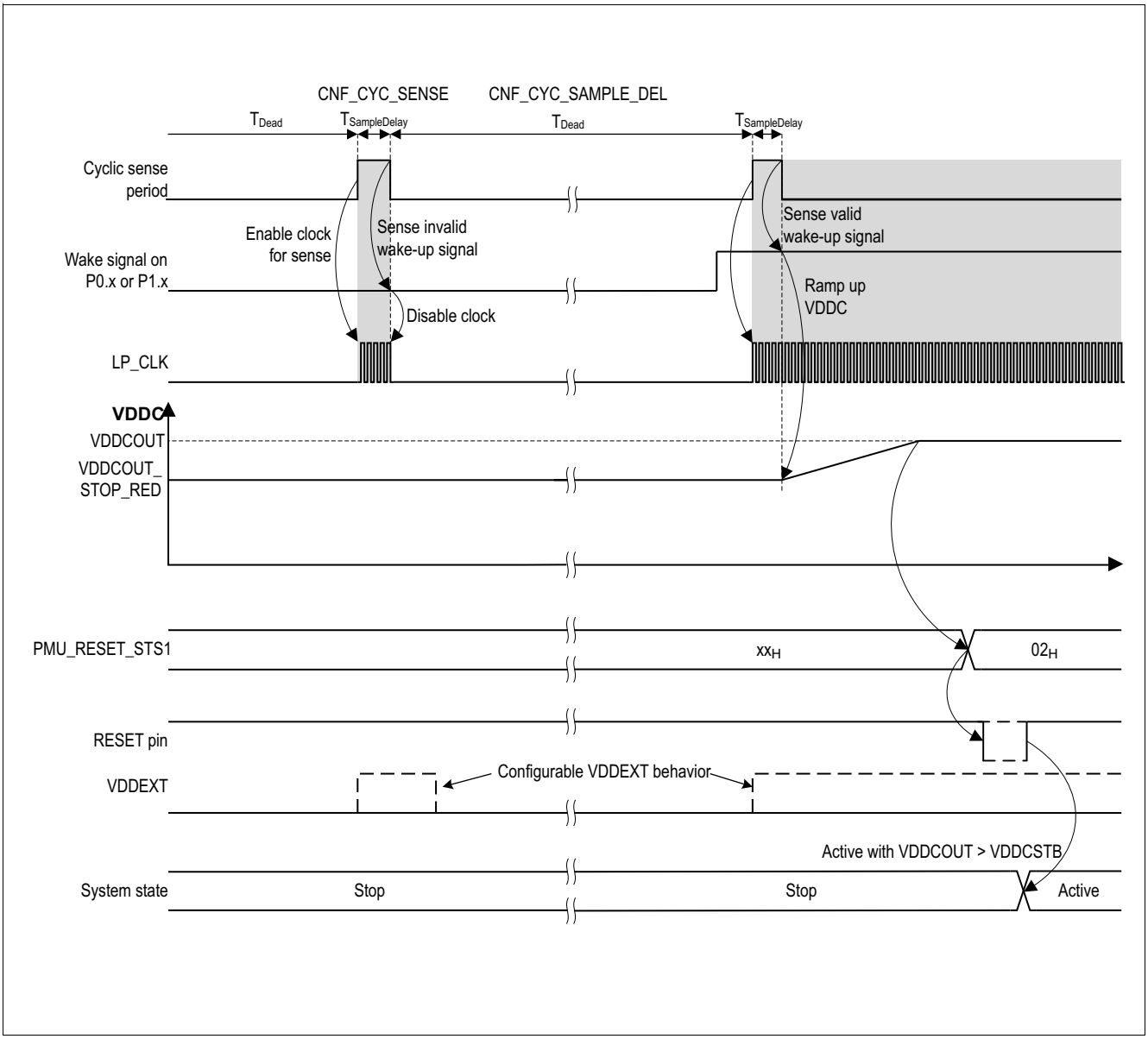


Figure 16 Timing diagram for cyclic sense

The arrows in the diagram above indicate causal connections between the various steps and not necessarily voltage-based connections.

5.6.2.1 Configuring cyclic sense mode

The configuration process for the cyclic sense mode is shown in **Figure 17**.

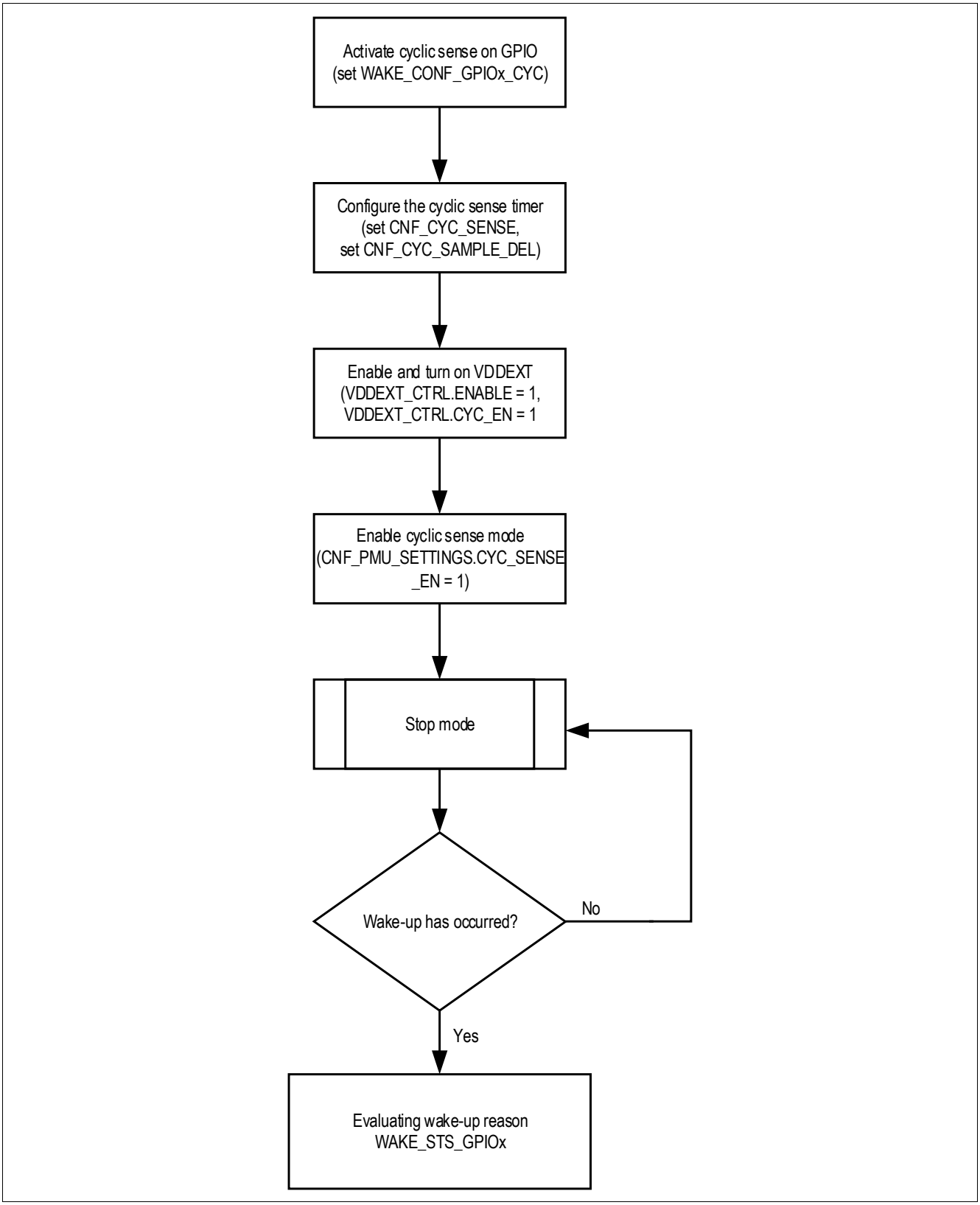


Figure 17 Configuration flow of cyclic sense mode

5.6.3 Cyclic Wake mode

Cyclic Wake mode provides a synchronous wake-up after a predefined time interval in Sleep mode or Stop mode. Once the time interval has elapsed, the PMU enters Startup mode and proceeds to Active mode where the software takes control of the system. The cyclic wake interval is set in the **CNF_CYC_WAKE** register.

5.6.4 Register Definition

Table 11 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Cyclic Wake Configuration Registers			
CNF_PMU_SETTINGS	PMU Settings Register	020_H	$0000\ 0100_B$
CNF_CYC_SENSE	Dead Time in Cyclic Sense Register	028_H	$0000\ 0000_B$
CNF_CYC_WAKE	Dead Time in Cyclic Wake Register	$02C_H$	$0011\ 0111_B$
CNF_CYC_SAMPLE_DEL	Sample Delay in Cyclic Sense Register	030_H	$0000\ 0000_B$

The registers are addressed wordwise.

5.6.4.1 Cyclic Wake Configuration Registers

PMU Settings Register

The register is reset by RESET_TYPE_2.

CNF_PMU_SETTINGS
PMU Settings Register **(020_H)** **Reset Value: $0000\ 0100_B$**

7	6	5	4	3	2	1	0
EN_VDDEXT_OC_OFF_N	**RES**		**RES**	**CYC_SENSE_EN**	**CYC_WAKE_EN**	**EN_0V9_N**	**WAKE_W_RST**
rw	r		r	rw	rw	rw	rw

Field	Bits	Type	Description
EN_VDDEXT_OC_OFF_N	7	rw	**Disabling VDDEXT Shutdown in Overload Condition** This bit enables the Hall Supply feature to be switched off when an overload condition is detected. 0_B Shutdown enabled 1_B Shutdown disabled
RES	6:5	r	**Reserved** Always read as 0
RES	4	r	**Reserved** Always read as 0
CYC_SENSE_EN	3	rw	**Enabling Cyclic Sense** This bit enables the cyclic sense feature for the power-saving modes. 0_B Cyclic sense disabled 1_B Cyclic sense enabled

Field	Bits	Type	Description
CYC_WAKE_EN	2	rw	**Enabling Cyclic Wake** This bit enables the cyclic wake feature for the power-saving modes. 0_B Cyclic wake disabled 1_B Cyclic wake enabled
EN_0V9_N	1	rw	**Disables the reduction of the VDDC regulator output to $V_{DDCOUT_Stop_Red}$ during Stop-Mode** 0_B Output voltage reduction enabled 1_B Output voltage reduction disabled
WAKE_W_RST	0	rw	**Wake-Up with reset execution** Enables the Stop-Exit with reset execution 0_B Stop-Exit without reset execution 1_B Stop-Exit with reset execution

Dead Time in Cyclic Sense Register

The register is reset by RESET_TYPE2.

The dead time of Cyclic Sense is configured in the CNF_CYC_SENSE register.

The dead time is calculated by:

$$4^{(E1E0)} \cdot (M3M2M1M0+1) \cdot 2ms$$

E1E0 represents the register flags CYC_SENSE_E01 and M3M2M1M0 represents the register flags CYC_SENSE_M03.

CNF_CYC_SENSE
Dead Time in Cyclic Sense Register **(028_H)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
OSC_100kHz_EN	**RES**	**E01**		**M03**			
rw	r	rw		rw			

Field	Bits	Type	Description
OSC_100kHz_EN	7	rw	**100 kHz Oscillator Enable** Enables the 100 kHz oscillator output from the PMU to be provided to Timer3 0_B **DISABLE**, oscillator is disabled 1_B **ENABLE**, oscillator is enabled
RES	6	r	**Reserved** Always read as 0
E01	5:4	rw	**Exponent** 00_B Exponent value is 0 01_B Exponent value is 1 10_B Exponent value is 2 11_B Exponent value is 3
M03	3:0	rw	**Mantissa** Mantissa value is calculated as CYC_SENSE_M03 + 1 0000_BMantissa value is 1 1111_BMantissa value is 16

Dead Time in Cyclic Wake Register

The register is reset by RESET_TYPE2.

The dead time for the cyclic wake-up is configured in the CNF_CYC_WAKE register.

The dead time is calculated by:

$$4^{(E1E0)} \cdot (M3M2M1M0+1) \cdot 2ms$$

E1E0 represents the register flags CYC_WAKE_E01 and M3M2M1M0 represents the register flags CYC_WAKE_M03.

CNF_CYC_WAKE
Dead Time in Cyclic Wake Register **($02C_H$)** **Reset Value: $0011\ 0111_B$**

7	6	5	4	3	2	1	0
RES		**E01**		**M03**			
r		rw		rw			

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Always read as 0
E01	5:4	rw	**Exponent** 00_B Exponent value is 0 01_B Exponent value is 1 10_B Exponent value is 2 11_B Exponent value is 3
M03	3:0	rw	**Mantissa** Mantissa value is calculated as CYC_WAKE_M03 + 1 0000_BMantissa value is 1 1111_BMantissa value is 16

Sample Delay in Cyclic Sense Register

This register is reset by RESET_TYPE_2.

$$(M3M2M1M0+1)\cdot 10\mu s$$

M3M2M1M0 represents the register flags M03.

CNF_CYC_SAMPLE_DEL
Sample Delay in Cyclic Sense Register **(030_H)** **Reset Value: $0000\ 0000_B$**

7	6	5	4	3	2	1	0
RES				M03			
r				rw			

Field	Bits	Type	Description
RES	7:4	r	**Reserved** Always read as 0
M03	3:0	rw	**Mantissa** Mantissa value is calculated as: M03 0000_BMantissa value M3M2M1M0 is 0 1111_BMantissa value M3M3M1M0 is 15

5.7 Reset management unit (PMU-RMU)

5.7.1 Functional description

The reset management unit (PMU-RMU) controls the reset behavior of the entire device. The master reset of the device is the power-on reset of the PMU itself. This reset is generated by the power-down supply and it is released when the battery voltage (V_S) reaches the minimum supply voltage for Active mode. Then the PMU starts the sequence to power up the supply generation module, which ends with the release of the MCU reset. The embedded system now works in Active mode. This scenario is signalled by the PMU_1V5DidPOR flag in the **PMU_RESET_STS1**. The figure below shows the power-on reset behavior.

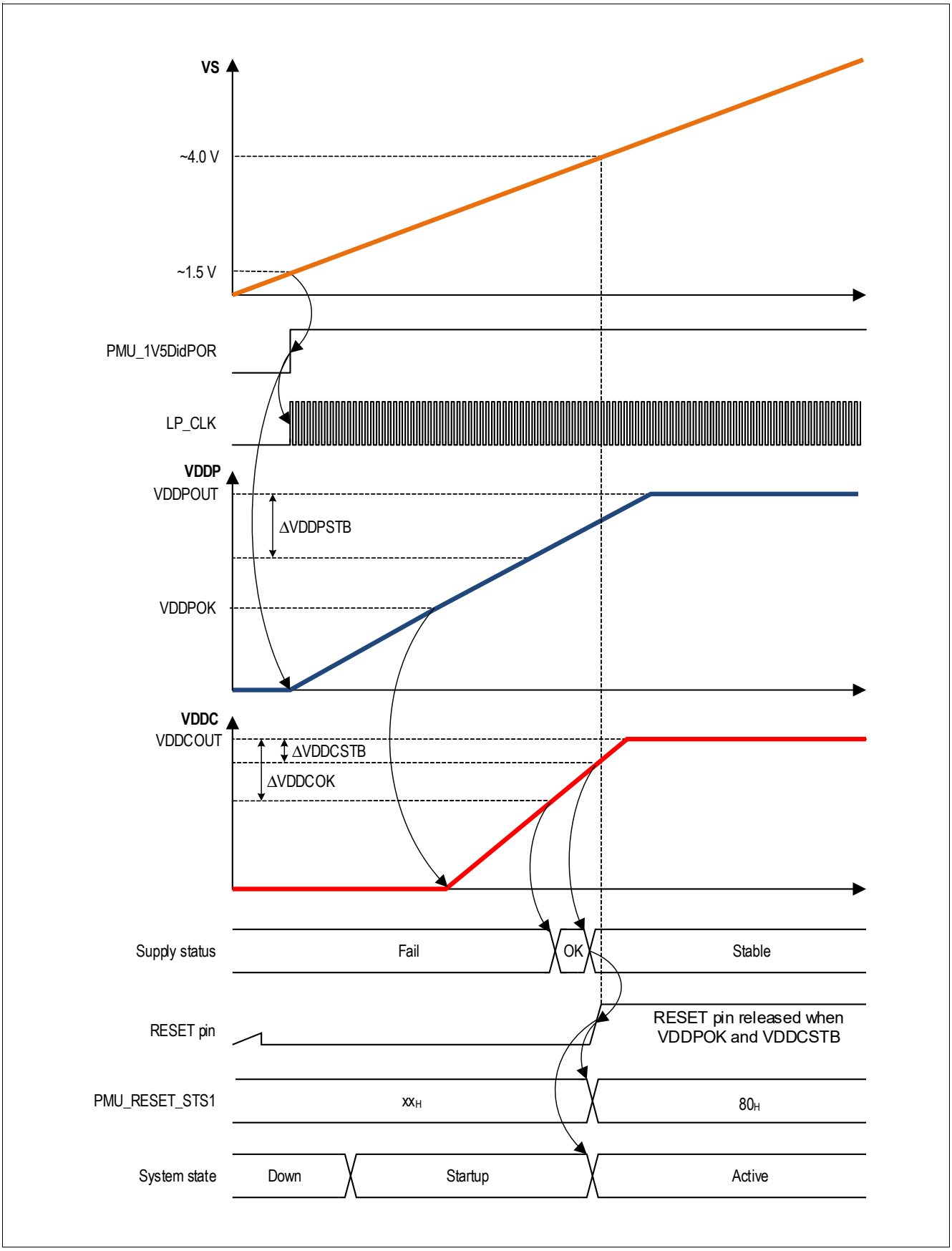


Figure 18 Power-on and startup behavior of reset

The arrows in the diagram above indicate causal connections between the various steps and not necessarily voltage-based connections.

In case of a Sleep mode exit a similar sequence like for battery ramp-up starts is used. If this sequence ends successfully then the PMU also releases the reset of the MCU. From the MCU point of view there is no difference to the battery ramp-up. Only inside of the PMU-RMU the identification bit **PMU_SleepEx** is set instead of the power-on identification bit. The power down sequence looks as follows:

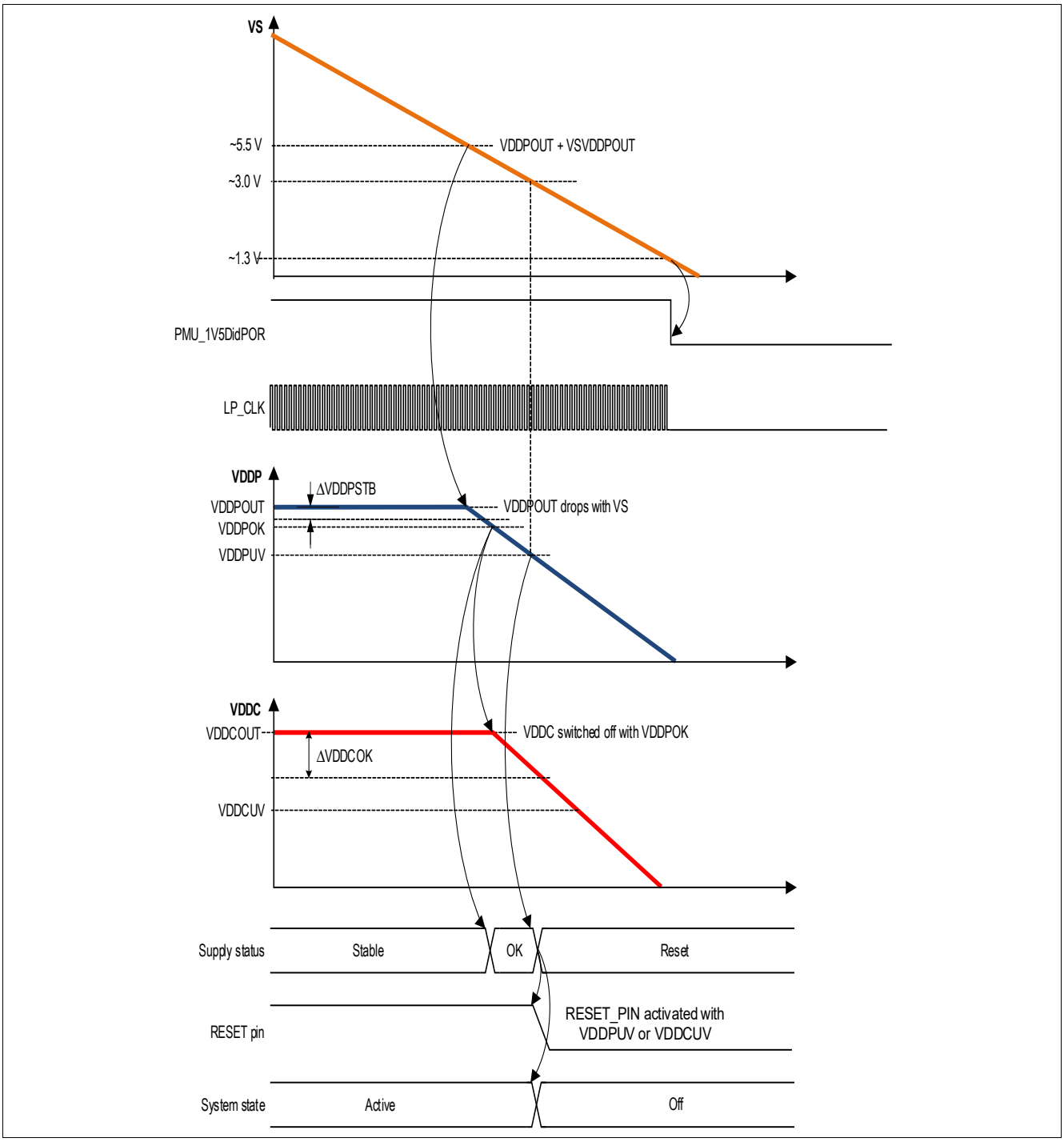


Figure 19 Power-down and power loss sequence

The arrows in the diagram above display indicate causal connections between the various steps and not necessarily voltage-based connections.

In the default configuration, the wake-up from Stop mode works without a reset. To wake the device up with a reset, the corresponding WAKE_W_RST SFR bit in the **CNF_PMU_SETTINGS** register must be configured. In that configuration, the wake-up signal sets the dedicated PMU_WAKE identification bit, which can be checked by the application software.

The third hardware-related reset source is the pin reset. The pin itself is supplied by the VDDP domain, which is available in Active mode and Stop mode. Therefore, the reset pin can be used only in Active mode and Stop mode. Due to the bidirectional use of the pin, the pin reset request is gated during the execution of another reset request (for example, soft reset). For this purpose, the pin reset request must be stable for more than 500 ns (see **Figure 16**). When a pin reset request arrives in Stop mode, the PMU goes to Active mode and sends the wake-up signal to the MCU. At this time, the PMU_PIN bit in the reset status register is also updated to indicate the reset source. All other reset sources can only affect the system behavior in Active mode.

The reset request caused by an unserved or wrongly served system watchdog is also processed as a hardware-related reset although this reset request is implicitly controlled by user software. The system watchdog only works in Active mode. In this case, it expects a periodic trigger (window watchdog) from the user software. If the trigger is missing, the PMU receives a signal that the watchdog was not serviced, which sets the PMU_ExtWDT identification bit from WDT1. After several clock cycles of the PMU's internal oscillator LP_CLK, the PMU resets the MCU.

The software reset and the reset request caused by the MCU's internal watchdog are controlled explicitly by user software and can be used only in Active mode. From the system's point of view, these two reset sources have the lowest priority. The software-related reset is executed within two MCU clock cycles. The system clock of the PMU works independently of the MCU clock. Due to these system conditions, the PMU processes the software-related resets asynchronously from its internal system clock. When a reset has been caused by software, the **PMU_SOFT** bit is set. When a reset has been caused by the MCU's internal watchdog, the **PMU_IntWDT** bit is set. Both flags are located in the **PMU_RESET_STS2** register.

Another reset source is the PGU module. If the main voltage regulators (VDDP and VDDC) fail, the system executes a system reset and enters Sleep mode afterwards. This case is indicated by setting the **SYS_FAIL** bit.

Some SFR registers are reset only by a certain reset type. A reset type combines several of the reset sources described above; when a reset type is listed for a register, it is reset by any of the resets combined in the type. For example, a register with RESET_TYPE_1 is reset when the system reset was caused by the SoC power-on, a PMU_FAIL, or a WDT1_FAIL.

A particular reset source can reset registers with different reset types. For example, a reset caused by exiting the Sleep mode resets registers with RESET_TYPE_3 and registers with RESET_TYPE_4.

Registers for which no reset type is listed (except NMI status flags) are always reset when a system reset occurs.

The reset sources belonging to each reset type are shown in **Figure 20**.

	SoC power-on	Sleep mode exit	Stop mode exit with reset	Pin reset	WDT1 wrong service	SOFT reset	WDT (int.) timeout	PMU_FAIL (error_sup = 5)	WDT1_FAIL (error_wdt1 = 5)
RESET_TYPE_0	X								
RESET_TYPE_1	X							X	X
RESET_TYPE_2	X		X	X	X	X	X	X	X
RESET_TYPE_3	X	X	X	X	X	X	X	X	X
RESET_TYPE_4	X	X	X	X	X			X	X

Figure 20 Reset types of SFRs provided by the PMU-RMU

The resets described above are combined into five main reset types:

- RESET_TYPE_0 is triggered by:
  - SoC power-on (PMU_1V5DidPOR = 1): This reset is triggered when the power-down supply detects undervoltage.
- RESET_TYPE_1 is an OR of:
  - SoC power-on (PMU_1V5DidPOR = 1): This reset is triggered when the power-down supply detects undervoltage.
  - PMU_FAIL: This reset is triggered when the VDDC or VDDP supply fail for the fifth consecutive time.
  - WDT1_FAIL: This reset is triggered when the WDT1 was not correctly triggered five consecutive times.
- RESET_TYPE_2 is an OR of:
  - SoC power-on (PMU_1V5DidPOR = 1): This reset is triggered when the power-down supply detects undervoltage.
  - Pin reset (PMU_PIN = 1): This reset is triggered when the RESET pin is pulled down.
  - WDT1 wrong service (PMU_ExtWDT = 1): This reset is triggered if the WDT1 was serviced wrongly.
  - WDT (int.) timeout (PMU_IntWDT = 1): This reset is triggered when there is an incorrect service of the internal WDT.
  - Soft reset (PMU_SOFT = 1): This reset is a software-related reset that resets the CPU and the CPU registers.
  - Stop mode exit with reset (PMU_Wake= 1): This reset is triggered when the device wakes up from Stop mode and is configured to do so with a reset.
  - PMU_FAIL: This reset is triggered when the VDDC or VDDP supply fail for the fifth consecutive time.
  - WDT1_FAIL: This reset is triggered when the WDT1 was not correctly triggered five times.
- RESET_TYPE_3 is an OR of:
  - SoC power-on (PMU_1V5DidPOR = 1): This reset is triggered when the power-down supply detects undervoltage.

  - Pin reset (PMU_PIN = 1): This reset is triggered when the RESET pin is pulled down.
  - WDT1 wrong service (PMU_ExtWDT = 1). This reset is triggered if the WDT1 was serviced wrongly.
  - WDT (int.) timeout (PMU_IntWDT = 1): This reset is triggered when there is an incorrect service of the internal WDT.
  - SOFT reset (PMU_SOFT = 1): This reset is a software-related reset that resets the CPU and the CPU registers.
  - Stop mode exit with reset (PMU_Wake = 1): This reset is triggered when the device wakes up from Stop mode and is configured to do so with a reset.
  - Sleep mode exit (PMU_SleepEx = 1): This reset is triggered when the device wakes up from Sleep mode.
  - PMU_FAIL: This reset is triggered when the VDDC or VDDP supply fail for the fifth consecutive time.
  - WDT1_FAIL: This reset is triggered when the WDT1 was not correctly triggered five times.
- RESET_TYPE_4 is an OR of:
  - SoC power-on (PMU_1V5DidPOR = 1). This reset is triggered when the power-down supply detects undervoltage.
  - Pin reset (PMU_PIN = 1): This reset is triggered when the RESET pin is pulled down.
  - WDT1 wrong service (PMU_ExtWDT = 1): This reset is triggered if the WDT1 was serviced wrongly.
  - Stop mode exit with reset (PMU_Wake = 1): This reset is triggered when the device wakes up from Stop mode and is configured to do so with a reset.
  - Sleep mode exit (PMU_SleepEx = 1): This reset is triggered when the device wakes up from Sleep mode.
  - PMU_FAIL: This reset is triggered when the VDDC or VDDP fail for the fifth consecutive time.
  - WDT1_FAIL: This reset is triggered when the WDT1 was not correctly triggered five times.

5.7.2 Register Definition

Table 12 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Reset Management Unit Registers (RMU)			
CNF_RST_TFB	Reset Blind Time Register (BE, BF, and A design steps)	$06C_H$	$0000\ 0003_H$
CNF_RST_TFB	Reset Blind Time Register (UH design step)	$06C_H$	$0000\ 0001_H$
PMU_RESET_STS1	Reset Status Hard Register	010_H	$0000\ 0000_H$
PMU_RESET_STS2	Reset Status Soft Register	014_H	$0000\ 0000_H$

The registers are addressed wordwise.

5.7.2.1 Reset Management Unit Registers (RMU)

Reset Blind Time Register (BE, BF, and A design steps)

The reset pin carries a bidirectional signal. Every reset is signaled on the pin for several hundred nanoseconds. To avoid reset deadlocks, there is a programmable reset blind time, during which no hardware pin reset is accepted. The blind time covers the period in the reset process during which the reset pin acts as an active reset output. It is required to avoid infinite reset loops. The register is reset by RESET_TYPE_2.

CNF_RST_TFB
Reset Blind Time Register **($06C_H$)** **Reset Value: $0000\ 0003_H$**

7	6	5	4	3	2	1	0
RES						**RST_TFB**	
r						rw	

Field	Bits	Type	Description
RES	7:2	r	**Reserved** Always read as 0
RST_TFB	1:0	rw	**Reset Pin Blind Time Selection Bits** These bits specify the blind time for reset input sampling. 00_B **RST_TFB_0**, 0.5 µs typ. 01_B **RST_TFB_1**, 1 µs typ. 10_B **RST_TFB_2**, 5 µs typ. 11_B **RST_TFB_3**, 31 µs typ. (default setting)

Reset Blind Time Register (UH design step)

The reset pin carries a bidirectional signal. Every reset is signaled on the pin for several hundred nanoseconds. To avoid reset deadlocks, there is a programmable reset blind time, during which no hardware pin reset is accepted. The blind time covers the period in the reset process during which the reset pin acts as an active reset output. It is required to avoid infinite reset loops. The register is reset by RESET_TYPE_2.

CNF_RST_TFB
Reset Blind Time Register **($06C_H$)** **Reset Value: $0000\ 0001_H$**

7	6	5	4	3	2	1	0
RES						**RST_TFB**	
r						rw	

Field	Bits	Type	Description
RES	7:2	r	**Reserved** Always read as 0
RST_TFB	1:0	rw	**Reset Pin Blind Time Selection Bits** These bits specify the blind time for reset input sampling. 00_B RST_TFB_0, 10 µs typ. 01_B RST_TFB_1, 30 µs typ. (default setting) 10_B RST_TFB_2, 200 µs typ. 11_B RST_TFB_3, 1240 µs typ.

Reset Status Hard Register

The PMU_RESET_STS1 register indicates the source of every executed reset request. The PMU updates the register flag that corresponds to the type of the executed reset. To clear the information of the PMU_RESET_STS1 register, you must overwrite the corresponding bit with zero. The register is reset by RESET_TYPE_0.

PMU_RESET_STS1
Reset Status Hard Register **(010_H)** **Reset Value: $0000\ 0000_H$**

7	6	5	4	3	2	1	0
PMU_1V5DidPOR	**PMU_PIN**	**PMU_ExtWDT**	**PMU_ClkWDT**	**PMU_LPR**	**PMU_SleepEX**	**PMU_WAKE**	**SYS_FAIL**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
PMU_1V5DidPOR	7	rwh	**Power-On Reset Flag** 0_B No power-on reset was executed 1_B Power-on reset was executed
PMU_PIN	6	rwh	**PIN-Reset Flag** 0_B No pin reset was executed 1_B Pin reset was executed
PMU_ExtWDT	5	rwh	**External Watchdog (WDT1) Reset Flag** 0_B No external watchdog reset was executed 1_B External watchdog reset was executed
PMU_ClkWDT	4	rwh	**Clock Watchdog (CLKWDT) Reset Flag** 0_B No clock watchdog reset was executed 1_B Clock watchdog reset was executed
PMU_LPR	3	rwh	**Low Priority Resets (see PMU_RESET_STS2)** 0_B No low-priority reset was executed 1_B Low-priority reset was executed
PMU_SleepEX	2	rwh	**Flag which indicates a reset caused by Sleep-Exit** 0_B No reset caused by a Sleep exit was executed 1_B Reset caused by a Sleep exit was executed
PMU_WAKE	1	rwh	**Flag which indicates a reset caused by Stop-Exit**[1)] *Note: A Stop exit with reset must be configured explicitly in the **CNF_PMU_SETTINGS**.WAKE_W_RST bit.* 0_B No reset caused by a Stop exit was executed 1_B A reset caused by a Stop exit was executed
SYS_FAIL	0	rwh	**Flag which indicates a reset caused by a System Fail reported in the corresponding Fail Register** 0_B No reset caused by a system fail was executed 1_B A reset caused by a system fail was executed

1) This flag is always set (together with the PMU_PIN flag) when a pin reset occurs in Stop mode, but not otherwise.

Reset Status Soft Register

The PMU_RESET_STS register indicates the source of every executed low-priority reset request. The PMU updates the register flag that corresponds to the type of the executed reset. To clear the information of the PMU_RESET_STS register, you must overwrite the corresponding bit with zero. The register is reset by RESET_TYPE_0.

*Note: The PMU_RESET_STS2 register is also cleared when **PMU_RESET_STS1**.PMU_LPR is cleared.*

PMU_RESET_STS2
Reset Status Soft Register **(014_H)** **Reset Value: $0000\ 0000_H$**

7	6	5	4	3	2	1	0
RES					**LOCKUP**	**PMU_SOFT**	**PMU_IntWDT**
		r			rwh	rwh	rwh

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Always read as 0
LOCKUP	2	rwh	**Lockup-Reset Flag** 0_B No lockup reset was executed 1_B A lockup reset was executed
PMU_SOFT	1	rwh	**Soft-Reset Flag** 0_B No soft reset was executed 1_B A soft reset was executed
PMU_IntWDT	0	rwh	**Internal Watchdog Reset Flag** 0_B No internal watchdog reset was executed 1_B An internal watchdog reset was executed

5.8 PMU data storage area

5.8.1 Functional description

The PMU lets the system store data in registers that will retain their values when the device is set to Sleep mode. There is a total of 6 × 8 bit for devices with Stop mode available.

5.8.2 Register definition

Table 13 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Data Storage Registers			
GPUDATA00	General Purpose User DATA0	$0C0_H$	00_H
GPUDATA01	General Purpose User DATA1	$0C4_H$	00_H
GPUDATA02	General Purpose User DATA2	$0C8_H$	00_H
GPUDATA03	General Purpose User DATA3	$0CC_H$	00_H
GPUDATA04	General Purpose User DATA4	$0D0_H$	00_H
GPUDATA05	General Purpose User DATA5	$0D4_H$	00_H
SystemStartConfig	System Startup Configuration	$2D4_H$	00_H

The registers are addressed wordwise.

5.8.2.1 Data Storage Registers

General Purpose User DATA0 Storage Register

The register is reset by RESET_TYPE_1.

GPUDATA00
General Purpose User DATA0 **($0C0_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
DATA0							
rw							

Field	Bits	Type	Description
DATA0	7:0	rw	**DATA0 Storage Byte** 1st byte of storage area

General Purpose User DATA1 Storage Register

The register is reset by RESET_TYPE_1.

GPUDATA01
General Purpose User DATA1 **($0C4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
DATA1							
rw							

Field	Bits	Type	Description
DATA1	7:0	rw	**DATA1 Storage Byte** 2nd byte of storage area

General Purpose User DATA2 Storage Register

The register is reset by RESET_TYPE_1.

GPUDATA02
General Purpose User DATA2 **($0C8_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
DATA2							
rw							

Field	Bits	Type	Description
DATA2	7:0	rw	**DATA2 Storage Byte** 3rd byte of storage area

General Purpose User DATA3 Storage Register

The register is reset by RESET_TYPE_1.

GPUDATA03
General Purpose User DATA3 **($0CC_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
DATA3							
rw							

Field	Bits	Type	Description
DATA3	7:0	rw	**DATA3 Storage Byte** 4th byte of storage area

General Purpose User DATA4 Storage Register

The register is reset by RESET_TYPE_1.

GPUDATA04
General Purpose User DATA4 **($0D0_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
DATA4							
rw							

Field	Bits	Type	Description
DATA4	7:0	rw	**DATA4 Storage Byte** 5th byte of storage area

General Purpose User DATA5 Storage Register

The register is reset by RESET_TYPE_1.

GPUDATA05
General Purpose User DATA5 **($0D4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
DATA5							
rw							

Field	Bits	Type	Description
DATA5	7:0	rw	**DATA5 Storage Byte** 6th byte of storage area

System Startup Configuration Register

The register is reset by RESET_TYPE_1.

SystemStartConfig
System Startup Configuration Register **($2D4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES							**MBIST_EN**
rw							rw

Field	Bits	Type	Description
RES	7:1	rw	**Free for use** Always reads back what was written to the bit before.
MBIST_EN	0	rw	**System Startup Configuration Bit for RAM MBIST at Sleep Mode exit** 0_B No MBIST is performed when the device exits Sleep mode. 1_B An MBIST is performed when the device exits Sleep mode.

5.9 Register access notes

Some registers in the PMU module are clocked with MI_CLK. Write operations to them are delayed by one MI_CLK cycle. This issue applies to the **VDDEXT_CTRL** and **SYS_FAIL_STS** registers.

Special considerations apply to clearing interrupt status flags. Under certain coding constellations, an extra write access to the interrupt status flag is necessary to add the extra MI_CLK cycle.

There are two different use cases when clearing interrupt status flags:

1. Between the code statements clearing an interrupt flag and reading that interrupt flag, other code is present.
2. There is no other code between the statements clearing and reading the interrupt status flag.

Use case 1: No extra write attempt to the interrupt status flag required

```
/* Clearing the VDDEXT short error interrupt status flag */
PMU->VDDEXT_CTRL.bit.SHORT = 1;
/* Some other code here */
...
/* Reading the VDDEXT short error interrupt status flag */
if ( PMU->VDDEXT_CTRL.bit.SHORT == 1 )
{
 /* Some additional code here */
}
```

In this example, sufficient time between the commands to clear the register and to read the register is ensured because at least one CPU instruction is executed between the two.

Use case 2: An extra write attempt to the interrupt clear register required

```
/* Clearing the VDDEXT short error interrupt status flag */
PMU->VDDEXT_CTRL.bit.SHORT = 1;
/* Insert another command to clear the status flag in order to add an extra MI_CLK
cycle */
PMU->VDDEXT_CTRL.bit.SHORT = 0;
/* Reading the VDDEXT short error interrupt status flag */
if ( PMU->VDDEXT_CTRL.bit.SHORT == 1 )
{
 /* Some additional code here */
}
```

In this example, an additional write command to clear the register is inserted between the commands to clear the register and to read the register. This adds another MI_CLK cycle, ensuring that the first interrupt status flag write attempt has reached the interrupt status register before it is read again. Without this extra clear attempt, the following interrupt status flag read might still reflect the previous content of the flag (before it was cleared).

In principle, any instruction can be used to add the required MI_CLK cycle. However, the instruction should not cause unwanted side-effects and not be eliminated by compiler optimization. In most cases, declaring the register addresses in question as volatile to the compiler and then writing zero to the clear register achieves this.

We also recommend to write a zero in the second write command because that will not cause functional changes to the register.

6 System control unit – digital modules (SCU-DM)

6.1 Features

- Flexible clock configuration features
- Reset management of all system resets
- System modes control for all power modes (Active mode, Stop mode, Sleep mode)
- Enabling interrupts for many system peripherals
- General-purpose input/output control
- Debug mode control of system peripherals

6.2 Introduction

The system control unit (SCU) supports all central control tasks in the TLE987x. The SCU is made up of the following sub-modules:

- Clock system and control (see **Section 6.3 on Page 94**)
- Reset control (see **Section 6.4 on Page 124**)
- Power management (see **Section 6.5 on Page 129**)
- Interrupt management (see **Section 6.6 on Page 134**)
- General port control (see **Section 6.7 on Page 147**)
- Flexible peripheral management (see **Section 6.8 on Page 159**)
- Module suspension control (see **Section 6.9 on Page 161**)
- Watchdog timer (see **Section 6.10 on Page 163**)
- Error detection and correction in data memory (see **Section 6.11 on Page 169**)
- Miscellaneous control (see **Section 6.12 on Page 172**)

The mapping of all registers is detailed in **Section 6.2.2 on Page 91**

6.2.1 Block diagram

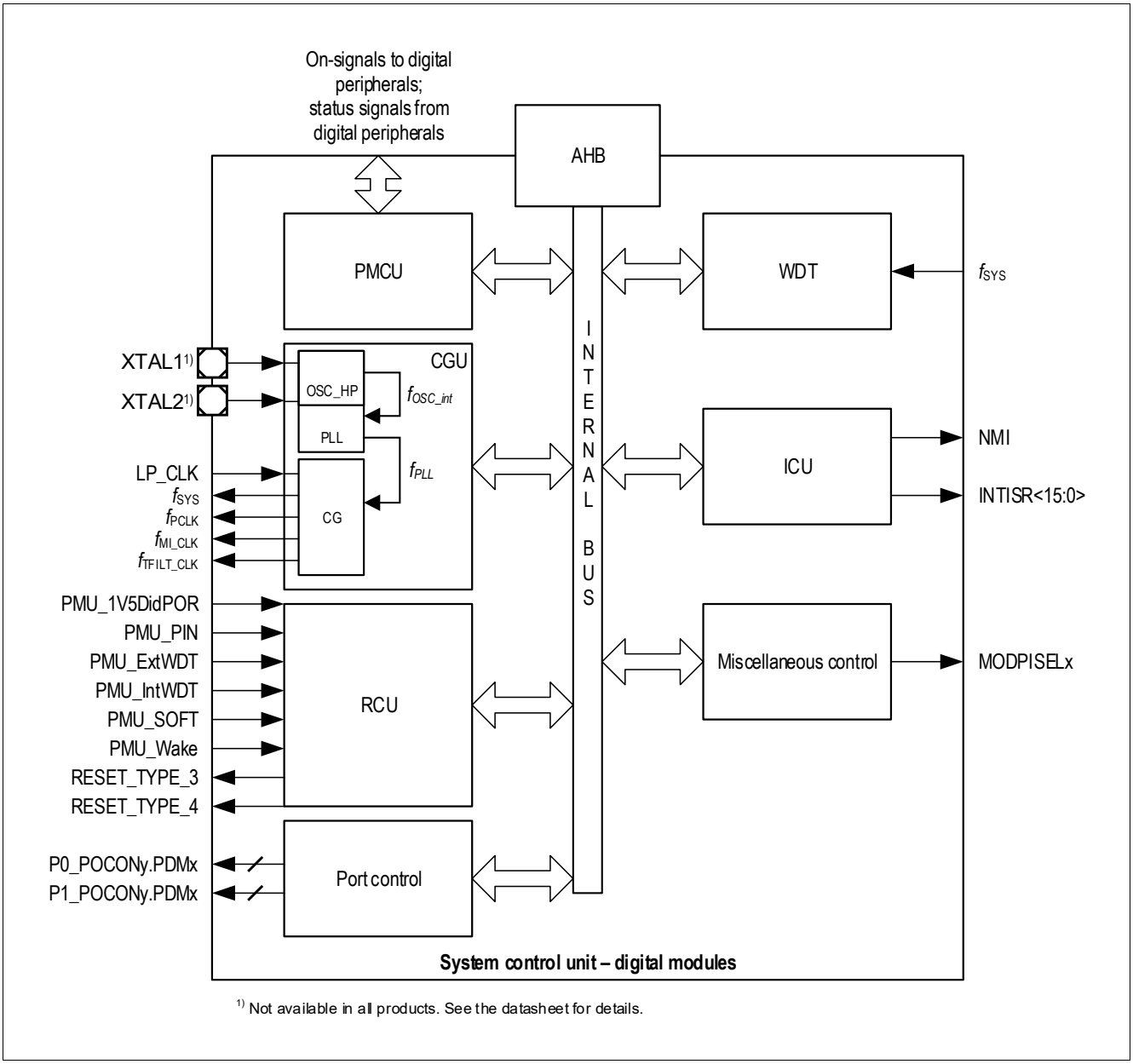


Figure 21 System control unit – digital modules block diagram

AHB (Advanced High-Performance Bus)

PMCU (power module control unit)

WDT (watchdog timer in SCU-DM)

- f_{SYS}: System clock

CGU (clock generation unit)

- f_{SYS}: System clock
- f_{PCLK}: Peripheral clock
- f_{MI_CLK}: Measurement interface clock
- f_{TFILT_CLK}: Analog module filter clock
- LP_CLK Clock source for all PMU submodules and WDT1

ICU (interrupt control unit)

- NMI (non-maskable interrupt)
- INTISR<15,13:4,1,0> External interrupt signals

RCU (reset control unit)

- PMU_1V5DidPOR Undervoltage reset of power-down supply
- PMU_PIN Reset generated by reset pin
- PMU_ExtWDT WDT1 reset
- PMU_IntWDT WDT (SCU) reset
- PMU_SOFT Software reset
- PMU_Wake Sleep mode/Stop mode exit with reset
- RESET_TYPE_3 Peripheral reset (contains all resets)
- RESET_TYPE_4 Peripheral reset (without SOFT and WDT reset)

Port control

- P0_POCONy.PDMx Driver strength control
- P1_POCONy.PDMx Driver strength control

Miscellaneous control

- MODPISELx mode selection registers for UART (source section) and timer (trigger or count selection)

6.2.2 SCU register overview

This chapter contains an overview of all SCU Registers.

6.2.2.1 Register map

Table 14 shows the SCU module base and end addresses.

Table 15 lists the addresses of the SCU SFRs.

Table 14 Register Address Space

Module	Base Address	End Address	Note
SCU	50005000_H	$50005FFF_H$	

Table 15 Register Overview SCU Module

Register Short Name	Register Long Name	Offset Address	Page Number
IRCON0	Interrupt Request Register 0	004_H	365
IRCON1	Interrupt Request Register 1	008_H	367
IRCON2	Interrupt Request Register 2	$00C_H$	369
IRCON3	Interrupt Request Register 3	010_H	371
IRCON4	Interrupt Request Register 4	014_H	372
NMISR	NMI Status Register	018_H	379
IEN0	Interrupt Enable Register 0	$01C_H$	362
NMICON	NMI Control Register	024_H	363
EXICON0	External Interrupt Control Register 0	028_H	364
IRCON0CLR	Interrupt Control 0 Clear Register	$02C_H$	366
MODIEN1	Peripheral Interrupt Enable Register 1	030_H	137
MODIEN2	Peripheral Interrupt Enable Register 2	034_H	138
MODIEN3	Peripheral Interrupt Enable Register 3	038_H	139
MODIEN4	Peripheral Interrupt Enable Register 4	$03C_H$	139
PMCON0	Power Mode Control Register 0	040_H	133
PLL_CON	PLL Control Register	044_H	110
CMCON1	Clock Control Register 1	048_H	112
CMCON2	Clock Control Register 2	$04C_H$	114
WDTCON	Watchdog Timer Control Register	050_H	166
APCLK_CTRL1	Analog Peripheral Clock Control 1 Register	054_H	116
APCLK1	Analog Peripheral Clock Register 1	058_H	119
APCLK2	Analog Peripheral Clock Register 2	$05C_H$	120
PMCON1	Peripheral Management Control Register 1	060_H	159
PMCON2	Peripheral Management Control Register 2	064_H	160
RSTCON	Reset Control Register	068_H	127
APCLK_CTRL2	Analog Peripheral Clock Control 2 Register	$06C_H$	118
SYSCON0	System Control Register 0	070_H	115
WDTREL	Watchdog Timer Reload Register	078_H	166
WDTWINB	Watchdog Window-Boundary Count	$07C_H$	168
WDTL	Watchdog Timer, Low Byte	080_H	167
WDTH	Watchdog Timer, High Byte	084_H	167
ID	Identity Register	$0A8_H$	177
PASSWD	Password Register	$0AC_H$	172
OSC_CON	OSC Control Register	$0B0_H$	108
COCON	Clock Output Control Register	$0B4_H$	122
MODPISEL	Peripheral Input Select Register	$0B8_H$	147
MODPISEL1	Peripheral Input Select Register 1	$0BC_H$	148

Table 15 Register Overview SCU Module

Register Short Name	Register Long Name	Offset Address	Page Number
MODPISEL2	Peripheral Input Select Register 2	$0C0_H$	149
MODPISEL3	Peripheral Input Select Register 3	$0C4_H$	150
MODSUSP1	Module Suspend Control Register 1	$0C8_H$	161
MODSUSP2	Module Suspend Control Register 2	$0CC_H$	162
GPT12PISEL	GPT12 Peripheral Input Select Register	$0D0_H$	158
EDCCON	Error Detection and Correction Control Register	$0D4_H$	169
EDCSTAT	Error Detection and Correction Status Register	$0D8_H$	170
MEMSTAT	Memory Status Register	$0DC_H$	178
MEMSTAT	Memory Status Register	$0DC_H$	180
NVM_PROT_STS	NVM Protection Status Register	$0E0_H$	175
MEM_ACC_STS	Memory Access Status Register	$0E4_H$	176
P0_POCON0	Port Output Control Register	$0E8_H$	151
P0_POCON1	Port Output Control Register	$0EC_H$	152
P0_POCON2	Port Output Control Register	$0F0_H$	153
TCCR	Temperature Compensation Control Register	$0F4_H$	157
P1_POCON0	Port Output Control Register	$0F8_H$	154
P1_POCON1	Port Output Control Register	$0FC_H$	155
P1_POCON2	Port Output Control Register	100_H	156
DMAIEN1	DMA Interrupt Enable Register 1	144_H	142
DMAIEN2	DMA Interrupt Enable Register 2	148_H	143
DMASRCSEL	DMA Source Selection Register	$14C_H$	144
DMAIRC1	DMA Interrupt Control Register 1	154_H	375
DMAIRC2	DMA Interrupt Control Register 2	158_H	377
GPT12IEN	GPT12 Interrupt Enable Register	$15C_H$	140
GPT12IRC	Timer and Counter Control/Status Register	160_H	373
IRCON1CLR	Interrupt Request 1 Clear Register	178_H	368
IRCON2CLR	Interrupt Request 2 Clear Register	$17C_H$	370
DMASRCSEL2	DMA Source Selection Register 2	180_H	145
DMAIRC1CLR	DMA interrupt Control 1 Clear Register	184_H	376
DMAIRC2CLR	DMA Interrupt Control 2 Clear Register	188_H	378
IRCON3CLR	Interrupt Request 3 Clear Register	190_H	371
IRCON4CLR	Interrupt Request 4 Clear Register	194_H	372
DMASRCCLR	DMA Source Selection Clear Register	198_H	145

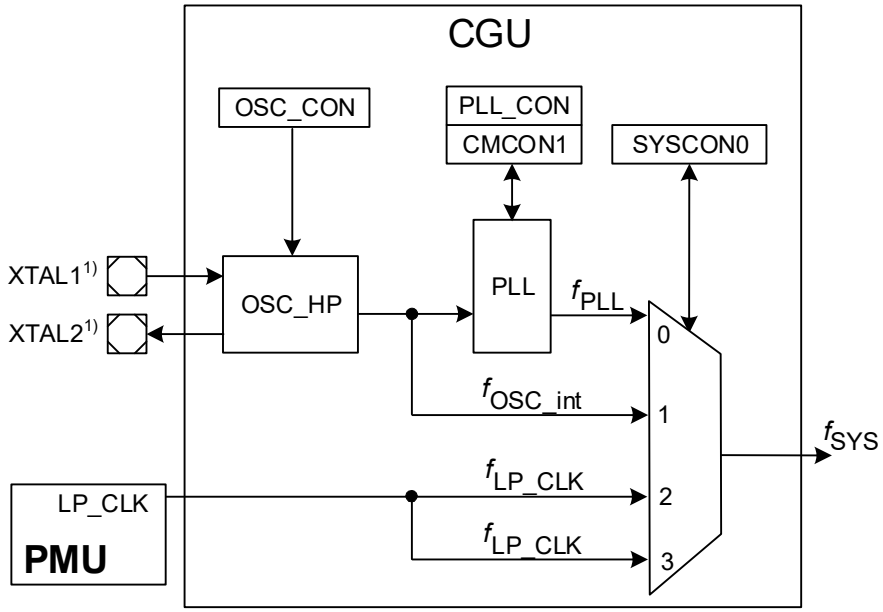
6.3 Clock generation unit

The clock generation unit (CGU) enables a flexible clock generation for the TLE987x. During user program execution, the frequency can be modified to optimize the performance/power consumption ratio, allowing power consumption to be adapted to the actual application state.

The CGU in the TLE987x consists of one oscillator circuit (OSC_HP), a phase-locked loop (PLL) module with an internal oscillator (OSC_PLL), and a clock control unit (CCU). The CGU can convert a low-frequency input/external clock signal to a high-frequency internal clock.

The system clock f_{SYS} is generated from one of the following selectable clocks:

- PLL clock output f_{PLL}
- Direct clock from oscillator OSC_HP f_{OSC}
- Low-precision clock f_{LP_CLK} (hardware-enabled for startup after reset and during power-down wake-up sequence)



1) Not available in all products. See the datasheet for details.

Figure 22 Clock generation unit block diagram

The following sections describe the different parts of the CGU.

6.3.1 Low-precision clock

The clock source LP_CLK is a low-precision RC oscillator (LP-OSC) with a nominal frequency of 18 MHz that is enabled by hardware as an independent clock source for the TLE987x startup after reset and during the power-down wake-up sequence. f_{LP_CLK} is not user-configurable.

6.3.2 High-precision oscillator circuit (OSC_HP)

Note: XTAL support is not available in all products of the TLE987x family. See the datasheet for details.

The high-precision oscillator circuit, designed to work with either an external crystal oscillator or an external stable clock source, consists of an inverting amplifier with XTAL1 as the input and XTAL2 as the output.

Figure 23 shows the recommended external circuitry for both operating modes, External Crystal mode and External Input Clock mode.

6.3.2.1 External Input Clock mode

When the clock signal is supplied directly – that is, not using an external crystal and bypassing the oscillator – the input frequency needs to be equal to or greater than 4 MHz if the PLL VCO part is used.

When an external clock signal is used, it must be connected to XTAL1. XTAL2 is left open (unconnected).

6.3.2.2 External Crystal mode

When using an external crystal, its frequency can be within the range of 4 MHz to 25 MHz. An external oscillator load circuitry must be used, connected to the XTAL1 and XTAL2 pins. It normally consists of two load capacitances, C1 and C2. A series damping resistor could be required for some crystals. The exact values and the corresponding operating ranges depend on the crystal and have to be determined and optimized in cooperation with the crystal vendor using the negative-resistance method. The load cap values listed in **Table 16** can be used as starting points for the evaluation:

Table 16 External CAP capacitors

Fundamental mode crystal frequency (approx., MHz)	Load caps C_1, C_2 (pF)
4	33
8	18
12	12
16	10
20	10
25	8

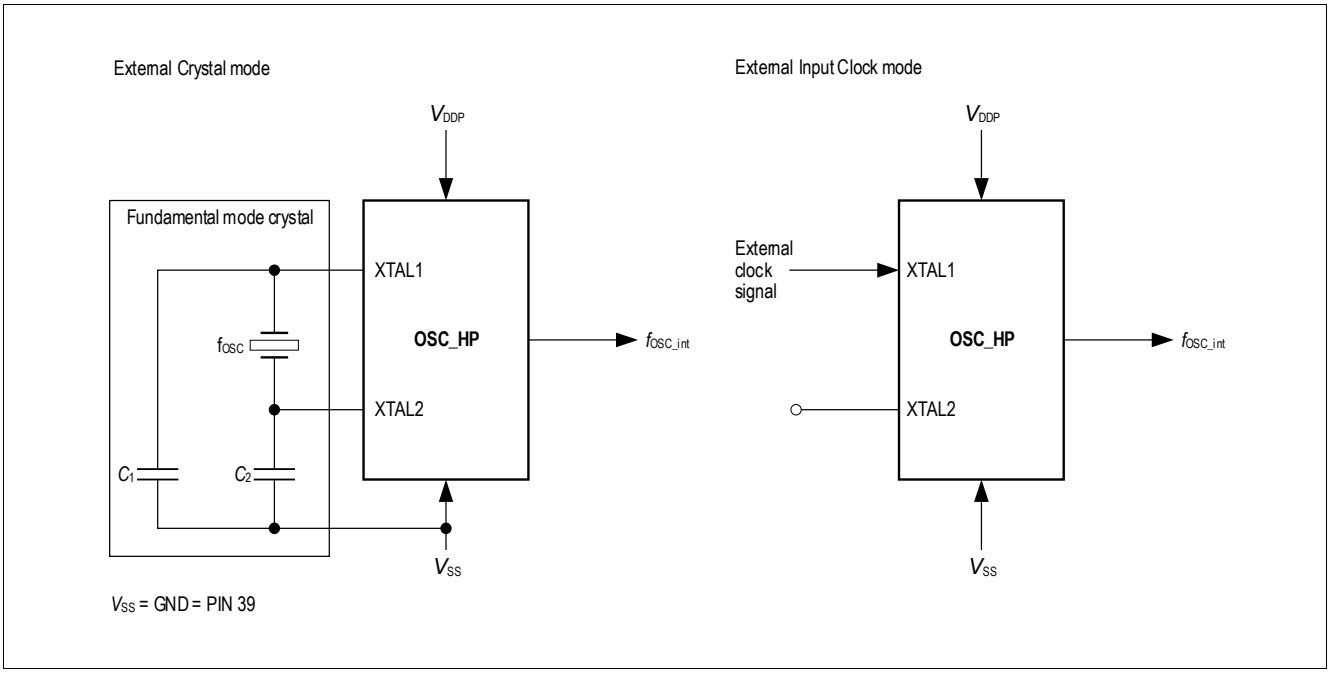


Figure 23 TLE987x external circuitry for the high-precision oscillator (XTAL is not available for all products)

6.3.3 Phase-locked loop (PLL) module

This section describes the TLE987x PLL module.

The f_{PLL} clock is generated in one of the following PLL configured modes:

- Prescaler mode, also called VCO Bypass mode
- Normal mode
- Freerunning mode

6.3.3.1 Features

The PLL module provides these features and functions:

- Programmable clock generation PLL
- Loop filter
- Input frequency: f_{OSC} (see datasheet for values)
- VCO frequency: $f_{VCO} = f_{VCO-0}$, or f_{VCO-1} (selectable by range **CMCON1**.VCOSEL, see datasheet for values)
- VCO lock detection
- Oscillator run detection
- Output frequency: f_{PLL} (see datasheet for values)
- Provided: Fixed input divider **P** = 1
- Provided: 4-bit feedback divider **N**
- Provided: 2-bit output divider **K2** and 1-bit output divider **K1**
- Oscillator watchdog
- Prescaler mode
- Freerunning mode
- Normal mode
- Sleep mode, also automatically activated in the device's power-saving mode
- Glitchless switching between both K-dividers
- Glitchless switching between Normal mode and Prescaler mode
- Internal oscillator for oscillator watchdog
- Internal oscillator as clock source – CGU oscillator, typically 5 MHz

6.3.3.2 PLL functional description

The following figure shows the PLL block structure.

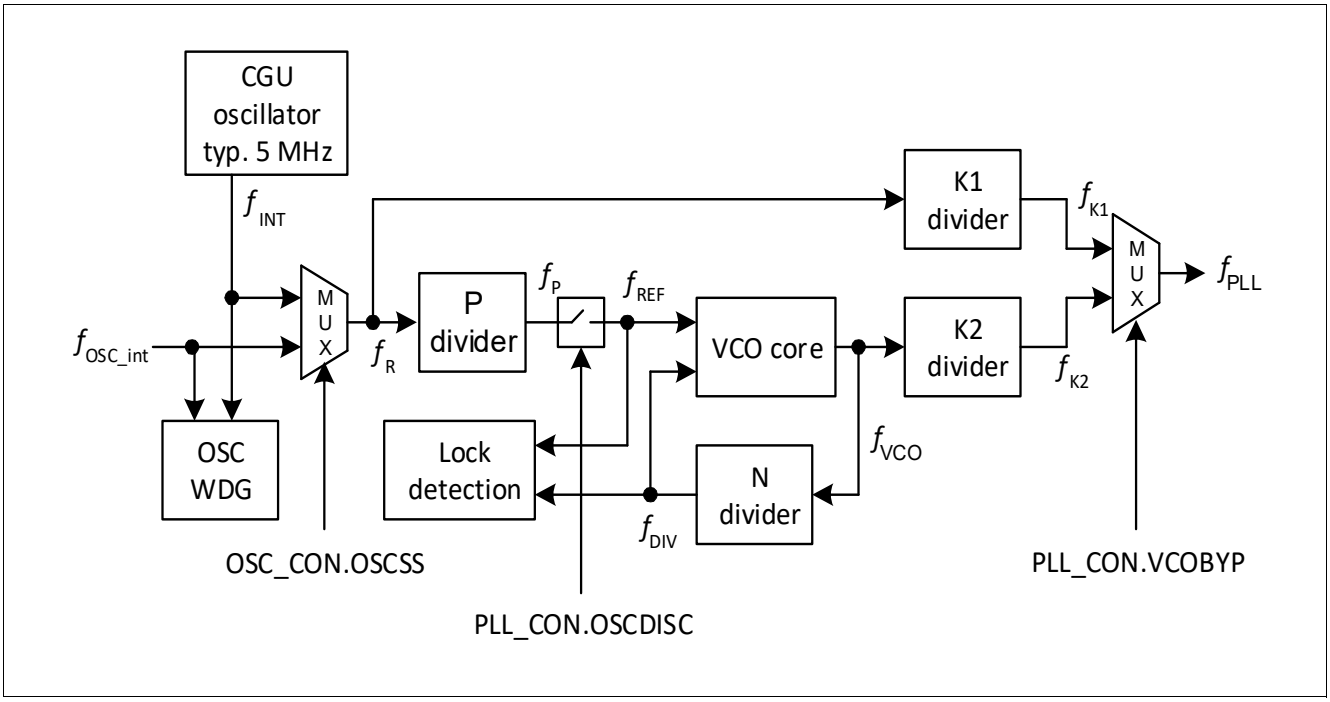


Figure 24 PLL block diagram for products with XTAL support

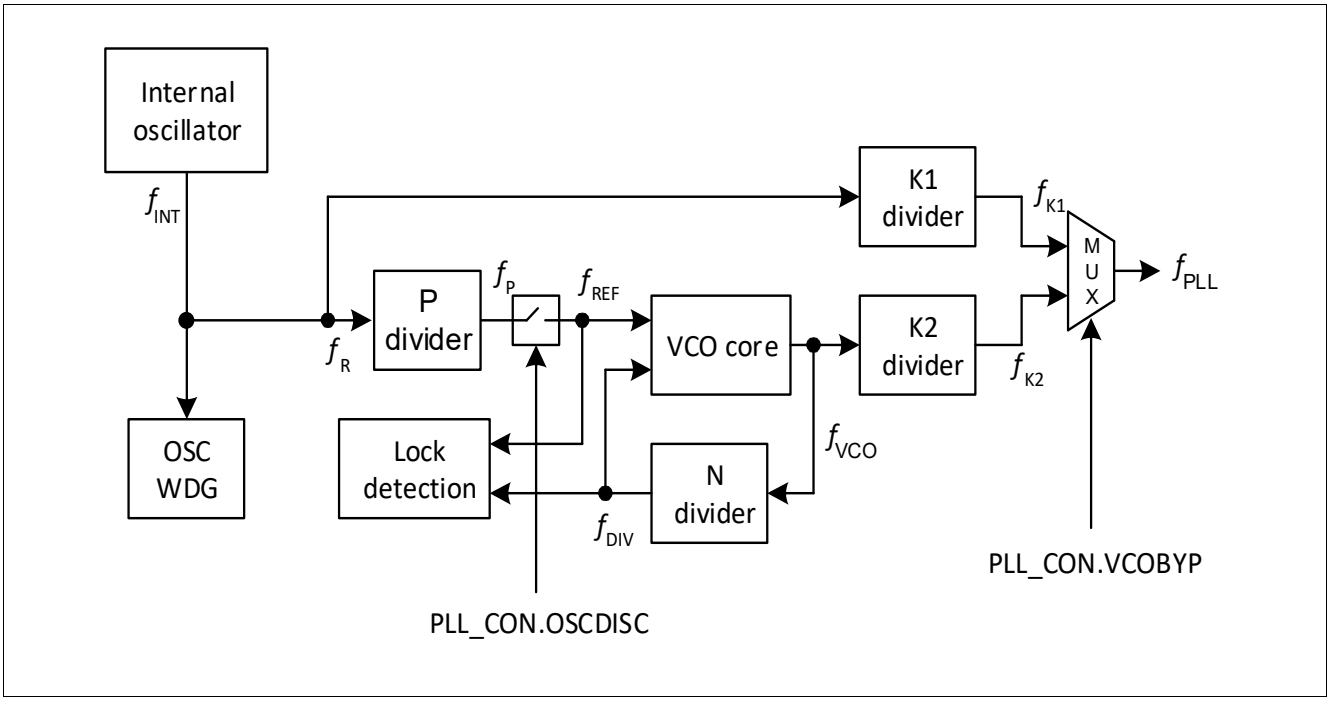


Figure 25 PLL block diagram for products without XTAL support

The reference frequency f_R can be selected to be taken either from the internal oscillator f_{INT} or from an external clock source f_{OSC_int}.

The PLL uses up to three dividers to manipulate the reference frequency in a configurable way. Each of the three dividers can be bypassed according to the PLL operating mode (based on f_{PLL}):

- Bypassing the P, N, and K2 dividers defines the Prescaler mode
- Bypassing the K1 divider defines the Normal mode
- Bypassing the K1 divider and ignoring the P divider defines the Freerunning mode

Table 17 shows the selectable clock source options.

Table 17 Clock option selection

VCOBYP	OSCDISC	Mode selected
0	0	Normal mode
1	0	Prescaler mode
0	1	Freerunning mode
1	1	Freerunning mode (OSCDISC has higher priority, VCOBYP will be set back to 0 by hardware)

Normal mode

In Normal mode, the reference frequency f_R is divided by a factor P, multiplied by a factor N, and then divided by a factor K2.

The output frequency is calculated by:

$$f_{PLL} = \frac{N}{P \times K2} \times f_R \tag{6.1}$$

The Normal mode is selected by the following settings:

- PLL_CON.VCOBYP = 0
- PLL_CON.OSCDISC = 0

The Normal mode is active when:

- PLL_CON.VCOBYP = 0
- PLL_CON.OSCDISC = 0
- PLL_CON.LOCK = 1

If f_{PLL} is selected as the clock source for the system frequency f_{SYS}, you should enable PLL in the Normal mode by default.

Note: The result of f_R times N must be within the VCO range. The VCO range selection has to be adjusted accordingly.

Prescaler mode (VCO Bypass mode)

In Prescaler mode, the reference frequency f_R is only divided by a factor K1.

The output frequency is calculated by:

$$f_{PLL} = \frac{f_R}{K1} \tag{6.2}$$

The Prescaler mode is selected by the following settings:

- PLL_CON.VCOBYP = 1
- PLL_CON.OSCDISC = 0

The Prescaler mode is active when:

- PLL_CON.VCOBYP = 1
- PLL_CON.OSCDISC = 0
- OSC_CON.OSC2L = 0 if f_{OSC_int} is provided as $f_{R\ (OSC_CON.OSCSS = 01B)}$

Freerunning mode

In Freerunning mode, the base frequency output of the voltage-controlled oscillator (VCO) $f_{VCObase}$ is only divided by a factor K2.

The output frequency is calculated by:

$$f_{PLL} = \frac{f_{VCObase}}{K2} \tag{6.3}$$

The Freerunning mode is enabled by any of the following settings and conditions:

- PLL_CON.VCOBYP = 0 and PLL_CON.LOCK = 0
- PLL_CON.VCOBYP = 1 and OSC_CON.OSCSS = 1 and OSC_CON.OSC2L = 1
- PLL_CON.VCOBYP = 0 and PLL_CON.OSCDISC = 1

The Freerunning mode is active when:

- PLL_CON.VCOBYP = 0
- PLL_CON.OSCDISC = 1
- PLL_CON.LOCK = 0

General configuration overview

The divider values and all necessary other values can be configured via the PLL configuration registers.

In TLE987x, the P factor is fixed to 1. **Table 18** gives the valid output frequency range for the P divider dependent on f_R frequency range:

Table 18 P-divider factor = 1

P	f_P for f_R =				
	4 MHz	**5 MHz**	**10 MHz**	**16 MHz**	**25 MHz**
1	4	5	10	16	Not supported

Note: The entire range between two f_R columns in the above table is also supported. For example, for a range f_R = 10 to 16, f_P = 10 to 16 MHz.

The P-divider output frequency f_P is fed to the voltage-controlled oscillator (VCO). The VCO is a part of PLL with a feedback path. A divider in the feedback path (N divider) divides the VCO frequency. The f_{VCO} range is defined by the configuration of VCOSEL.

Table 19 VCO range

VCOSEL	f_{VCOmin}	f_{VCOmax}	$f_{VCObase}$ [1)]	Unit
0	48	112	Approx. max. 38	MHz
1	96	160	Approx. max. 76	MHz

1) $f_{VCObase}$ is the free-running operation frequency of the PLLVCO, when no input reference clock is available.

The following table shows the possible N-loop division rates and provides the valid output frequency range for f_{REF} depending on N and the VCO frequency range:

Table 20 N-loop division rates

N	f_{DIV} for f_{VCO} =					
	48	**72**	**96**	**112**	**136**	**160**
8	6.00	9.00	12.00	14.00	Not supported[1)]	
9	$5.3\overline{3}$	8.00	$10.6\overline{6}$	$12.4\overline{4}$	$15.1\overline{1}$	Not supported[1)]
10	4.80	7.20	9.60	11.20	13.60	16.00
11	$4.\overline{36}$	$6.\overline{54}$	$8.\overline{72}$	$10.\overline{18}$	$12.3\overline{6}$	$14.5\overline{4}$
12	4.00	6.00	8.00	$9.3\overline{3}$	11.33	$13.3\overline{3}$
13	Not supported[1)]	5.54	7.38	8.62	10.46	12.31
14 …17		…	…	…	…	…
18		4.00	$5.3\overline{3}$	$6.2\overline{2}$	$7.5\overline{5}$	$8.8\overline{8}$
19	Not supported[1)]		5.05	5.89	7.16	8.42
20 …23			…	…	…	…
24			4.00	$4.6\overline{6}$	$5.6\overline{6}$	$6.6\overline{6}$
25	Not supported[1)]			4.48	5.44	6.40
26 …27				…	…	…
28				4.00	4.86	5.71

1) Values in this range are supported only in Freerunning mode, but have no effect there.

Note: The entire range between two f_{VCO} columns in the table above is supported.

The N-divider output frequency f_{DIV} is then compared with f_{REF} in the phase detector logic, part of the VCO logic. The phase detector determines the difference between the two clock signals and modifies the output frequency of the VCO, f_{VCO}, accordingly.

Note: Although the N divider is used to divide a frequency, it is sometimes also called a multiplier. This is caused by the fact that the f_{VCO} frequency is a multiple of the f_{DIV} frequency (produced by applying the N divider in the feedback path).

The output frequency of the VCO, f_{VCO}, is divided by K2 to provide the final output frequency f_{PLL}. **Table 21** shows the output frequency range depending on the K2 divisor and the VCO frequency range:

Table 21 K2 divisor table

K2	f_{PLL} for f_{VCO} =						Duty cycle [%]
	48	**72**	**96**	**112**	**136**	**160**	
2	24.0	36.0	48.0	56.0	68.0	80.0	50
3	16.0	24.0	32.0	$37.\overline{3}$	$45.\overline{3}$	$53.\overline{3}$	46 to 54
4	12.0	18.0	24.0	28.0	34.0	40.0	50
5	9.6	14.4	19.2	22.4	27.2	32.0	48.5 to 51.5

Notes

1. *The entire range between two f_{vco} columns in the table above is supported.*
2. *If the clock's duty cycle deviates from 50%, you have to ensure that not only the cycle time remains valid but also the minimum clock pulse width. This issue becomes more important with higher frequencies and larger deviations from a duty cycle of 50%.*

Table 22 shows the output frequency range depending on the K1 divisor and f_R.

Table 22 K1 divisor table

K1	f_{PLL} for f_R =			Duty cycle [%]
	5	8	16	
1	5.0	8.0	16.0	40 to 60
2	2.5	4.0	8.0	50

For different source oscillators, the values required to select f_{PLL} = 24 MHz or 40 MHz are shown in **Table 23**.

Table 23 System frequency

f_{PLL} selected	Oscillator	f_{OSC}	N	P	K	Actual f_{SYS}
40 MHz	On-chip	5 MHz	16	1	2	40 MHz
	External	10 MHz	8	1	2	40 MHz
		8 MHz	10	1	2	40 MHz
24 MHz	On-chip	5 MHz	24	1	5	24 MHz
	External	12 MHz	8	1	4	24 MHz
		8 MHz	12	1	4	24 MHz
		6 MHz	12	1	3	24 MHz

For the TLE987x, the value of P is fixed to 1. In order to obtain the required f_{PLL}, the values of VCOSEL, N, and K can be chosen, respectively, by setting the VCOSEL, NDIV, and KDIV (either K2DIV or K1DIV) bits for different oscillator input frequencies.

6.3.3.3 Oscillator watchdog

The oscillator watchdog monitors the external incoming clock f_{OSC_int}. Only incoming frequencies that are too low to enable a stable operation of the VCO circuit (below 300 kHz) are detected.

The internal oscillator (OSC_PLL) frequency f_{INT} is used as the reference clock and therefore must be in operation.

By setting the OSC_CON.OSCWDTRST bit, detection can be restarted without resetting the complete PLL. The detection status output is only valid after several cycles of f_{INT}.

6.3.3.4 PLL VCO lock detection

The PLL has a lock detection that supervises the VCO part of the PLL in order to differentiate between stable and instable VCO circuit behavior. The lock detector marks the VCO circuit and therefore the output f_{VCO} of the VCO as instable if the two inputs f_{REF} and f_{DIV} differ too much. Changes in one or both input frequencies below a certain level are not marked as a loss of lock because the VCO can handle such small changes without any problems for the system. **Table 24** shows values below which lock is not lost for different input values.

Table 24 Loss of VCO lock definition

Maximum tolerated change		
$\frac{df_{DIV}}{dt}$ for f_{REF} =		(6.4)
4 MHz	**10 MHz**	**16 MHz**
≤ 0.6 kHz/µs	≤ 3.7 kHz/µs	≤ 9.5 kHz/µs
20 MHz	**25 MHz**	**40 MHz**
≤ 14.9 kHz/µs	≤ 23.2 kHz/µs	≤ 59.5 kHz/µs

6.3.3.5 Internal oscillator (OSC_PLL)

The PLL's internal oscillator is used for two different purposes:

- Operating the oscillator watchdog
  - The frequency of the direct PLL input from OSC_HP (XTAL) is supervised using the OSC_PLL as the reference frequency. For more information, see **Section 6.3.3.3**.
- Providing an input clock to the PLL
  - The OSC_PLL can be used as an input clock for all PLL modes. This is controlled and configured via OSC_CON.OSCSS.
  - OSC_PLL operates at a nominal frequency of typ. 5 MHz.

6.3.3.6 Switching PLL parameters

Follow this procedure to change PLL parameters via the PLL_CON register:

1. Disable the PLL loss-of-lock NMI, because the PLL unlocks while it is being configured.
2. If you want to use the Prescaler mode, set **CMCON1**.K1DIV = 1 (div 1) to set f_{PLL} to 5 MHz.
3. If you want to use the Prescaler mode, switch to it by setting **PLL_CON**.VCOBYP = 1[1)].
4. Set the oscillator trimming flag **OSC_CON**.OSCTRIM_8 = 1.
5. Set the reserved bit 5 in **OSC_CON**.Bit5 = 1.
6. Set **PLL_CON**.NDIV to the desired value.
7. Set **CMCON1**.VCOSEL to the appropriate range, as described in **Table 19**. **Table 20** lists the f_{VCO} values for the various NDIV values. Set **CMCON1**.K2DIV to the desired value. These two operations can be performed together.
8. Restart the PLL lock detection by writing **PLL_CON**.RESLD = 1.
9. Wait until the PLL locks. It is locked if **PLL_CON**.LOCK = 1.
10. Switch to Normal mode by writing **PLL_CON**.VCOBYP = 0.
11. Enable the PLL loss-of-lock NMI if desired.

1) This operation sets PLL_CON.OSCDISC to 0 through hardware.

6.3.3.7 Oscillator watchdog event or PLL loss-of-lock detection

If a too-low frequency of the external clock source f_{OSC_int} is detected, the OSC-Too-Low flag (OSC_CON.OSC2L) is set. A corresponding trap request to the CPU is activated only if it is enabled by NMICON.NMIOWD and at least one of these conditions is true:

1. The PLL is in Prescaler mode, OSCSS = 01 (selecting f_{OSC_int} as PLL input clock source), and SYSCON0.SYSCLKSEL selects the PLL clock output as the system frequency.
2. SYSCON0.SYSCLKSEL selects f_{OSC_int} as the system frequency.

If either of these two cases is true and the OSC2L condition is set, the OWD NMI flag FNMIOWD in NMISR is set.

Note: Do not restart the oscillator watchdog detection by setting the OSC_CON.OSCWDTRST bit while PLL is in Prescaler mode, as the detection status (OSC_CON.OSC2L) takes some time to become stable.

An oscillator watchdog event is normally followed by a PLL loss-of-lock detection.

If PLL is not the system clock source (SYSCON0.SYSCLKSEL deselects PLL, or PLL is in Prescaler mode) when the loss of lock is detected, only the lock flag is reset (PLL_CON.LOCK = 0). No loss-of-lock NMI is generated and no further action is taken. Otherwise if PLL is selected as the clock source for system frequency and VCOBYP = 0, the PLL loss-of-lock NMI flag FNMIPLL in NMISR is set. If enabled by NMICON.NMIPLL, an NMI trap request to the CPU is activated. In addition, the lock flag is reset.

Note: A lock-of-loss NMI request can only be generated if the lock flag is set in the first place. This avoids a potential PLL loss-of-lock NMI request after a power-on reset.

When an oscillator watchdog event occurs (when PLL is in Prescaler mode and the external clock – OSC_HP – is selected as the PLL clock input) or when a PLL loss of lock is detected (when PLL is in Normal mode), the PLL is switched to run in the Freerunning mode on the VCO base frequency divided by K2, which is enforced by hardware until the Prescaler mode is selected.

Due to the above, the PLL only runs in Prescaler mode when the PLL configuration is changed or it switches between PLL operation modes.

6.3.3.8 Oscillator watchdog event or loss-of-lock recovery

In case of oscillator watchdog NMI, user software can first check if the PLL remains locked. If not, the clock system can be configured by executing the following sequence as the OWD NMI routine:

1. Restart the oscillator watchdog detection by setting bit OSC_CON.OSCWDTRST
2. Wait until OSC_CON.OSC2L has been cleared.
3. When the OSC_CON.OSC2L bit has been cleared:
   a) Select the Prescaler mode (PLL_CON.VCOBYP = 1).
   b) Set the restart lock detection bit PLL_CON.RESLD = 1.
   c) Wait until the PLL VCO part becomes locked (PLL_CON.LOCK = 1).
   d) When the lock is set again, deselect the Prescaler mode (PLL_CON.VCOBYP = 0). Normal PLL operation resumes.
4. Clear the OWD NMI flag FNMIOWD.

In the general case of PLL loss-of-lock or to reconfigure the PLL settings, user software can try to configure the clock system again by executing the following sequence:

1. If the input clock source is XTAL (f_{OSC_int} from OSC_HP), ensure that the input frequency is above its threshold by checking OSC_CON.OSC2L.
2. Select the Prescaler mode (PLL_CON.VCOBYP = 1)

3. If desired, configure the PLL divider settings.
4. Set the restart lock detection bit PLL_CON.RESLD = 1
5. Wait until the PLL VCO part becomes locked (PLL_CON.LOCK = 1)
6. When the lock is set again, deselect the Prescaler mode (PLL_CON.VCOBYP = 0). Normal PLL operation resumes.
7. Clear the PLL loss-of-lock NMI flag FNMIPLL.

6.3.4 Clock control unit

The clock control unit (CCU) receives the PLL clock f_{PLL}, or the external input clock f_{OSC_int}, or the low-precision input clock f_{LP_CLK}. The system frequency is derived from one of these clock sources.

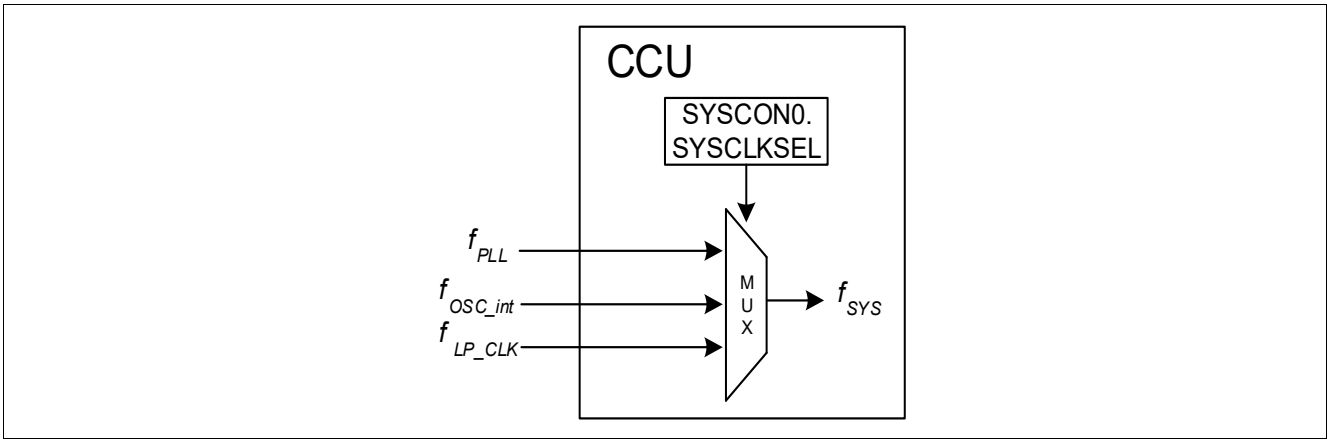


Figure 26 Clock inputs to the clock control unit

The CCU generates all necessary clock signals for the microcontroller from the system clock. It consists of:

- Clock slow-down circuitry
- Central circuit to enable or disable clock control

In normal operation the main module frequencies (synchronous unless otherwise stated) are as follows:

- System frequency, f_{SYS}: Configurable up to 40 MHz (the measurement interface clock MI_CLK is derived from this clock)
- CPU clock (CCLK, SCLK): Configurable up to 40 MHz (created by dividing the NVM access clock)
- NVM access clock (NVMACCCLK): Configurable up to 40 MHz
- Peripheral clock (PCLK, PCLK2, NVMCLK): Configurable up to 40 MHz (equals CPU clock; must be same or higher)

Some peripherals are clocked by PCLK, others by PCLK2, and the NVM is clocked by both NVMCLK and NVMACCCLK. During normal running mode, PCLK = PCLK2 = NVMCLK = CCLK. After a wake-up from power-down, PCLK2 and NVMCLK are restored. PCLK is restored only after PLL has been locked.

For optimized NVM read and write access with reduced wait states and adapted to the CPU‘s operational frequency, the NVMCLKFAC bit field is provided for setting the multiplier between the NVM access clock NVMACCCLK and the CPU clock CCLK. For details, refer to the separate NVM documentation.

In the slow-down mode, the slow-down circuitry reduces the operating frequency using the clock divider setting from the CLKREL bit field. The settings in the CLKREL bit field only have an effect when the slow-down mode is enabled via the SFR bit PMCON0.SD bit. Note that the slow-down setting of the CLKREL bit field correspondingly reduces the NVMACCCLK frequency. Slow-down settings do not influence the erase and write cycles for the NVM.

6.3.4.1 Clock tree

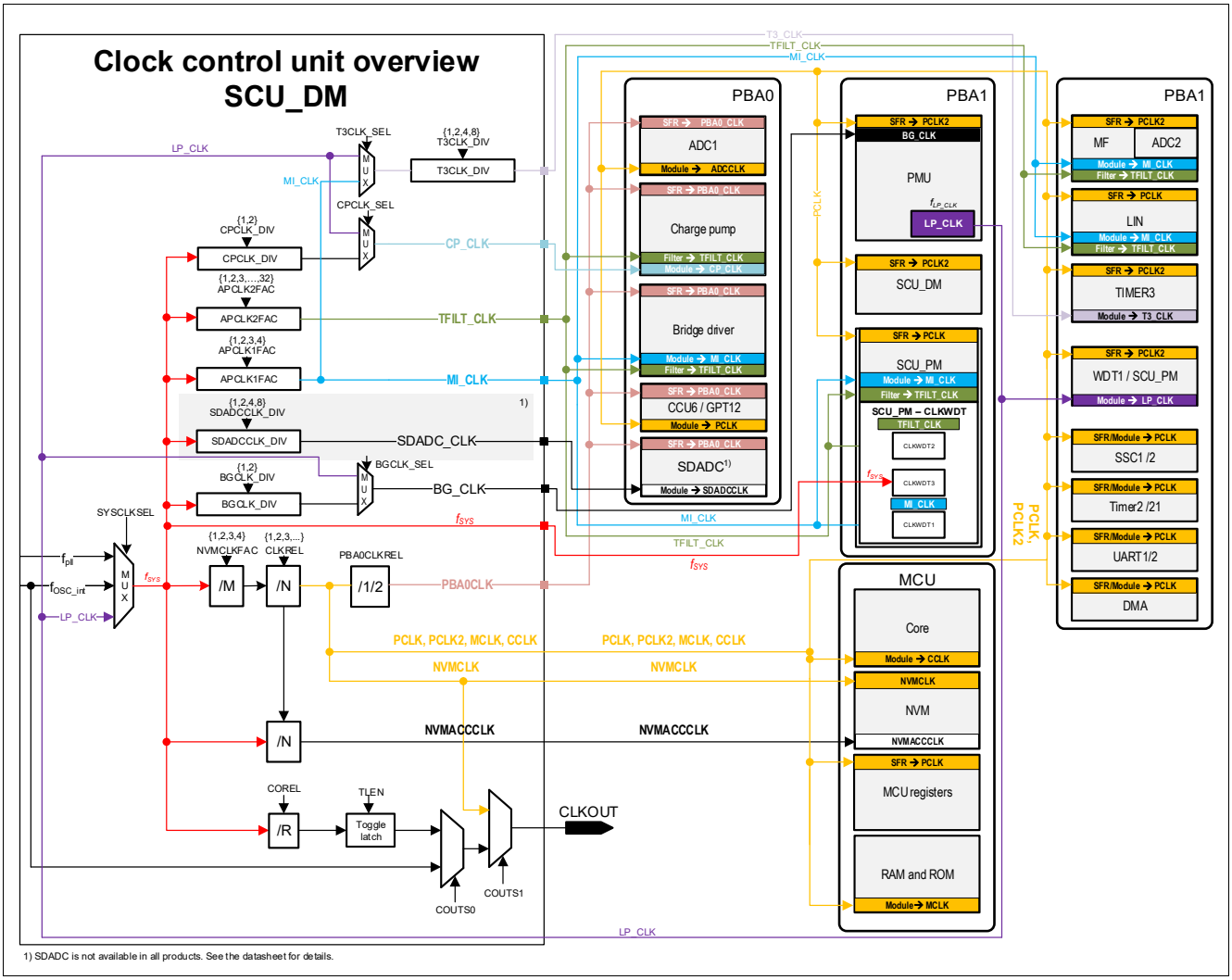


Figure 27 Clock tree

6.3.4.2 Startup control for system clock

Typically when the TLE987x starts up after reset, the LP_CLK is selected by hardware to provide the system frequency f_{SYS}. The CPU runs based on this system frequency during the startup operation (unless otherwise specified and configured by firmware). Meanwhile, the system clock input is switched to the PLL output. The default user boot configuration uses the internal oscillator (5 MHz) as the input. User code can modify the default PLL configuration as required.

The above does not apply to resets that do not reset the clock system, which are the watchdog timer (WDT) reset and the soft reset. With these resets, the previous user configuration of PLL and clock system is retained across the reset.

Note: If the PLL fails to lock during startup operation, the LP_CLK continues to provide the system clock input. The system clock input source is indicated by the SYSCON0.SYSCLKSEL bit field.

6.3.5 External clock output

An external clock output is provided as CLKOUT. This output clock can be enabled and disabled via the COCON.EN bit. One of three clock sources (f_{CCLK} or f_{SYS}/n or f_{OSC_int}) can be selected for the output, configured via the COCON.COUTS1 and COUTS0 bit fields.

If COUTS1 = 0, the output clock is f_{CCLK} (irrespective of the COUTS0 setting). Otherwise, if COUTS0 = 0, the output clock is from oscillator output frequency; if COUTS0 = 1, the clock output frequency is chosen by the COREL bit field which selects the N divider factor on f_{SYS}. If this selection is active, the clock output frequency can further be divided by 2 using a toggle latch (TLEN = 1). The resulting output frequency has 50% duty cycle.

6.3.6 CGU registers

The registers of the clock generation unit for PLL and oscillator control are not affected by the watchdog timer (WDT) reset and soft reset. Therefore, the system clock configuration and frequency is maintained across these types of reset.

Unless otherwise stated, the reset values stated for the following registers apply only to power-on resets, brown-out resets, hard resets, WDT1 resets, and wake-up resets.

6.3.6.1 PLL oscillator register

These registers control the setting and trimming of OSC_PLL, the power-down of XTAL (OSC_HP), and the configuration and status monitor of the oscillator watchdog.

The OSC_CON register is initialized with a value of 98_H by the BootROM code during the boot process.

OSC_CON
OSC Control Register **($0B0_H$)** **Reset Value: 10_H**

7	6	5	4	3	2	1	0
OSCTRIM_8	**RES**	**RES**	**XPD**	**OSC2L**	**OSCWDTRST**	**OSCSS**	
rw	r	rw	rw	rh	rwh	rw	

Field	Bits	Type	Description
OSCTRIM_8	7	rw	**OSC_PLL Trim Configuration Bit [8]** This bit field enables the trimming of the OSC_PLL. You should always set this bit when you write to this register. This bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
RES	6	r	**Reserved** This bit field is always read as zero.
RES	5	rw	**Reserved** You should always set this bit to 1 when you write to this register. This bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
XPD	4	rw	**XTAL (OSC_HP) Power Down Control**[1)] 0_B XTAL (OSC_HP) is not powered down. 1_B XTAL (OSC_HP) is powered down. The XPD bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**. *Note: When XPD is set, switching the clock to the internal oscillator must be done asynchronously.*

Field	Bits	Type	Description
OSC2L	3	rh	**OSC-Too-Low Condition Flag** The oscillator watchdog monitors $f_{OSC_int.}$ 0_B f_{OSC_int} is above threshold. 1_B f_{OSC_int} is below threshold. On OSC-too-low detection (OSC2L: 0 →1) and VCOBYP = 1 and OSCSS = 01, PLL switches to freerunning mode. In that case, and when f_{OSC_int} is selected as the system clock source, the hardware switches the system clock source to PLL. (**SYSCON0**.SYSCLKSEL is also updated.) *Note: An OWD NMI request is activated by an OSC-too-low condition only in two cases:* *1) When VCOBYP = 1, OSCSS = 01, and SYSCLKSEL selects PLL clock as the system clock source.* *2) When SYSCLKSEL selects f_{OSC_int} as the system clock source.*
OSCWDTRST	2	rwh	**Oscillator Watchdog Reset** Setting this bit resets the OSC2L status flag to 1 and restarts the oscillator detection. This bit is then automatically reset to 0 and thus is always read as 0. 0_B No effect. 1_B Reset OSC2L flag and restart the oscillator watchdog of the PLL.
OSCSS	1:0	rw	**Oscillator Source Select** 00_B PLL internal oscillator OSC_PLL (f_{INT}) is selected synchronously as f_R. 01_B XTAL[1)] (f_{OSC} from OSC_HP) is selected synchronously as f_R. 10_B PLL internal oscillator OSC_PLL (f_{INT}) is selected asynchronously as f_R. 11_B PLL internal oscillator OSC_PLL (f_{INT}) is selected asynchronously as f_R. The OSCSS bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**. *Note: Synchronous switching the clock source to the internal oscillator is not possible when XPD = 1 or when no external clock is available (check bit OSC2L).* *Note: Use the 10_B or 11_B options only when the external clock is not available.*

1) XTAL support is not available in all products of the TLE987x family. See the datasheet for details.

This OSC_CON register is reset by RESET_TYPE_4.

6.3.6.2 PLL registers

These registers control the PLL settings.

The PLL_CON register is initialized with a value of 61_H by the BootROM code during the boot process.

PLL_CON
PLL Control Register **(044_H)** **Reset Value: 64_H**

7	6	5	4	3	2	1	0
NDIV				**VCOBYP**	**OSCDISC**	**RESLD**	**LOCK**
rw				rwh	rwh	rwh	r

Field	Bits	Type	Description
NDIV	7:4	rw	**PLL N-Divider** 0000_B N = 8 0001_B N = 9 0010_B N = 10 0011_B N = 12 0100_B N = 14 0101_B N = 15 0110_B N = 16 0111_B N = 18 1000_B N = 20 1001_B N = 21 1010_B N = 22 1011_B N = 24 1100_B N = 25 1101_B N = 26 1110_B N = 27 1111_B N = 28 The NDIV bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
VCOBYP	3	rwh	**PLL VCO Bypass Mode Select** 0_B Normal (or freerunning) operation (default) 1_B Prescaler mode: VCO is bypassed. (The PLL output clock is derived from the input clock divided by the K1 divider.) This bit is cleared by hardware when the PLL switches to freerunning operation. When the bit value changes from 0 to 1, the OSCDISC bit becomes 0.
OSCDISC	2	rwh	**Oscillator Disconnect** 0_B Oscillator is connected to the PLL 1_B Oscillator is disconnected from the PLL. By default after a power-on reset, the PLL is running in Freerunning mode (the oscillator is disconnected).

Field	Bits	Type	Description
RESLD	1	rwh	**Restart Lock Detection** Setting this bit resets the PLL lock status flag and restarts the lock detection. This bit is then automatically reset to 0 and thus is always read as 0. 0_B No effect. 1_B Reset lock flag and restart lock detection.
LOCK	0	r	**PLL Lock Status Flag** 0_B The frequency difference between f_{REF} and f_{DIV} is greater than supported. The VCO part of the PLL cannot lock on a target frequency. 1_B The frequency difference between f_{REF} and f_{DIV} is small enough to enable stable VCO operation. **Notes** *1. When the VCO lock is lost, the f_{VCO} goes to the upper boundary of the selected VCO band if the reference clock input is greater than expected.* *2. When the VCO lock is lost, the f_{VCO} goes to the lower boundary of the selected VCO band if the reference clock input is lower than expected.* *3. When the loss of lock is detected (LOCK: 1 →0) and when VCOBYP = 0, the PLL switches to Freerunning mode.* *4. A loss-of-lock NMI request is activated by a loss-of-lock detection only when VCOBYP = 0 and SYSCON0.SYSCLKSEL selects the PLL clock as the system frequency.*

The PLL_CON register is reset by RESET_TYPE_4.

CMCON1
Clock Control Register 1 **(048_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
VCOSEL	**K1DIV**	**K2DIV**		**CLKREL**			
rw	rw	rw		rw			

Field	Bits	Type	Description
VCOSEL	7	rw	**VCOSEL Setting** 0_B VCOSEL = 0 1_B VCOSEL = 1
K1DIV	6	rw	**PLL K1-Divider** 0_B K1 = 2 1_B K1 = 1 The K1DIV bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
K2DIV	5:4	rw	**PLL K2-Divider** 00_B K2 = 2 01_B K2 = 3 10_B K2 = 4 11_B K2 = 5 The K2DIV bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**. *Note: Depending on the VCOSEL setting, you must set the K2 divider large enough to ensure that the PLL output frequency in Freerunning mode is never higher than the maximum specified for the device.*

Field	Bits	Type	Description
CLKREL	3:0	rw	**Slow Down Clock Divider for f_{CCLK} Generation** 0000_B f_{sys} 0001_B $f_{sys}/2$ 0010_B $f_{sys}/3$ 0011_B $f_{sys}/4$ 0100_B $f_{sys}/8$ 0101_B $f_{sys}/16$ 0110_B $f_{sys}/24$ 0111_B $f_{sys}/32$ 1000_B $f_{sys}/48$ 1001_B $f_{sys}/64$ 1010_B $f_{sys}/96$ 1011_B $f_{sys}/128$ 1100_B $f_{sys}/192$ 1101_B $f_{sys}/256$ 1110_B $f_{sys}/384$ 1111_B $f_{sys}/512$ This setting has an effect only when the Slow-Down mode is enabled. *Note: f_{SYS} is further divided by the NVMCLKFAC factor to generate f_{CCLK}.*

The CMCON1 register is reset by RESET_TYPE_4.

CMCON2
Clock Control Register 2 **($04C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES							**PBA0CLKREL**
r							rw

Field	Bits	Type	Description
RES	7:1	r	**Reserved** This bit field is always read as zero.
PBA0CLKREL	0	rw	**PBA0 Clock Divider** This flag configures the PBA0 clock divider. 0_B divide by 1 1_B divide by 2 The PBA0CLKREL bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.

The CMCON2 register is reset by RESET_TYPE_4.

6.3.6.3 System clock control registers

The clock source for the system is selected via the SYSCON0 register.

The SYSCON0 register is initialized with a value of 00_H by the BootROM code during the boot process.

SYSCON0
System Control Register 0 **(070_H)** **Reset Value: $C0_H$**

7	6	5	4	3	2	1	0
SYSCLKSEL		NVMCLKFAC		RES	RES	RES	RES
rw		r		r	r	r	r

Field	Bits	Type	Description
SYSCLKSEL	7:6	rw	**System Clock Select** The clock source that is used as the system clock. 00_B The PLL clock output signal f_{PLL} is used 01_B The direct clock input from f_{OSC_int} is used[1)] 10_B The direct low-precision clock input from f_{LP_CLK} is used. 11_B The direct low-precision clock input from f_{LP_CLK} is used. *Note: In normal application, the system is expected to run on the PLL clock output.*
NVMCLKFAC	5:4	r	**NVM Access Clock Factor** The factor by which the system clock is divided to create the synchronous NVMACCCLK clock. 00_B Divide by 1 01_B Divide by 2 10_B Divide by 3 11_B Divide by 4 *Note: This value can only be changed via a dedicated BootROM routine.*
RES	3, 2, 1, 0	r	**Reserved** Returns 0 if read; should be written with 0.

1) Available only for products with XTAL support. See the datasheet for details.

Table 25 Reset values of the SYSCON0 register

Reset source	Reset value
Power-on reset, brown-out reset, WDT1 reset, wake-up reset, hardware reset	$C0_H$
Watchdog timer reset, soft reset	$U0_H$ (U = unchanged)

The SYSCON0 register is reset by RESET_TYPE_4.

6.3.6.4 Analog Peripherals Clock Control registers

The clock frequency for the analog modules is selected via the APCLK1 and APCLK2 registers. APCLK1 specifies the operating clock for all analog peripherals. For this reason, it is important to always adapt the Analog Peripherals Clock Control register settings to the required frequency range if the system clock is changed.

The APCLK_CTRL1 register is initialized with a value of 05_H by the BootROM code during the boot process.

APCLK_CTRL1
Analog Peripheral Clock Control 1 Register (054_H) Reset Value: 30_H

7	6	5	4	3	2	1	0
CPCLK_DIV	**CPCLK_SEL**	**BGCLK_DIV**	**BGCLK_SEL**	**CLKWDT_IE**	**T3CLK_SEL**	**APCLK_SET**	**PLL_LOCK**
rw	rw	rw	rw	rw	rw	rwh1	r

Field	Bits	Type	Description
CPCLK_DIV	7	rw	**Charge Pump Clock Divider** This flag configures the charge pump clock divider. 0_B Divide by 2 1_B Divide by 1 The CPCLK_DIV bit is a password-protected bit.
CPCLK_SEL	6	rw	**Charge Pump Clock Selection** This flag selects the charge pump clock. 0_B LP_CLK is selected 1_B f_{sys} is selected The CPCLK_SEL bit is a password-protected bit. *Note: If SYSCLKSEL[1] = 1, the CPCLK_SEL default value 0 (LP_CLK) is used.*
BGCLK_DIV	5	rw	**Bandgap Clock Divider** This flag configures the bandgap clock divider. 0_B Divide by 2 1_B Divide by 1 The BGCLK_DIV bit is a password-protected bit.
BGCLK_SEL	4	rw	**Bandgap Clock Selection** This flag selects the bandgap clock. 0_B LP_CLK is selected 1_B f_{sys} is selected The BGCLK_SEL bit is a password-protected bit. *Note: If SYSCLKSEL[1] = 1, the BGCLK_SEL default value 0 (LP_CLK) is used.*
CLKWDT_IE	3	rw	**Clock Watchdog Interrupt Enable** 0_B Interrupt disabled 1_B Interrupt enabled The CLKWDT_IE bit is a password-protected bit.

Field	Bits	Type	Description
T3CLK_SEL	2	rw	**Timer3 Clock Selection** This flag selects the Timer3 clock. 0_B LP_CLK is selected 1_B MI_CLK is selected The T3CLK_SEL bit is a password-protected bit.
APCLK_SET	1	rwh1	**Set and Use Clock Settings flag** This flag makes the SDADCCLK_DIV[1)], APCLK1, APCLK2, and T3CLK_DIV settings take effect. 0_B Clock settings are ignored (previous values are used) 1_B Clock settings are used *Note: APCLK_SET is cleared by hardware once the clock settings are starting to be used.*
PLL_LOCK	0	r	**PLL Lock Indicator** 0_B Not locked 1_B Locked

1) Available only for products with SDADC. See the datasheet for details.

Table 26 Reset values of the APCLK_CTRL1 register

Reset source (RESET_TYPE_4)	Reset value
Power-on reset, brown-out reset, WDT1 reset, wake-up reset, hardware reset	50_H
Watchdog timer reset, soft reset	UU_H (U = unchanged)

The APCLK_CTRL1 register is reset by RESET_TYPE_4.

The clock source for the analog modules is selected via the APCLK1 and APCLK2 registers.

APCLK_CTRL2
Analog Peripheral Clock Control 2 Register **($06C_H$)** **Reset Value: 01_H**

7	6	5	4	3	2	1	0
RES				T3CLK_DIV		SDADCCLK_DIV	
r				rw		rw	

Field	Bits	Type	Description
RES	7:4	r	**Reserved** Returns 0 if read; should be written with 0.
T3CLK_DIV	3:2	rw	**Timer3 Clock Divider** This Flag configures the Timer3 clock divider. 00_B Divide by 1 01_B Divide by 2 10_B Divide by 4 11_B Divide by 8 T3CLK_DIV bit is a password-protected bit.
SDADCCLK_DIV	1:0	rw	**SDADC Clock Divider**[1)] This Flag configures the SDADC clock divider. 00_B Divide by 1 01_B Divide by 2 10_B Divide by 4 11_B Divide by 8 SDADCCLK_DIV is not a protected bit.

1) Available only for products with SDADC. See the datasheet for details.

Table 27 Reset value of the APCLK_CTRL2 register

Reset source (RESET_TYPE_4)	Reset value
Power-on reset, brown-out reset, WDT1 reset, wake-up reset, hardware reset	01_H
Watchdog timer reset, soft reset	UU_H (U = unchanged)

The APCLK_CTRL2 register is reset by RESET_TYPE_4.

The clock source for the analog modules is selected via the APCLK1 and APCLK2 registers.

The APCLK1 register is initialized with a value of 01_H by the BootROM code during the boot process.

APCLK1
Analog Peripheral Clock Register 1 **(058_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
APCLK3SCLR	**APCLK3STS**	**APCLK1STS**		**RES**	**APCLK1SCLR**	**APCLK1FAC**	
w	r	r		r	w	rw	

Field	Bits	Type	Description
APCLK3SCLR	7	w	**APCLK3STS Clear** Setting this bit clears the APCLK3STS flag.
APCLK3STS	6	r	**fSYS Loss of Clock Status** This bit indicates whether a loss of the f_{SYS} clock has been detected. 0_B No loss of f_{SYS} detected 1_B Loss of f_{SYS} clock detected
APCLK1STS	5:4	r	**Analog Peripherals Clock Status** This bit field indicates the status of the analog peripheral clock. 00_B The MI_CLK clock is in the required range 01_B The MI_CLK clock exceeds the upper limit 10_B The MI_CLK clock exceeds the lower limit 11_B The MI_CLK clock is outside the specified limits. The clock watchdog (see **Chapter 7**) monitors the frequency of the analog subsystem. If the clock is outside the required range, the watchdog triggers a system reset. *Note: The proper operation of the analog modules can only be ensured if their clock is in the required range.*
RES	3	r	**Reserved** Always read as zero.
APCLK1SCLR	2	w	**APCLK1STS Clear** Setting this bit clears the APCLK1STS flag.
APCLK1FAC	1:0	rw	**Analog Module Clock Factor** The factor by which the system clock is divided to generate the synchronous MI_CLK clock. 00_B Divide by 1 01_B Divide by 2 10_B Divide by 3 11_B Divide by 4 This bit field is not protected. This setting has an effect only when APCLK_SET = 1. *Note: If SYSCLKSEL[1] = 1 (LP_CLK), APCLK1FAC default value 00_H (divide by 1) is used.*

Table 28 Reset values of the APCLK1 register

Reset source (RESET_TYPE_4)	Reset value
Power-on reset, brown-out reset, WDT1 reset, wake-up reset, hardware reset	00_H
Watchdog timer reset, soft reset	UU_H (U = unchanged)

The APCLK1 register is reset by RESET_TYPE_4.

The clock source for the analog modules is selected via the APCLK1 and APCLK2 registers.

The APCLK2 register is initialized with a value of 13_H by the BootROM code during the boot process.

APCLK2
Analog Peripheral Clock Register 2 **($05C_H$)** **Reset Value: 08_H**

7	6	5	4	3	2	1	0
APCLK2SCLR	**APCLK2STS**		**APCLK2FAC**				
w	r		rw				

Field	Bits	Type	Description
APCLK2SCLR	7	w	**APCLK2STS Clear** Setting this bit clears the APCLK2STS flag.
APCLK2STS	6:5	r	**Analog Peripherals Clock Status** This bit field indicates the status of the analog peripheral clock source. 00_B The TFILT_CLK clock is in the required range 01_B The TFILT_CLK clock exceeds the upper limit 10_B The TFILT_CLK clock exceeds the lower limit 11_B The TFILT_CLK clock is outside the specified limit. The clock watchdog (see chapter **System control unit – power modules (SCU-PM)**) monitors the frequency of the analog subsystem. If the clock is not inside the required range, the watchdog triggers a system reset. *Note: The proper operation of the analog modules can only be ensured if their clock is in the required range.*

Field	Bits	Type	Description
APCLK2FAC	4:0	rw	**Slow Down Clock Divider for TFILT_CLK Generation** 00000_B f_{sys} 00001_B $f_{sys}/2$ 00010_B $f_{sys}/3$ 00011_B $f_{sys}/4$ 00100_B $f_{sys}/5$ 00101_B $f_{sys}/6$ 00110_B $f_{sys}/7$ 00111_B $f_{sys}/8$ 01000_B $f_{sys}/9$ 01001_B $f_{sys}/10$ 01010_B $f_{sys}/11$ 01011_B $f_{sys}/12$ … 11110_B $f_{sys}/31$ 11111_B $f_{sys}/32$ This setting has an effect only when APCLK_SET = 1. *Note: If SYSCLKSEL[1] = 1 (LP_CLK), the APCLK2FAC default value 8 is used.* *Note: f_{SYS} is divided by the APCLK2FAC factor to generate TFILT_CLK. TFILT_CLK should be always at 2 MHz.*

Table 29 Suggested values for APCLK2

Clock Frequency	APCLK2FAC
18 MHz (LP_CLK)	08_H (default)
20 MHz (PLL_CLK)	09_H
24 MHz (PLL_CLK)	$0B_H$
40 MHz (PLL_CLK)	13_H

Table 30 Reset values of the APCLK2 register

Reset source (RESET_TYPE_4)	Reset value
Power-on reset, brown-out reset, WDT1 reset, wake-up reset, hardware reset	$0B_H$
Watchdog timer reset, soft reset	UU_H (U = unchanged)

The APCLK2 register is reset by RESET_TYPE_4.

6.3.6.5 External Clock Control register

This register controls the settings of the external clock for CLKOUT.

COCON
Clock Output Control Register **($0B4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
EN	**COUTS1**	**TLEN**	**COUTS0**	**COREL**			
rw	rw	rw	rw	rw			

Field	Bits	Type	Description
EN	7	rw	**CLKOUT Enable** 0_B No external clock signal is provided. 1_B The configured external clock signal is provided.
COUTS1	6	rw	**Clock Out Source Select Bit 1** 0_B f_{CCLK} is selected. 1_B Based on setting of COUTS0.
TLEN	5	rw	**Toggle Latch Enable** Enable this bit if 50% duty cycle is desired on CLKOUT. This bit has an effect only when both COUTS1 and COUTS0 are set to 1. 0_B Toggle latch is disabled. The clock output frequency is specified by the COREL bit field. 1_B Toggle latch is enabled. The clock output frequency is half the frequency that is specified by the COREL bit field. The resulting output frequency has a 50% duty cycle.
COUTS0	4	rw	**Clock Out Source Select Bit 0** This bit has an effect only if COUTS1 is set to 1. 0_B The oscillator output frequency is selected. 1_B The clock output frequency is specified by the COREL bit field.

Field	Bits	Type	Description
COREL	3:0	rw	**Clock Output Divider** 0000_B f_{sys} 0001_B $f_{sys}/2$ 0010_B $f_{sys}/3$ 0011_B $f_{sys}/4$ 0100_B $f_{sys}/6$ 0101_B $f_{sys}/8$ 0110_B $f_{sys}/10$ 0111_B $f_{sys}/12$ 1000_B $f_{sys}/14$ 1001_B $f_{sys}/16$ 1010_B $f_{sys}/18$ 1011_B $f_{sys}/20$ 1100_B $f_{sys}/24$ 1101_B $f_{sys}/32$ 1110_B $f_{sys}/36$ 1111_B $f_{sys}/40$

The COCON register is reset by RESET_TYPE_4.

6.4 Reset control

This section describes the types of reset and the effects of each reset on the TLE987x.

6.4.1 Types of reset

The following reset types are recognized by the TLE987x:

- Power-on reset: The request for this reset is triggered by V_S undervoltage and transmitted asynchronously. The reset is released when the V_S supply voltage reaches the upper threshold of the undervoltage range.
- Brown-out reset: Is handled the same as a power-on reset.
- Wake-up reset: Requested asynchronously by a wake-up event in a power-saving mode (Stop mode or Sleep mode).
- Hardware reset: Requested asynchronously by an event on the external reset input pin .
- WDT1 reset: Activated asynchronously by a WDT1-reset event. See **Watchdog timer (WDT1)**.
- WDT reset: Requested by a WDT-reset event. See **Watchdog timer**.
- Soft reset: Requested synchronously by a soft-reset event.

6.4.2 Overview

When the TLE987x is first powered up or a brown-out condition triggered by supply voltage input(s) going below the threshold, proper voltage thresholds must be reached before the MCU system starts operating with the release of the MCU, CPU and NVM resets. With all resets (except soft and WDT resets), the boot configuration is latched. The CPU starts executing from the boot ROM firmware with the release of the MCU reset.

If the system is in Stop mode, it can be woken up with a reset. A wake-up reset is basically equivalent to a power-on reset, except that it is a ‘warm’ reset (VDDP does not need to ramp up) and certain registers are maintained across the reset. A wake-up via hard reset pin in Stop Mode is handled as a wake-up reset.

The hardware reset function via pin can be used anytime to restart the system.

The external watchdog timer (WDT1) can trigger a WDT1 reset of the system if the WDT1 is serviced wrongly, for example, if it overflows, or being serviced in the closed window part.

Likewise, the SCU watchdog timer (WDT) can trigger a watchdog timer reset of the system if the timer is not refreshed before it overflows.

A soft reset can be triggered by application software.

Note that the boot configuration is only latched during power-on, brown-out, WDT1, wake-up, and hardware resets.

6.4.3 Module reset behavior

Table 31 gives an overview of how the reset types affect the various modules or functions of the TLE987x. “n” means that the module or function is reset to its default state. See **Table 33** for the effects of multiple simultaneous resets.

Table 31 Effect of resets on modules and functions

Module or function	Power-on or brown-out reset	Wake-up reset[1)]	Hardware reset[1)]	WDT1 reset[1)]	WDT reset	Soft reset[2)]
CPU core	n	n	n	n	n	n
SCU	n Except reset indication bit	n Except indication bits	n Except reset indication bit	n Except reset indication bit	n Except certain status bits[3)]	n Except certain status bits[3)]
Peripherals	n	n	n	n	n	n
Debug system	n	n	n	n	n	n
Port control	n	n	n	n	n	n
Firmware startup execution	Executes all INIT	Sleep: Executes all INIT	Executes most INIT	Executes most INIT	Skips INIT that is not required	Skips INIT that is not required
On-chip static RAM	Initialized to 0	Sleep: Initialized to 0 Stop: Not affected[4)]	Not affected [4)5)]	Not affected[4)5)]	Not affected[4)]	Not affected[4)]
Memory extension stack RAM	Affected[6)]	Affected[6)]	Affected[6)]	Affected[6)]	Affected[6)]	Affected[6)]
NVM	n	n	n	n	n Except map RAM	n Except map RAM
Clock system incl. PLL	n	n	n	n	Not affected[7)]	Not affected[7)]

1) MCU subsystem: Hardware resets, WDT1 resets, and wake-up resets (from Stop mode or Sleep mode) mostly affect the hardware like power-on or brown-out resets. Any exceptions are mainly due to a power-on reset being a 'cold' start.
2) A soft reset can be requested using the CPU_AIRCR.**SYSRESETREQ** bit.
3) These bits include the reset requestor indication bit, the last latched boot configuration (for power on, brown-out, WDT1, and wake-up resets), and NMI status flags such as NMISR.
4) Not affected: A reset has no direct effect on RAM contents.
5) If the reset happens during a write to SRAM, the byte in the targeted write address may be corrupted.
6) If the reset happens when the CPU pushes data to the stack, the word at the memory address accessed by the push may be corrupted.
7) All configurations including trim settings.

6.4.4 Functional descriptions of reset types

This section describes the sources and priorities of the various reset types.

6.4.4.1 Power-on and brown-out reset

The power-on reset is the highest-level reset. It powers up and resets the entire system. A brown-out reset occurs when any required voltage drops below its minimum threshold.

In user mode, the system clock is switched to the PLL output at the defined frequency of the device.

6.4.4.2 Wake-up reset

A wake-up reset occurs when an enabled event is received in a power-saving mode on a defined functional input pin. Whether a wake-up reset happened in Sleep mode or Stop mode is shown by the respective indicator bits. When the system wakes up from Sleep mode, a reset is always executed and indicated by the corresponding flags.

A wake-up reset has the next-highest priority after power-on or brown-out resets.

In user mode, the system clock is switched to the PLL output at the defined frequency of the device.

6.4.4.3 Hardware reset

A hardware reset is requested asynchronously by an event on the external RESET (low active) input pin, and has the next-highest priority after a wake-up reset.

When a hardware reset is triggered while the device is in a power-saving mode, it is effectively a wake-up reset.

In user mode, the system clock is switched to the PLL output at the defined frequency of the device.

For details of programming the filter time of the external RESET (low active) input pin, see the corresponding reset pin blind time register, **CNF_RST_TFB**.

6.4.4.4 WDT1 reset

A WDT1 reset occurs due to WDT1 timer overflow or when servicing in a closed window, and has the next-highest priority after hardware resets.

In user mode, the system clock is switched to the PLL output at the defined frequency of the device.

6.4.4.5 WDT and soft reset

A WDT reset occurs due to WDT timer overflow. A soft reset occurs when software sets the soft reset request bit.

These two resets have the same priority – the lowest priority – and the same effect on the system. With these resets, the device continues to run on the previous clock system configuration.

6.4.5 Reset register description

RSTCON
Reset Control Register **(068_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
LOCKUP_EN	**RES**						**LOCKUP**
rwh	r						rw

Field	Bits	Type	Description
LOCKUP_EN	7	rwh	**Lockup Reset Enable Flag** 0_B Lockup reset is disabled. 1_B Lockup reset is enabled. The LOCKUP_EN bit is protected. When the protection scheme is activated, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
RES	6:1	r	**Reserved** Returns 0 if read; should be written with 0.
LOCKUP	0	rw	**Lockup Flag** 0_B Lockup status not active. 1_B Lockup status active. The LOCKUP bit is protected. When the protection scheme is activated, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.

The RSTCON register is reset by RESET_TYPE_3.

Note: The lockup reset is a reset provided by the Arm® Core. The effect of a lockup reset is the same as that of a software reset.

The PMU_RESET_STS1 and PMU_RESET_STS2 registers are located in the PMU and are described in **PMU_RESET_STS1** and **PMU_RESET_STS2**. The registers retain their values across all resets unless no power is available. In case of a power-on reset, only the PMU_1V5DidPOR bit is set.

Table 32 Reset values of the PMU_RESET_STS1 register

Reset source	Reset value
Wake-up reset in Stop mode	$0000\ 0010_B$
Wake-up reset via reset pin in Stop mode	$0100\ 0010_B$
Wake-up reset in Sleep mode	$0000\ 0100_B$
Soft reset	$0000\ 1000_B$
WDT reset	$0001\ 0000_B$
WDT1 reset	$0010\ 0000_B$

Table 32 Reset values of the PMU_RESET_STS1 register (cont'd)

Reset source	Reset value
Hardware reset (reset pin)	$0100\ 0000_B$
Power-on reset or brown-out reset	$1000\ 0000_B$

Possible combinations of reset sources

Table 33 lists all possible combinations of reset status bits set in the TLE987x. An "x" indicates that the reset listed in the column header has occurred.

Table 33 Effective resets for multiple simultaneous reset sources

Effective reset	Power-on or brown-out reset	Wake-up reset		Hardware reset (reset pin)	WDT1 reset	WDT reset	Soft reset
		Sleep mode	Stop mode				
Power-on	x	x	x	x	x	x	x
Wake-up from Sleep mode	–	x	–	x	x	x	–
Wake-up from Stop mode	–	–	x	x	x	x	–
Hardware (reset pin)	–	–	–	x	x	x	x
WDT1	–	–	–	–	x	x	x
WDT	–	–	–	–	–	x	–
Software	–	–	–	–	–	–	x
WDT and software	–	–	–	–	–	x	x

6.4.6 Booting scheme

After any power-on reset, brown-out reset, hardware reset, WDT1 reset, or wake-up reset, the TMS, P0.0, and P0.2 pins in combination specify a system mode. **Table 34** shows the boot selection options available in the TLE987x.

Table 34 TLE987x Boot Options

TMS	P0.0	P0.2	Mode
0	x	x	User mode / BSL mode
1	1	0	Debug mode with Serial Wire port

6.5 Power management

This section describes the features and functionality provided for the power management of the device.

6.5.1 Overview

The TLE987x power management system enables software to configure the various processing units so that they automatically adjust to draw the minimum necessary power for the application.

There are four power modes: Active mode, Slow-Down mode, Stop mode and Sleep mode, as shown in **Figure 28**. Sleep mode is a special case: It can only be exited with a system reset.

The operation of the system components in each of these states can be configured by software. The power modes enable flexible reduction of the power consumption through a combination of techniques, including:

- Stopping the CPU clock
- Stopping the clocks of other system components individually
- Reducing the clock speeds of some peripheral components
- Powering down the entire system with fast-restart capability
- Reducing or removing the power supply to power domains

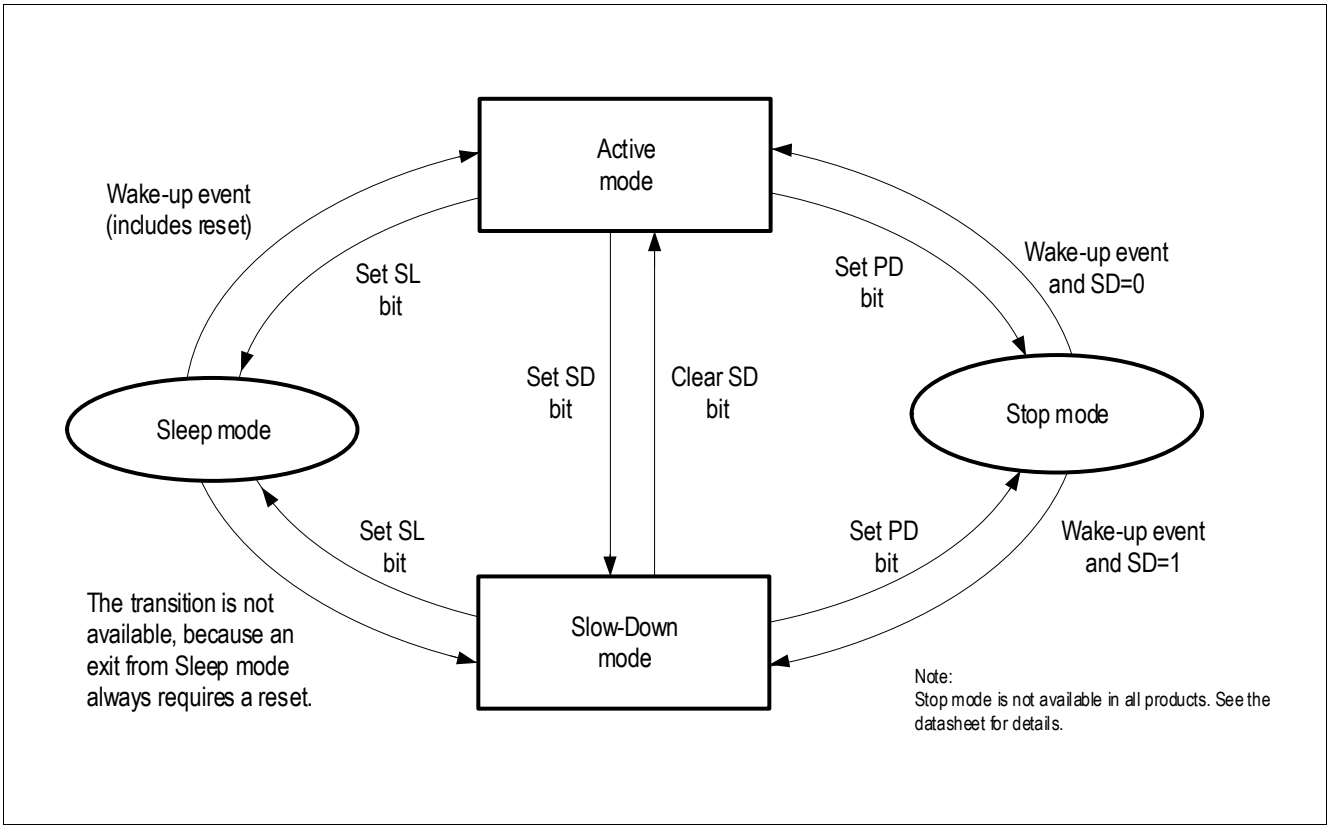


Figure 28 Transition between various modes of operation (without reset)

In Slow-Down mode, the clock generation unit is instructed to reduce its clock frequency by dividing the clock to the system, that is, core and peripherals, by a programmable factor.

In Stop mode, the clock is turned off. The device cannot, therefore, be awakened by an interrupt or the watchdog timer. It wakes only when it receives an external wake-up signal or reset signal. The application must be set up so that the TLE987x can be served with one of these signals. A wake-up circuit is used to detect enabled wake-up signals and activate the Stop mode wake-up. During Stop mode, this circuit remains active.

In Sleep mode, the power supply to the whole MCU subsystem is removed. When a wake-up event is detected, a system reset is generated, the MCU is reset to its default configuration, and then the restart operation is initialized.

The priority for entry to the power-saving modes, from highest to lowest, is: Sleep mode, Stop mode, then Slow-Down mode.

6.5.2 Functional description

This section describes the power-saving modes, their operation, and how they are entered and exited. It also describes the respective behavior of the TLE987x system components.

6.5.2.1 Slow-Down mode

The Slow-Down mode is used to reduce the power consumption by decreasing the internal clock in the device.

The Slow-Down mode is activated by setting the SD bit in the **PMCON0** SFR. The **CMCON1**.CLKREL bit field is used to select different slow-down frequencies. The CPU and peripherals are clocked at this lower frequency. The Slow-Down mode is terminated by clearing the SD bit.

6.5.2.2 Stop mode

In the Stop mode, the NVM is put into NVM Shutdown mode (analog and digital parts except the map RAM are shut down). The 5-V (VDDP) power supply to the analog modules ADC, PLL, and the internal oscillator is not removed. The MCU digital parts and the NVM map RAM are powered by the V_{DDCOUT} (VDDC) regulator (supplying $V_{DDCOUT_Stop_Red}$ in this mode). All functions of the microcontroller are stopped while the contents of the NVM, on-chip RAM, and the SFRs are maintained. The GPIOs are still powered.

In Stop mode, the clock is turned off. Hence, the system cannot be awakened by an interrupt or the watchdog timer. It will be awakened only when it receives an external wake-up signal (with or without a following system reset) or with a reset by asserting the hard reset pin.

Software requests Stop mode by setting the **PMCON0.PD** bit to 1. In addition to setting this flag, the **WFI** or **WFE** instructions have to be executed. As soon as the controller has finished its currently executed interrupt task, it will enter the Stop mode. **Figure 29** shows the correct sequence to enter Stop mode:

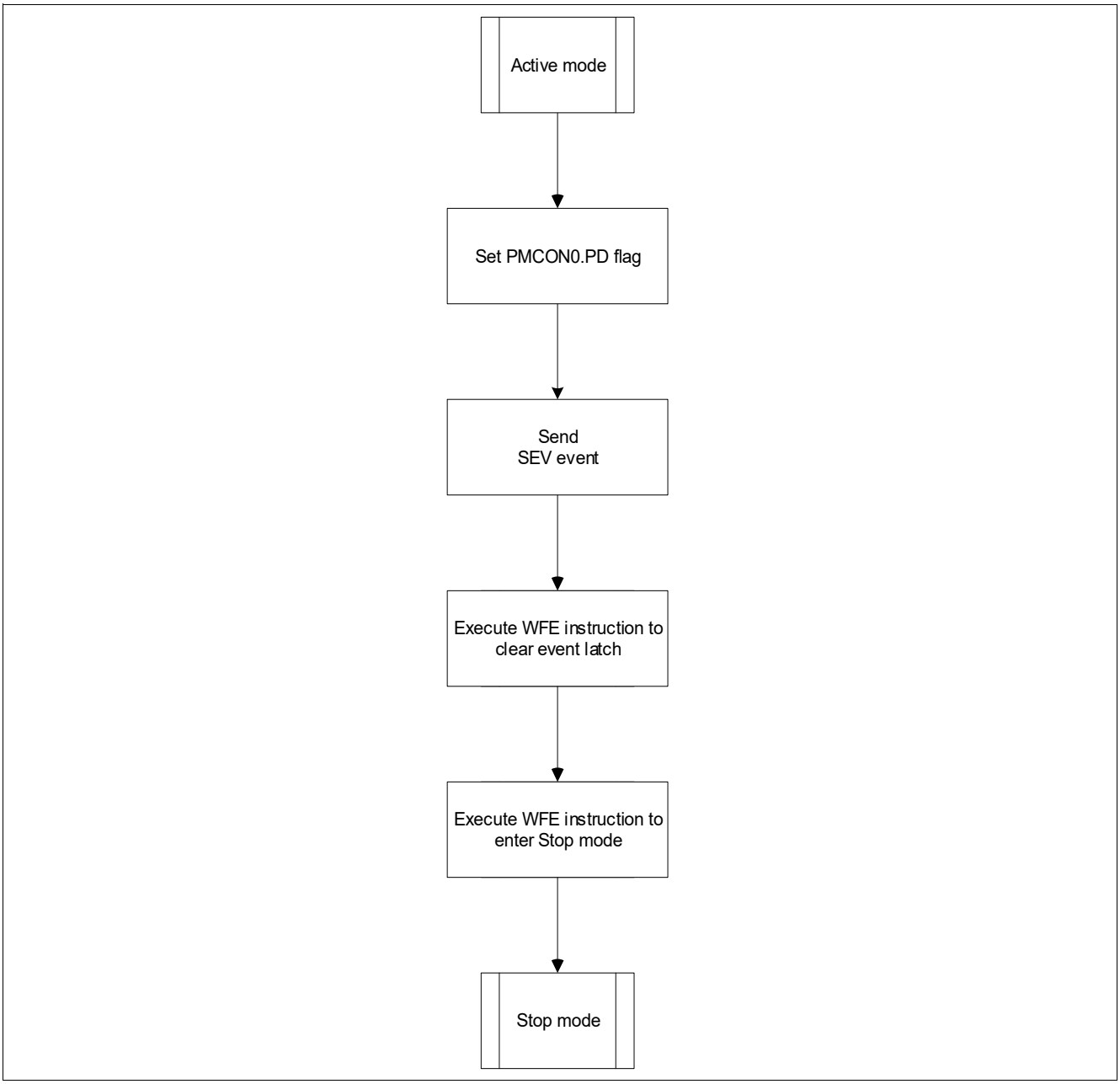


Figure 29 Stop mode entry programming sequence

Exiting Stop mode

Stop mode is exited when an active edge is detected on an enabled wake-up pin or by asserting the hard reset pin.

The wake-up circuitry performs a sequence of predefined actions such as restoring all supply voltages and restoring modules (including the oscillator and PLL) to operational mode. When the PLL lock is locked (that is, when the PLL clock is stable and the user configuration has been restored), the gating of the peripheral and CPU clocks is removed. The CPU starts to run, beginning with the instruction following the one that entered Stop mode (WFE). The user code must insert three NOP instructions following the one that sets the PD bit.

Note: If you have selected the PLL output as the system clock (typical use case), but the lock status of the PLL cannot be achieved, the PLL enters Freerunning mode and software execution continues in the PLL's Freerunning mode.

Usage of Arm® Core low-power modes for the Stop mode

The Arm® Core provides two low-power modes, which are called Sleep and Deep Sleep. For the system's Stop mode, the Deep Sleep mode of the core is used. To enable the Deep Sleep mode, configure the **SCR.SLEEPDEEP** register accordingly. Another way of entering Deep Sleep mode is to execute two dedicated mode change instructions:

- WFI
- WFE

When the controller enters stop mode via the WFI instruction, it executes the lowest-priority pending interrupt and after that enters Sleep mode. It is not recommended to use this feature for normal operation using the Stop mode, because the controller would only operate interrupt-triggered.

When the WFE instruction is used, the controller starts to operate triggered by an external event. If the CPU will be woken up be this external event, it stays in thread mode and continues to execute the code that it had executed before it entered Stop mode.

This is the recommended procedure to enter Stop mode.

6.5.2.3 Sleep mode

In the Sleep mode, the supply to the whole MCU subsystem including the ADC, PLL, and NVM is removed. The wake-up detection circuitry remains supplied. Only the non-volatile memory retains its contents. The external ports are not powered. The supply to ADC pads is removed.

Sleep mode is always exited with a system reset, which is triggered by an active edge on an enabled wake-up pin. It is not possible to exit Sleep mode by asserting the hard reset pin because the digital 5-V pads are not powered.

Software requests Sleep mode by setting the **PMCON0.SL** bit to 1.

Exiting Sleep mode

Sleep mode can only be exited with a system reset, triggered by an active edge on an enabled wake-up pin.

6.5.3 Register description

PMCON0
Power Mode Control Register 0 **(040_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES				**SD**	**PD**	**SL**	**XTAL_ON**
r				rw	rwh1	rwh1	rw

Field	Bits	Type	Description
RES	7:4	r	**Reserved** Returns 0 if read; should be written with 0.
SD	3	rw	**Slow Down Mode Enable. Active High.** Setting this bit causes the device to go into Slow-Down mode. It is reset by user code. The SD bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
PD	2	rwh1	**Stop Mode Enable. Active High.** Setting this bit causes the device to go into a Stop mode. It is reset by the wake-up circuit. The PD bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
SL	1	rwh1	**Sleep Mode Enable. Active High.** Setting this bit causes the device to go into Sleep mode. It is reset by the wake-up circuit. The SL bit is protected. When the protection scheme is active, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**.
XTAL_ON	0	rw	**OSC_HP Operation in Stop Mode**[1)] 0_B OSC_HP (XTAL) is suspended by hardware in Stop mode. 1_B OSC_HP (XTAL) continues to operate in Stop mode, if it is enabled by **OSC_CON**.XPD. This setting provides a way to reduce the power consumption in the Stop mode. However, note that the startup time of OSC_HP can be several milliseconds. Alternatively, to enable fast wake-ups from Stop mode while still reducing power consumption, you can selectively enable the internal oscillator as the clock source and disable OSC_HP before entering Stop mode.

1) XTAL support is not available in all products of the TLE987x family. See the datasheet for details.

6.6 Interrupt management

This section describes the management of interrupts by the system control unit.

6.6.1 Overview

The interrupt management submodule in the SCU controls the non-core-generated interrupt requests to the core. The core has one non-maskable interrupt (NMI) node and sixteen maskable interrupt nodes. **Figure 30** shows the block diagram of the interrupt management submodule.

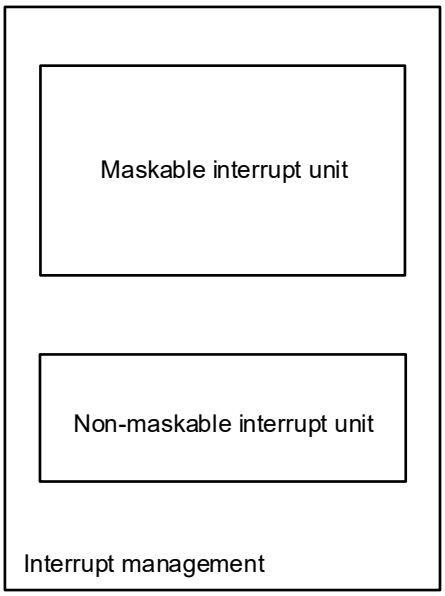


Figure 30 Interrupt management block diagram

The non-maskable interrupt unit controls the NMI requests. An incoming NMI request is not maskable and in this sense, differs from the regular interrupts. In addition, an NMI request always has the highest priority to be processed. In the TLE987x, eight different sources can generate NMIs: Watchdog timer prewarning, PLL loss-of-lock, oscillator watchdog event, NVM map error, memory-ECC error, NVM operation complete, Debug mode user IRAM event, and supply prewarning. Some NMI sources can be triggered by one of several events. These NMI sources are ORed to generate an NMI interrupt directly to the core. The triggering NMI sources or events are indicated in the NMI Status register (NMISR), and in some cases the event flags are located in the peripheral register. The NMI node source control can be configured via the NMI Control register (NMICON).

There are generally three types of maskable inputs into the core: internal, external, and extended interrupts. The maskable interrupt unit generates the respective interrupt node request to the core and maintains the corresponding SCU flags and control. In general, to support all types of peripheral interrupts, an interrupt node of the core may be shared among several interrupt sources.

6.6.1.1 External interrupts

Figure 31 shows how the SCU generates an interrupt request from an external source using edge detection. External interrupts can be triggered by positive, negative, or double edges. The EXICON0 register specifies the active edge for the external interrupt.

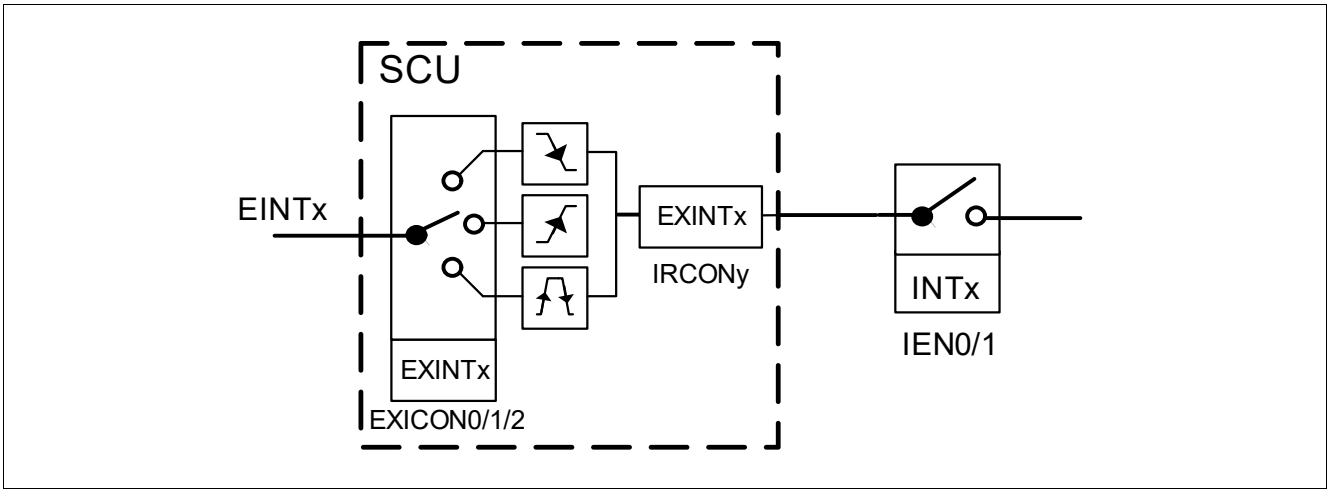


Figure 31 Interrupt request generation of external interrupts

6.6.1.2 Extended interrupts

Extended interrupts enable on-chip peripherals outside the core to trigger interrupt requests in the core. There are nine such interrupts.

Interrupt signals from such on-chip peripherals are pulse-triggered and active for two clock cycles. Any such interrupt signals belonging to the same interrupt node will be latched as one direct interrupt request to the core. IRCONx (where x = 0, 1, 3, 4) or peripheral registers hold the interrupt event flags for these extended and external interrupt events. Corresponding bits in the Interrupt Enable registers (IEN) in the core may block these interrupt requests or transfer them to the core interrupt controller. An enabled interrupt request is acknowledged when the core vectors to the interrupt routine. The software routine should clear the interrupt flags in the IRCONx registers.

As there are more peripheral interrupts than interrupt nodes supported by the core, some interrupts are multiplexed to the same interrupt node. Where possible and necessary, critical peripheral interrupts (for example, SSC) have their own dedicated interrupt node.

6.6.2 Interrupt node assignment

Table 35 shows the interrupt node assignment for TLE987x.

Table 35 NMI

Interrupt node	Vector address	Assignment for TLE987x
NMI	0008_H	Watchdog timer, PLL, NVM operation complete, oscillator watchdog, NVM map error, ECC error, supply prewarning, overtemperature prewarning

Table 36 Interrupt vector table

Service request	Node ID	Description
GPT1	0	GPT1 interrupt (T2-T4)
GPT2	1	GPT2 interrupt (T5-T6, CAPIN)
MU- ADC2/T3	2	Measurement unit, VBG, SDADC[1)], Timer3, BEMF
ADC1	3	ADC1 interrupt, VREF5V overload, VREF5V overvoltage or undervoltage, 10-bit ADC

Table 36 Interrupt vector table (cont'd)

Service request	Node ID	Description
CCU0	4	CCU6 node 0 interrupt
CCU1	5	CCU6 node 1 interrupt
CCU2	6	CCU6 node 2 interrupt
CCU3	7	CCU6 node 3 interrupt
SSC1	8	SSC1 interrupt (receive, transmit, error)
SSC2	9	SSC2 interrupt (receive, transmit, error)
UART1	10	UART1 (ASC-LIN) interrupt (receive, transmit), Timer2, linsync1, LIN
UART2	11	UART2 interrupt (receive, transmit), Timer21, external interrupt (EINT2)
EXINT0	12	External interrupt (EINT0), MON
EXINT1	13	External interrupt (EINT1)
BDRV/CP	14	Bridge driver and charge pump
DMA	15	DMA controller

1) Available only for products with SDADC. See the datasheet for details.

6.6.3 Interrupt-related registers

Several interrupt related registers are located in the SCU.

6.6.3.1 Interrupt event enable control

The two interrupt events of UART and three interrupt events of the SSC module have the interrupt structure 1 which is described in **Chapter 12.3.2**. As there are no bits in the module to enable or disable these interrupt events, bits are defined in the SCU register MODIEN1 and MODIEN2 for this purpose.

Register MODIEN1 is initialized with a value of 00_H by the BootROM code during the boot process.

Register MODIEN2 is initialized with a value of 00_H by the BootROM code during the boot process.

MODIEN1
Peripheral Interrupt Enable Register 1 **(030_H)** **Reset Value: $C7_H$**

7	6	5	4	3	2	1	0
TIEN1	**RIEN1**	**RES**			**RIREN1**	**TIREN1**	**EIREN1**
rw	rw		r		rw	rw	rw

Field	Bits	Type	Description
TIEN1	7	rw	**UART 1 Transmit Interrupt Enable** 0_B Transmit interrupt is disabled 1_B Transmit interrupt is enabled
RIEN1	6	rw	**UART 1 Receive Interrupt Enable** 0_B Receive interrupt is disabled 1_B Receive interrupt is enabled
RES	5:3	r	**Reserved** Returns 0 if read; should be written with 0.
RIREN1	2	rw	**SSC 1 Receive Interrupt Enable** 0_B Receive interrupt is disabled 1_B Receive interrupt is enabled
TIREN1	1	rw	**SSC 1 Transmit Interrupt Enable** 0_B Transmit interrupt is disabled 1_B Transmit interrupt is enabled
EIREN1	0	rw	**SSC 1 Error Interrupt Enable** 0_B Error interrupt is disabled 1_B Error interrupt is enabled

MODIEN2
Peripheral Interrupt Enable Register 2 **(034_H)** **Reset Value: $C7_H$**

7	6	5	4	3	2	1	0
TIEN2	**RIEN2**	**EXINT2_EN**	**RES**		**RIREN2**	**TIREN2**	**EIREN2**
rw	rw	rw	r		rw	rw	rw

Field	Bits	Type	Description
TIEN2	7	rw	**UART 2 Transmit Interrupt Enable** 0_B Transmit interrupt is disabled 1_B Transmit interrupt is enabled
RIEN2	6	rw	**UART 2 Receive Interrupt Enable** 0_B Receive interrupt is disabled 1_B Receive interrupt is enabled
EXINT2_EN	5	rw	**External Interrupt 2 Enable** 0_B External interrupt is disabled 1_B External interrupt is enabled
RES	4:3	r	**Reserved** Returns 0 if read; should be written with 0.
RIREN2	2	rw	**SSC 2 Receive Interrupt Enable** 0_B Receive interrupt is disabled 1_B Receive interrupt is enabled
TIREN2	1	rw	**SSC 2 Transmit Interrupt Enable** 0_B Transmit interrupt is disabled 1_B Transmit interrupt is enabled
EIREN2	0	rw	**SSC 2 Error Interrupt Enable** 0_B Error interrupt is disabled 1_B Error interrupt is enabled

MODIEN3
Peripheral Interrupt Enable Register 3 **(038_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		MONSTS	MONIE	RES			IE0
r		r	rw	r			rw

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.
MONSTS	5	r	**MON Input Status** 0_B Status zero 1_B Status one
MONIE	4	rw	**MON Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
RES	3:1	r	**Reserved** Returns 0 if read; should be written with 0.
IE0	0	rw	**External Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

MODIEN4
Peripheral Interrupt Enable Register 4 **($03C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES							IE1
r							rw

Field	Bits	Type	Description
RES	7:1	r	**Reserved** Returns 0 if read; should be written with 0.
IE1	0	rw	**External Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

Register GPT12IEN is initialized with a value of 00_H by the BootROM code during the boot process.

GPT12IEN
GPT12 Interrupt Enable Register ($15C_H$) **Reset Value: $3F_H$**

7	6	5	4	3	2	1	0
RES		CRIE	T6IE	T5IE	T4IE	T3IE	T2IE
r		rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.
CRIE	5	rw	**General Purpose Timer12 Capture and Reload Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
T6IE	4	rw	**General Purpose Timer12 T6 Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
T5IE	3	rw	**General Purpose Timer12 T5 Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
T4IE	2	rw	**General Purpose Timer12 T4 Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
T3IE	1	rw	**General Purpose Timer12 T3 Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
T2IE	0	rw	**General Purpose Timer12 T2 Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

Other interrupt-related registers

The following interrupt-related registers are located in the SCU:

- NMICON
- NMISR
- IRCON0, IRCON1, IRCON2, IRCON3, IRCON4
- EXICON0
- MODIEN1, MODIEN2

All these registers, except MODIEN1 and MODIEN2, are described in **Chapter 12.5**.

6.6.3.2 DMA interrupt event enable and select control

The integrated DMA controller has 14 dedicated interrupt sources. They are:

- 10-bit ADC programmed sequencer sequence done
- 10-bit ADC single sequence done
- SSC transmit request
- SSC receive request
- ADC1-CH1
- ADC1-CH2
- ADC1-CH3
- ADC1-CH4
- ADC1-CH5
- ADC1-CH6
- ADC1-CH7, CCU T12_ZM
- ADC1-CH8, CCU T12_PM
- Timer3 , GPT12-T3
- SDADC[1)]

All DMA related interrupt-enable registers are described below.

1) Available only for products with SDADC. See the datasheet for details.

DMAIEN1
DMA Interrupt Enable Register 1 **(144_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
CH8IE	**CH7IE**	**CH6IE**	**CH5IE**	**CH4IE**	**CH3IE**	**CH2IE**	**CH1IE**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
CH8IE	7	rw	**DMA Channel 11 Interrupt Enable (ADC1 Channel 7)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH7IE	6	rw	**DMA Channel 10 Interrupt Enable (ADC1 Channel 6)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH6IE	5	rw	**DMA Channel 9 Interrupt Enable (ADC1 Channel 5)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH5IE	4	rw	**DMA Channel 8 Interrupt Enable (ADC1 Channel 4)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH4IE	3	rw	**DMA Channel 7 Interrupt Enable (ADC1 Channel 3)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH3IE	2	rw	**DMA Channel 6 Interrupt Enable (ADC1 Channel 2)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH2IE	1	rw	**DMA Channel 5 Interrupt Enable (ADC1 Channel 1)** 0_B Interrupt is disabled 1_B Interrupt is enabled
CH1IE	0	rw	**DMA Channel 4 Interrupt Enable (ADC1 Channel 0)** 0_B Interrupt is disabled 1_B Interrupt is enabled

DMAIEN2
DMA Interrupt Enable Register 2 **(148_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**SDADCIE**	**GPT12IE**	**SSCRXIE**	**SSCTXIE**	**TRSEQ2RDYIE**	**TRSEQ1RDYIE**	**TRERRIE**
r	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
SDADCIE	6	rw	**DMA SDADC Transfer Interrupt Enable**[1)] 0_B Interrupt is disabled 1_B Interrupt is enabled
GPT12IE	5	rw	**DMA Channel 12 Interrupt Enable (GPT12/Timer3)** 0_B Interrupt is disabled 1_B Interrupt is enabled
SSCRXIE	4	rw	**DMA Channel 3 Interrupt Enable (SSCx Receive)** 0_B Interrupt is disabled 1_B Interrupt is enabled
SSCTXIE	3	rw	**DMA Channel 2 Interrupt Enable (SSCx Transmit)** 0_B Interrupt is disabled 1_B Interrupt is enabled
TRSEQ2RDYIE	2	rw	**DMA Channel 1 Interrupt Enable (ADC1 ESM)** 0_B Interrupt is disabled 1_B Interrupt is enabled
TRSEQ1RDYIE	1	rw	**DMA Channel 0 Interrupt Enable (ADC1 Sequence)** 0_B Interrupt is disabled 1_B Interrupt is enabled
TRERRIE	0	rw	**DMA Transfer Error Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

1) Available only for products with SDADC. See the datasheet for details.

DMASRCSEL
DMA Source Selection Register **($14C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
GPT12_T3	**SSCRX**	**SSCTX**	**RES**	**T12PM_DMAEN**	**T12ZM_DMAEN**	**SSCRXSRCSEL**	**SSCTXSRCSEL**
r	r	r	r	rw	rw	rw	rw

Field	Bits	Type	Description
GPT12_T3	7	r	**DMA Channel 12 Request (GPT12E, Timer3)** 0_B GPT12_T3 transfer DMA request not present 1_B GPT12_T3 transfer DMA request present
SSCRX	6	r	**DMA Channel 3 Request (SSCx Receive)** 0_B SSC receive DMA request not present 1_B SSC receive DMA request present
SSCTX	5	r	**DMA Channel 2 Request (SSCx Transmit)** 0_B SSC transmit DMA request not present 1_B SSC transmit DMA request present
RES	4	r	**Reserved** *Note: This bit should be written with zero.*
T12PM_DMAEN	3	rw	**CC6_T12_PM (Period Match) DMA multiplexer, DMA channel 11** 0_B ADC1 channel 7 used as trigger for DMA channel 11 1_B CCU6 T12_PM used as trigger for DMA channel 11
T12ZM_DMAEN	2	rw	**CC6_T12_ZM (Zero Match) DMA multiplexer, DMA Channel 9** 0_B ADC1 Channel 5 used as trigger for DMA channel 9 1_B CCU6 T12_ZM used as trigger for DMA channel 9
SSCRXSRCSEL	1	rw	**SSCx Receive Source Select** 0_B SSC1 receive as DMA input is enabled 1_B SSC2 receive as DMA input is enabled
SSCTXSRCSEL	0	rw	**SSCx Transmit Source Select** 0_B SSC1 transmit as DMA input is enabled 1_B SSC2 transmit as DMA input is enabled

DMASRCCLR
DMA Source Selection Clear Register **(198_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
GPT12_T3C	SSCRXC	SSCTXC	RES				
w	w	w	r				

Field	Bits	Type	Description
GPT12_T3C	7	w	**DMA Channel 12 Transmit Request Clear (GPT12E, Timer3)** 0_B GPT12_T3 transfer DMA request not cleared 1_B GPT12_T3 transfer DMA request cleared
SSCRXC	6	w	**DMA Channel 3 Request Clear (SSCx Receive)** 0_B SSC receive DMA request not cleared 1_B SSC receive DMA request cleared
SSCTXC	5	w	**DMA Channel 2 Request Clear (SSCx Transmit)** 0_B SSC transmit DMA request not cleared 1_B SSC transmit DMA request cleared
RES	4:0	r	**Reserved** *Note: These bits should be written with zero.*

DMASRCSEL2
DMA Source Selection Register 2 **(180_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES						GPT12_DMAEN	
r						rw	

Field	Bits	Type	Description
RES	7:2	r	**Reserved** *Note: These bits should be written with zero.*
GPT12_DMAEN	1:0	rw	**DMA Channel 12 trigger mux select (GPT12E, Timer3)** *Note: Do not change the DMA channel 12 multiplexer if DMA channel 12 is enabled in the DMA controller.* 00_B Timer3 triggers DMA channel 12 01_B GPT12 T3 triggers DMA channel 12 10_B Timer3 or CCU6 T12_ZM triggers DMA channel 12 11_B GPT12 T3 or CCU6 T12_ZM triggers DMA channel 12

6.6.4 NMI event flag handling

All NMI event and status flags are retained across WDT and soft resets. Specifically, this includes all flags of the NMISR register: FNMIWDT, FNMIPLL, FNMINVM, FNMIOT, FNMIOWD, FNMIMAP, and, indirectly, FNMIECC and FNMISUP. In the case of watchdog resets, the requestor can be identified via the reset indicator bits WDT1RST and WDTRST. The ECC NMI is indicated by the respective event flags of SFR EDCSTAT.IRDBE, XRDBE, and NVMDBE. Likewise, the supply prewarning NMI and MI_CLK WDT NMI is indicated by the respective event flags located in SCU_PM. See **Chapter 7** for details.

These NMI event and status flags are otherwise reset to their default values by all other resets, that is, power-on, brown-out, hardware, WDT1 (except WDT1 reset), and wakeup reset.

6.7 General port control

The SCU contains control registers for the selection of:

- Alternate input functions of UART, timers and external interrupts (**Section 6.7.1**)
- Port output driver strength and temperature compensation (**Section 6.7.2**)

For functional descriptions of GPIO ports, refer to **Chapter 14**.

6.7.1 Input pin function selection

MODPISELx registers control the selection of the input pin functions. For UART, selecting an RXD line also enables the corresponding TXD line.

MODPISEL
Peripheral Input Select Register **($0B8_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
U_TX_CONDIS	**URIOS1**	**EXINT2IS**		**EXINT1IS**		**EXINT0IS**	
rw	rw	rw		rw		rw	

Field	Bits	Type	Description
U_TX_CONDIS	7	rw	**UART1 TxD Connection Disable** 0_B UART1 TxD1 output is routed to the LIN transmitter 1_B UART1 TxD1 output is disconnected from the LIN transmitter Transceiver TxD Input is routed to P0.1 *Note: To select TXD_1 as the UART1 output on P0.1, the Port ALTSELx registers also need to be configured appropriately.*
URIOS1	6	rw	**UART1 Input/Output Select** 0_B UART1 receiver input RxD1 is routed to the LIN receiver 1_B UART1 receiver input RxD1 is routed to P1.4 *Note: To select the RxD output of the transmitter on P1.4, the Port ALTSELx registers also need to be configured appropriately.*
EXINT2IS	5:4	rw	**External Interrupt 2 Input Select** 00_B External Interrupt Input EXINT2_0 is selected (P2.7)[1)] 01_B External interrupt input EXINT2_1 is selected (P1.4) 10_B External interrupt input EXINT2_2 is selected (P0.4) 11_B External interrupt input EXINT2_3 is selected (P0.0)
EXINT1IS	3:2	rw	**External Interrupt 1 Input Select** 00_B External interrupt input EXINT1_0 is selected (P1.1) 01_B External interrupt input EXINT1_1 is selected (P1.3) 10_B External interrupt input EXINT1_2 is selected (P1.0) 11_B External interrupt input EXINT1_3 is selected (P2.4)

Field	Bits	Type	Description
EXINT0IS	1:0	rw	**External Interrupt 0 Input Select** 00_B External interrupt input EXINT0_0 is selected (P2.0) 01_B External interrupt input EXINT0_1 is selected (P1.2) 10_B External interrupt input EXINT0_2 is selected (P0.1) 11_B External interrupt input EXINT0_3 is selected (P2.3)

1) P2.7 is not available in all products. See the datasheet for details.

MODPISEL1

Peripheral Input Select Register 1 **($0BC_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
T21EXCON	**T2EXCON**	**RES**		**RES**			**GPT12CAPINB**
rw	rw	r		r			rw

Field	Bits	Type	Description
T21EXCON	7	rw	**Timer21 External Input Control** 0_B Timer21 input T21EX is selected by the MODPISEL2.T2EXIS bit field 1_B Timer21 input T21EX is connected to the CCU6.COUT60 (Ch0) signal
T2EXCON	6	rw	**Timer2 External Input Control** 0_B Timer2 input T2EX is selected by the MODPISEL2.T2EXIS bit field 1_B Timer2 input T2EX is connected to the CCU6.CC60 (Ch0) signal
RES	5:4	r	**Reserved** Returns 0 if read; should be written with 0
RES	3:1	r	**Reserved** Returns 0 if read; should be written with 0
GPT12CAPINB	0	rw	**GPT12 CAPINB Input Control** Set GPT12E->**PISEL**.ISCAPIN = 1 0_B GPT12 CAPINB input is connected to P0.3 1_B GPT12 CAPINB input is connected to the BEMF comparator output. See **TRIM_BEMFx**.BEMF_GPT_CAPIN_SEL.

MODPISEL2
Peripheral Input Select Register 2 **($0C0_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
T21EXIS		**T2EXIS**		**T21IS**		**T2IS**	
rw		rw		rw		rw	

Field	Bits	Type	Description
T21EXIS	7:6	rw	**Timer21 External Input Select** 00_B Timer21 input T21EX_0 (P0.2) is selected 01_B Timer21 input T21EX_1 (P1.4) is selected 10_B Timer21 input T21EX_2 (P2.3) is selected 11_B Timer21 input T21EX_3 (P1.2) is selected *Note: This setting takes effect only when MODPISEL1.T21EXCON = 0.*
T2EXIS	5:4	rw	**Timer2 External Input Select** 00_B Timer2 input is connected to the integrated LIN transceiver 01_B Timer2 input T2EX_1 (P1.2) is selected Others: Reserved *Note: This setting takes effect only when MODPISEL1.T2EXCON = 0.*
T21IS	3:2	rw	**Timer21 Input Select** 00_B Timer21 input T21_0 (P0.1) is selected 01_B Timer21 input T21_1 (P1.1) is selected 10_B Timer21 input T21_2 (P0.4) is selected 11_B Reserved
T2IS	1:0	rw	**Timer2 Input Select** 00_B Timer2 input T2_0 (P0.0) is selected 01_B Timer2 input T2_1 (P2.5) is selected Others: Reserved

MODPISEL3
Peripheral Input Select Register 3 **($0C4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**URIOS2**	**RES**					
r	rw	r					

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
URIOS2	6	rw	**UART2 Input/Output Select** 0_B UART2 receiver input RXD2_0 (P1.2) and transmitter output TXD2_0 (P1.1) are selected 1_B UART2 receiver input RXD2_1 (P2.5) and transmitter output TXD2_1 (P1.3) are selected *Note: To select TXD2_1 as the transmitter output, the Port ALTSELx registers also need to be configured appropriately.*
RES	5:0	r	**Reserved** Returns 0 if read; should be written with 0.

6.7.2 Port output control

Px_POCONy registers control the output driver strength for each of the bidirectional port pins through the PDMn bit fields, where x denotes the port number and n denotes the pin number.

P0_POCON0
Port Output Control Register **($0E8_H$)** **Reset Value: 44_H**

7	6	5	4	3	2	1	0
RES	**PDM1**			**RES**	**PDM0**		
r	rw			r	rw		

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
PDM1	6:4	rw	**P0.1 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Strong driver and sharp edge 001_B Strong driver and medium edge 010_B Strong driver and soft edge 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
PDM0	2:0	rw	**P0.0 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Strong driver and sharp edge 001_B Strong driver and medium edge 010_B Strong driver and soft edge 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver

1) Specifies the current the respective driver can deliver to the external circuitry.
2) Specifies the switching characteristics of the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is, when changing the output level.

P0_POCON1
Port Output Control Register **($0EC_H$)** **Reset Value: 44_H**

7	6	5	4	3	2	1	0
RES	**PDM3**			**RES**	**PDM2**		
r	rw			r	rw		

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
PDM3	6:4	rw	**P0.3 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Strong driver and sharp edge 001_B Strong driver and medium edge 010_B Strong driver and soft edge 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
PDM2	2:0	rw	**P0.2 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Strong driver and sharp edge 001_B Strong driver and medium edge 010_B Strong driver and soft edge 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver

1) Specifies the current the respective driver can deliver to the external circuitry.
2) Specifies the switching characteristics of the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is, when changing the output level.

P0_POCON2
Port Output Control Register **($0F0_H$)** **Reset Value: 44_H**

7	6	5	4	3	2	1	0
RES	**RES**			**RES**	**PDM4**		
r	r			r	rw		

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
RES	6:4	r	**Reserved** Returns 100 if read; should be written with 100.
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
PDM4	2:0	rw	**P0.4 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Not used 001_B Not used 010_B Not used 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver

1) Specifies the current the respective driver can deliver to the external circuitry.
2) Specifies the switching characteristics of the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is, when changing the output level.

P1_POCON0
Port Output Control Register **($0F8_H$)** **Reset Value: 44_H**

7	6	5	4	3	2	1	0
RES	**PDM1**			**RES**	**PDM0**		
r	rw			r	rw		

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
PDM1	6:4	rw	**P1.1 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Not used 001_B Not used 010_B Not used 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
PDM0	2:0	rw	**P1.0 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Not used 001_B Not used 010_B Not used 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver

1) Specifies the current the respective driver can deliver to the external circuitry.
2) Specifies the switching characteristics of the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is, when changing the output level.

P1_POCON1
Port Output Control Register **($0FC_H$)** **Reset Value: 44_H**

7	6	5	4	3	2	1	0
RES	**PDM3**			**RES**	**PDM2**		
r	rw			r	rw		

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
PDM3	6:4	rw	**P1.3 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Strong driver and sharp edge 001_B Strong driver and medium edge 010_B Strong driver and soft edge 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
PDM2	2:0	rw	**P1.2 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Not used 001_B Not used 010_B Not used 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver

1) Specifies the current the respective driver can deliver to the external circuitry.
2) Specifies the switching characteristics of the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is, when changing the output level.

P1_POCON2
Port Output Control Register **(100_H)** **Reset Value: 04_H**

7	6	5	4	3	2	1	0
RES					**PDM4**		
r					rw		

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.
PDM4	2:0	rw	**P1.4 Port Driver Mode** Code driver strength[1] and edge shape[2] 000_B Strong driver and sharp edge 001_B Strong driver and medium edge 010_B Strong driver and soft edge 011_B Weak driver 100_B Medium driver 101_B Medium driver 110_B Medium driver 111_B Weak driver

1) Specifies the current the respective driver can deliver to the external circuitry.
2) Specifies the switching characteristics of the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is, when changing the output level.

The TCCR register controls the temperature compensations of all output port pins with strong drivers on the device level. The TCCR register has no effect on output port pins that operate in weak and medium driver modes.

TCCR
Temperature Compensation Control Register (0F4$_H$) **Reset Value: 03$_H$**

7	6	5	4	3	2	1	0
RES						TCC	
r						rw	

Field	Bits	Type	Description
RES	7:2	r	**Reserved** Returns 0 if read; should be written with 0.
TCC	1:0	rw	**Temperature Compensation Control** The slew rate of the output driver is kept stable over the selected temperature range: 00_B T_J: -40°C to 0°C 01_B T_J: 0°C to 40°C 10_B T_J: 40°C to 80°C 11_B T_J: 80°C to 175°C

6.7.3 GPT12 T3IN and T4IN input pin function selection

GPT12PISEL registers control the selection of the input pin functions of T3INB and T4IND in GPT12.

GPT12PISEL
GPT12 Peripheral Input Select Register **($0D0_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		**T3_GPT12_SEL**	**TRIG_CONF**	**GPT12**			
r		rw	rw	rw			

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.
T3_GPT12_SEL	5	rw	**CCU6_INT_SEL** 0_B CCU6_INT is triggered by Timer3 1_B CCU6_INT is triggered by GPT12PISEL.GPT12.
TRIG_CONF	4	rw	**CCU6 Trigger Configuration** 0_B Trigger is present for just one measurement (default) 1_B Trigger is present until the next input edge (selected by GPT12) – continuous measurement.
GPT12	3:0	rw	**GPT12 TIN3B / TIN4D Input Select** 0000_BCC60 0001_BCC61 0010_BCC62 0011_BT12 ZM. 0100_BT12 PM. 0101_BT12 CM0. 0110_BT12 CM1. 0111_BT12 CM2. 1000_BT13 PM. 1001_BT13 ZM. 1010_BT13 CM. 1011_BAny positive or negative edge on CC60/61/62. 1100_BRES. 1101_BRES. 1110_BRES. 1111_BRES.

6.8 Flexible peripheral management

The Flexible Peripheral Management submodule provides greater control of the operational status of each individual digital peripheral when you design a system. Peripherals which are not required for a particular functionality can be disabled by programming the assigned register bits to gate off the clock inputs. Doing so reduces the overall power consumption of the microcontroller.

Each register bit controls one peripheral. When this bit is set, the request signal to gate the peripheral clock is activated. The peripheral then synchronizes the gating-off of the clock to the peripheral.

6.8.1 Peripheral management registers

PMCON1
Peripheral Management Control Register 1 **(060_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		**RES**	**GPT12_DIS**	**T2_DIS**	**CCU6_DIS**	**SSC1_DIS**	**ADC1_DIS**
r		r	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.
RES	5	r	**Reserved** Returns 0 if read; should be written with 0.
GPT12_DIS	4	rw	**General Purpose Timer12 Disable Request. Active high.** 0_B GPT12 operates normally (default) 1_B Request to disable GPT12
T2_DIS	3	rw	**T2 Disable Request. Active high.** 0_B T2 operates normally (default) 1_B Request to disable T2
CCU6_DIS	2	rw	**CCU6 Disable Request. Active high.** 0_B CCU6 operates normally (default) 1_B Request to disable CCU6
SSC1_DIS	1	rw	**SSC1 Disable Request. Active high.** 0_B SSC1 operates normally (default) 1_B Request to disable SSC1
ADC1_DIS	0	rw	**ADC1 Disable Request. Active high.** 0_B ADC1 operates normally (default) 1_B Request to disable ADC1

PMCON2
Peripheral Management Control Register 2 **(064_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		T3_DIS	RES	T21_DIS	RES	SSC2_DIS	RES
r		rw	r	rw	r	rw	r

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.
T3_DIS	5	rw	**T3 Disable Request. Active high.** 0_B T3 operates normally (default) 1_B Request to disable T3
RES	4	r	**Reserved** Returns 0 if read; should be written with 0.
T21_DIS	3	rw	**T21 Disable Request. Active high.** 0_B T21 operates normally (default) 1_B Request to disable T21
RES	2	r	**Reserved** Returns 0 if read; should be written with 0.
SSC2_DIS	1	rw	**SSC2 Disable Request. Active high.** 0_B SSC2 operates normally (default) 1_B Request to disable SSC2
RES	0	r	**Reserved** Returns 0 if read; should be written with 0.

6.9 Module suspend control

If the Arm® core is halted while in Debug mode, timers in certain modules in the TLE987x can be suspended based on the settings of the corresponding module suspension bits in the in the **MODSUSP1** and **MODSUSP2** registers. When a module is suspended, only the timer stops counting as the counter input clock is gated off. The module is still clocked so that module registers are accessible.

MODSUSP1
Module Suspend Control Register 1 **($0C8_H$)** **Reset Value: 81_H**

7	6	5	4	3	2	1	0
RES	**T21_SUSP**	**RES**	**GPT12_SUSP**	**T2_SUSP**	**T13SUSP**	**T12SUSP**	**WDTSUSP**
r	rw	r	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
T21_SUSP	6	rw	**Timer21 Debug Suspend Bit** 0_B Timer21 is not suspended. 1_B Timer21 is suspended.
RES	5	r	**Reserved** Returns 0 if read; should be written with 0.
GPT12_SUSP	4	rw	**GPT12 Debug Suspend Bit** 0_B GPT12 is not suspended. 1_B GPT12 is suspended.
T2_SUSP	3	rw	**Timer2 Debug Suspend Bit** 0_B Timer2 is not suspended. 1_B Timer2 is suspended.
T13SUSP	2	rw	**Timer13 Debug Suspend Bit** 0_B T13 in the capture/compare unit is not suspended. 1_B T13 in the capture/compare unit is suspended. When this timer is suspended, its PWM output is also set to the inactive level.
T12SUSP	1	rw	**Timer12 Debug Suspend Bit** 0_B T12 in the capture/compare unit is not suspended. 1_B T12 in the capture/compare unit is suspended. When this timer is suspended, its T12 PWM outputs are also set to the inactive level and capture inputs are disabled.
WDTSUSP	0	rw	**SCU Watchdog Timer Debug Suspend Bit** 0_B WDT is not suspended. 1_B WDT is suspended.

MODSUSP2
Module Suspend Control Register 2 **($0CC_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES					**ADC1_SUSP**	**MU_SUSP**	**T3_SUSP**
r					rw	rw	rw

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.
ADC1_SUSP	2	rw	**ADC1 Unit Debug Suspend Bit** 0_B ADC1 is not suspended. 1_B ADC1 is suspended.
MU_SUSP	1	rw	**Measurement Unit Debug Suspend Bit** 0_B MU is not suspended. 1_B MU is suspended.
T3_SUSP	0	rw	**Timer3 Debug Suspend Bit** 0_B Timer3 is not suspended. 1_B Timer3 is suspended.

6.10 Watchdog timer

There are two watchdog timers in the system: the SCU watchdog timer (WDT) in the TLE987x, and the external watchdog timer (WDT1). This section describes the SCU WDT.

The watchdog timer is a submodule of the system control unit (SCU). The watchdog timer (WDT) provides a highly reliable and secure way to detect and recover from software or hardware failures. The WDT helps to abort an accidental malfunction of the TLE987x in a user-specified time period. When enabled, the WDT causes the TLE987x system to be reset if the WDT is not serviced within a user-programmable time period. The CPU must service the WDT within this time interval to prevent the WDT from causing a TLE987x system reset. Hence, routine service of the WDT confirms that the system is functioning properly.

The WDT is disabled by default.

In Debug Monitor mode, the WDT is suspended by default and stops counting. Its debug suspend bit is set by default, that is, **MODSUSP1**.WDTSUSP = 1. During debugging, there is therefore no need to refresh the WDT. Refer to **Chapter 6.9**.

Features:

- 16-bit watchdog timer
- Programmable reload value for the upper 8 bits of the timer
- Programmable window
- Selectable input frequency of $f_{PCLK}/2$ or $f_{PCLK}/128$

6.10.1 Functional description

The watchdog timer is a 16-bit timer, which is incremented by a count rate of $f_{PCLK}/2$ or $f_{PCLK}/128$. This 16-bit timer is implemented as two concatenated 8-bit timers. The upper 8 bits of the watchdog timer can be preset to a user-programmable value via a watchdog service access in order to vary the watchdog expiration interval. The lower 8 bits are reset by each service access. **Figure 32** shows the block diagram of the watchdog timer unit.

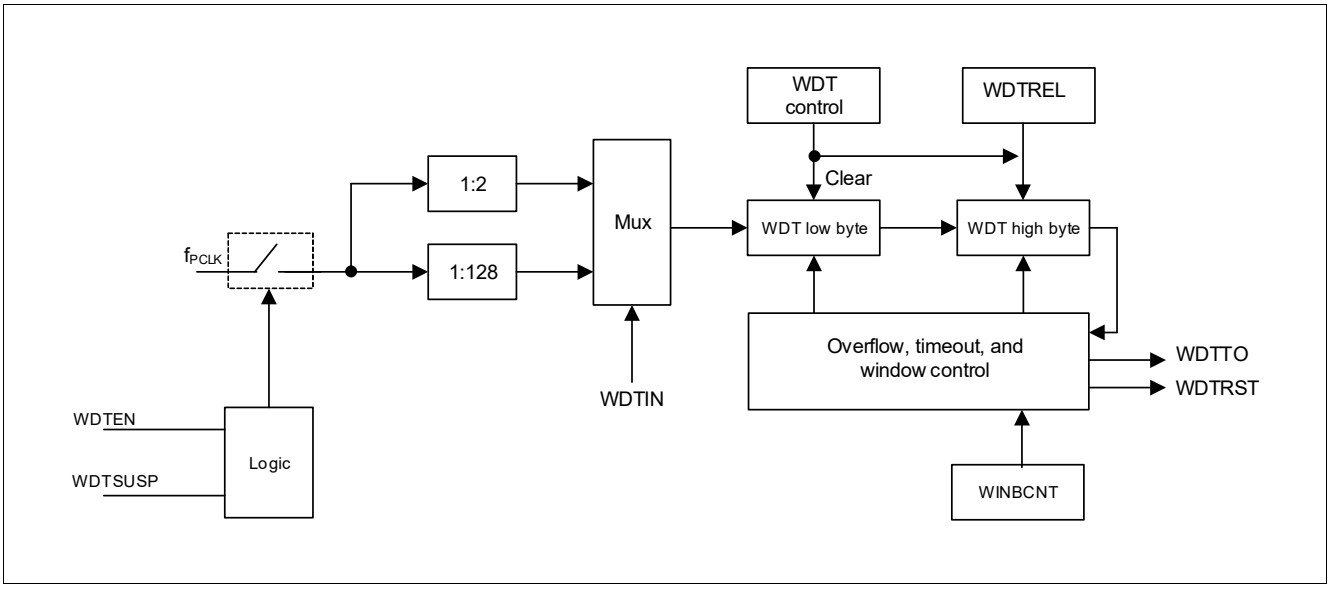


Figure 32 WDT block diagram

If the watchdog timer is enabled by setting the WDTEN bit to 1, the timer is set to a user-defined start value and begins counting up. It must be serviced before the counter overflows. Servicing is performed through refreshing the timer. This reloads the timer with the start value, and normal operation continues.

If the WDT is not serviced before the timer overflows, a system malfunction is assumed and normal operation is terminated. A Watchdog Timer NMI request (WDTTO) is asserted, and the prewarning period begins, which lasts for 30_H counts. During the prewarning period, watchdog timer refreshes are ignored and the watchdog timer cannot be disabled. A reset (WDTRST) of the TLE987x is imminent and can no longer be stopped. If a refresh and an overflow happen at the same time, the watchdog timer does not start the prewarning period.

The watchdog timer must be serviced periodically so that its count value will not overflow. Servicing the watchdog timer clears the low byte and reloads the high byte with the preset value from the WDTREL bit field. Servicing the watchdog timer also clears the WDTRS bit.

The watchdog timer has a programmable window that prevents the watchdog timer from being refreshed while it counts up. A refresh that happens within this window causes the watchdog timer to activate WDTRST. The window extends from from 0000_H to WDTWINB × 2^8. This feature can be enabled by setting WINBEN.

After having been serviced, the watchdog timer continues to count up from the value WDTREL × 2^8. The time period for an overflow of the watchdog timer is programmable in two ways:

- The input frequency of the watchdog timer can be selected via the WDTIN bit in the WDTCON register to be either $f_{PCLK}/2$ or $f_{PCLK}/128$.
- The reload value WDTREL for the high byte of WDT can be programmed in the WDTREL register.

The period PWDT between servicing the Watchdog Timer and the next overflow is determined by this equation:

$$P_{WDT} = \frac{2^{(1+WDTIN\times 6)} \times (2^{16} - WDTREL \times 2^{8})}{f_{PCLK}} \quad (6.5)$$

If the refresh window feature of the watchdog timer is enabled, the period P_{WDT} between servicing the watchdog timer and the next overflow is shortened if WDTWINB is greater than WDTREL. See also **Figure 33**. This period can be calculated using the same equation by replacing WDTREL with WDTWINB. In order for this feature to be useful, WDTWINB cannot be smaller than WDTREL.

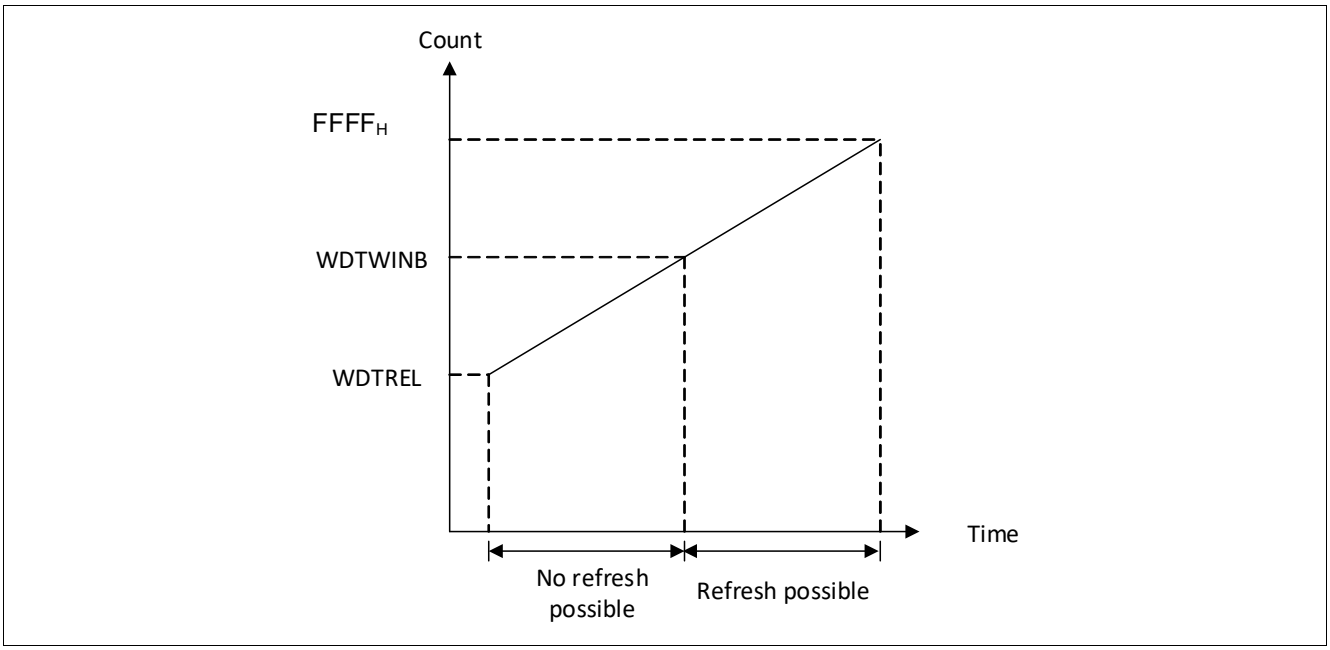


Figure 33 Watchdog timer timing diagram

The following table lists the possible ranges for the watchdog intervals which can be achieved using a given module clock. Some numbers are rounded to three significant digits.

Table 37 Watchdog interval ranges

Reload value in WDTREL	Prescaler for f_{PCLK}					
	2 (WDTIN = 0)			128 (WDTIN = 1)		
	40 MHz	20 MHz	13.3 MHz	40 MHz	20 MHz	13.3 MHz
FF_H	12.8 µs	25.6 µs	38.4 µs	0.82 ms	1.64 ms	2.46 ms
$7F_H$	1.65 ms	3.30 ms	4.95 ms	106 ms	211 ms	317 ms
00_H	3.28 ms	6.55 ms	9.83 ms	210 ms	419 ms	629 ms

Notes

1. *For safety reasons, we recommend that you rewrite WDTCON each time before the watchdog timer is serviced.*
2. *The watchdog timer can be suspended when the core is halted and the Debug-Suspend signal is activated, provided the respective suspend bit, WDTSUSP in SFR SCU_MODSUSP, is set. See **Chapter 6.9**.*

6.10.2 Register description

The current count value of the watchdog timer is contained in the Watchdog Timer register WDT, which is a non-bit-addressable read-only register. The operation of the Watchdog Timer is controlled by the bit-addressable Watchdog Timer Control register WDTCON. The WDTREL register specifies the reload value for the high byte of the timer. WDTWINB specifies the count values for the watchdog timer window.

WDTREL
Watchdog Timer Reload Register **(078_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
WDTREL							
rw							

Field	Bits	Type	Description
WDTREL	7:0	rw	**Watchdog Timer Reload Value** The value which is written to the WDT's high byte when it is reloaded.

WDTCON
Watchdog Timer Control Register **(050_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		**WINBEN**	**WDTPR**	**RES**	**WDTEN**	**WDTRS**	**WDTIN**
r		rw	rh	r	rw	rwh	rw

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.
WINBEN	5	rw	**Watchdog Window-Boundary Enable** 0_B Watchdog window feature is disabled. (default) 1_B Watchdog window feature is enabled.
WDTPR	4	rh	**Watchdog Prewarning Mode Flag** 0_B Normal mode (default after reset) 1_B The watchdog is operating in prewarning mode This bit is set to 1 when a watchdog error is detected. The watchdog timer has issued an NMI trap and is in prewarning mode. The device will be reset after the prewarning period has expired.
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.

Field	Bits	Type	Description
WDTEN	2	rw	**WDT Enable** 0_B WDT is disabled 1_B WDT is enabled WDTEN is a protected bit. If the protection scheme is activated, this bit cannot be written directly. For more information on the protection scheme, see **Chapter 6.12**. *Note: Clearing the WDTEN bit during the prewarning period (WDTPR = 1) has no effect.*
WDTRS	1	rwh	**WDT Refresh Start** Active high. Set to start refresh operation on the watchdog timer. Cleared automatically by hardware after it is set by software.
WDTIN	0	rw	**Watchdog Timer Input Frequency Selection** 0_B Input frequency is $f_{PCLK}/2$ 1_B Input frequency is $f_{PCLK}/128$

WDTL
Watchdog Timer, Low Byte **(080_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
WDT							
rh							

Field	Bits	Type	Description
WDT	7:0	rh	**Watchdog Timer Current Value**

WDTH
Watchdog Timer, High Byte **(084_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
WDT							
rh							

Field	Bits	Type	Description
WDT	7:0	rh	**Watchdog Timer Current Value**

WDTWINB
Watchdog Window-Boundary Count **($07C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
WDTWINB							
rw							

Field	Bits	Type	Description
WDTWINB	7:0	rw	**Watchdog Window-Boundary Count Value** This value is programmable. Within the specified window range from 0000_H to (WDTWINB × 2^8), the WDT cannot be refreshed. Any refresh attempt causes a WDT reset to be asserted. WDTWINB is compared with WDTH.

6.11 Error detection and correction control for memory

This section describes the registers used controlling memory error detection and correction (both in RAM and NVM).

6.11.1 Error Detection and Correction Control register

The EDCCON register determines whether double-bit ECC errors when reading RAM or NVM generate an NMI.

EDCCON
Error Detection and Correction Control Register ($0D4_H$) **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES				**RES**	**NVMIE**	**RES**	**RIE**
r				r	rw	r	rw

Field	Bits	Type	Description
RES	7:4	r	**Reserved** Returns 0 if read; should be written with 0.
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
NVMIE	2	rw	**NVM Double Bit ECC Error Interrupt Enable** 0_B No NMI is generated when a double-bit ECC error occurs when reading NVM. 1_B An NMI is generated when a double-bit ECC error occurs when reading NVM.
RES	1	r	**Reserved** Returns 0 if read; should be written with 0.
RIE	0	rw	**RAM Double Bit ECC Error Interrupt Enable** 0_B No NMI is generated when a double-bit ECC error occurs when reading RAM. 1_B An NMI is generated when a double-bit ECC error occurs when reading RAM.

6.11.2 Error Detection and Correction Status register

The EDCSTAT register contains flags that indicate whether ECC errors have occurred while reading from RAM or NVM.

The EDCSTAT register is reset by RESET_TYPE_4.

EDCSTAT
Error Detection and Correction Status Register, ($0D8_H$) **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**RES**	**RES**	**RSBE**	**RES**	**NVMDBE**	**RES**	**RDBE**
r	r	r	r	r	r	r	r

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.
RES	5	r	**Reserved** Returns 0 if read; should be written with 0.
RSBE	4	r	**RAM Single Bit Error** This bit is set by hardware and can be cleared only by software. 0_B No single-bit error has occurred while reading from RAM. 1_B A single-bit error has occurred while reading from RAM.
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
NVMDBE	2	r	**NVM Double Bit Error** This bit is set by hardware and can be cleared only by software. 0_B No double-bit error has occurred while reading from NVM. 1_B A double-bit error has occurred while reading from NVM.
RES	1	r	**Reserved** Returns 0 if read; should be written with 0.
RDBE	0	r	**RAM Double Bit Error** This bit is set by hardware and can be cleared only by software. 0_B No double bit error has occurred while reading from RAM. 1_B A double bit error has occurred while reading from RAM.

EDCSCLR

Error Detection and Correction Status Clear Register($10C_H$) **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**RES**	**RES**	**RSBEC**	**RES**	**NVMDBEC**	**RES**	**RDBEC**
r	r	r	w	r	w	r	w

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.
RES	5	r	**Reserved** Returns 0 if read; should be written with 0.
RSBEC	4	w	**RAM Single Bit Error Clear** This bit is set by software and can be cleared only by hardware. 0_B A single-bit RAM error indication has not been cleared. 1_B A single-bit RAM error indication has been cleared.
RES	3	r	**Reserved** Returns 0 if read; should be written with 0.
NVMDBEC	2	w	**NVM Double Bit Error Clear** This bit is set by software and can be cleared only by hardware. 0_B A double-bit NVM error indication has not been cleared. 1_B A double-bit NVM error indication has been cleared.
RES	1	r	**Reserved** Returns 0 if read; should be written with 0.
RDBEC	0	w	**RAM Double Bit Error Clear** This bit is set by software and can be cleared only by hardware. 0_B A double-bit RAM error indication has not been cleared. 1_B A double-bit RAM error indication has been cleared.

6.12 Miscellaneous control

This module consists of registers for the bit-protection scheme and general system-control SFRs.

6.12.1 Bit-protection register

The protection scheme prevents the software from writing to specific bits (the so-called protected bits) without unlocking the PASSWD register. When the MODE bit field is set to 11_B, writing 10011_B to the PASS bit field opens write access to all protected bits. Writing 10101_B to PASS closes that write access again. Note that write access to protected bits closes again automatically after a maximum of 32 clock cycles, even if the “close access” value is not written to PASS. Anytime the “open access” value is written to PASS; the 32-cycle count restarts.

Note: Because the write access to a register unlocked by the access to PASSWD is granted only for 32 clock cycles, it is recommended to disable interrupts before and re-enable them afterwards. If an interrupt is issued right after the register write access has been opened, the 32 clock cycles may already have passed when the interrupt returns, and the write access to the desired register is blocked again. Any write access to a locked register that holds only PASSWD-protected bits, causes a hard fault exception.

PASSWD
Password Register **($0AC_H$)** **Reset Value: 07_H**

7	6	5	4	3	2	1	0
PASS					**PROTECT_S**	**MODE**	
wh					rh	rw	

Field	Bits	Type	Description
PASS	7:3	wh	**Password Bits** The protection scheme only recognizes three patterns. 11000_B Writing to the MODE bit field is enabled. 10011_B Write access to protected bits is opened. 10101_B Write access to protected bits is closed.
PROTECT_S	2	rh	**Bit-Protection Signal Status Bit** This bit indicates the protection status. 0_B Software is able to write to protected bits. 1_B Software is unable to write to protected bits.
MODE	1:0	rw	**Bit-Protection Scheme Control Bit** 00_B Protection scheme disabled 11_B Protection scheme enabled (default) Others: Protection scheme enabled These two bits cannot be written without unlocking them first. To change their value between 11_B and 00_B, 11000_B must be written to the PASS bit field first. Only then will writes to MODE be processed.

The PASSWD register and the registers which contain protected bits are located in page 2 of the SCU SFR address map. The list of protected bits is shown in **Table 38**.

Table 38 List of protected bits

Register	Bit field
SYSCON0	NVMCLKFAC
	SYSCLKSEL
OSC_CON	OSCSS
	XPD
OSC_CON	OSCTRIM8
PLL_CON	NDIV
CMCON1	K1DIV
	K2DIV
CMCON2	PBA0CLKREL
PMCON0	SL
	PD
	SD
WDTCON	WDTEN
APCLK_CTRL1	CPCLK_DIV
	CPCLK_SEL
	BGCLK_DIV
	BGCLK_SEL
	T3CLK_SEL
TRIM_DRVx	All bits

6.12.2 System control and status registers

System startup status register

SYS_STRTUP_STS contains the main system control and status bits.

The System Startup Status register provides information about the system initialization with the user-programmable 100 TP page at startup. This register is written by firmware. The user application needs to check the SYS_STRTUP_SYS.MRAMINITSTS in order to verify the consistency of the data flash mapping. Only if no failure is signaled can the data flash write and erase operations be safely performed. In case of failure a reset can be issued in order to start the service algorithm to try to fix the integrity issue inside the data flash. If the MRAMINISTS is still flagged afterwards, the data flash sector has to be reinitialized by performing a SECTOR_ERASE.

The results of the check performed by the service algorithm are available in the MEMSTAT register.

SYS_STRTUP_STS
System Startup Status Register **(074_H)** **Reset Value: 40_H**

7	6	5	4	3	2	1	0
RES					**PG100TP_CHKS_ERR**	**MRAMINITSTS**	**INIT_FAIL**
		r			rwp	rwp	rwp

Field	Bits	Type	Description
RES	7:3	r	**Reserved** This bit field is always read as zero.
PG100TP_CHKS_ERR	2	rwp	**100 TP Page Checksum Error** Initializing the trimming parameters from NVM failed. Default values from the boot ROM are used because of a checksum error.
MRAMINITSTS	1	rwp	**Map RAM Initialization Status** This bit reflects the status of the map RAM initialization. 0: No fail – the map RAM was successfully initialized. 1: Fail – the map RAM could not be initialized.
INIT_FAIL	0	rwp	**Initialization at startup failed** This bit is a logical OR between PLL_LOCK failure, map RAM initialization failure, and the trimming values checksum error. 0: No initialization error during startup. 1: Initialization error during startup.

This register is reset by RESET_TYPE_4. It reflects the NVM protection status. It is written by firmware. This register can only be written if write access is first opened by writing to the PASSWD register.

The NVM_PROT_STS register is initialized to $0F_H$ by the BootROM code during the boot process.

NVM_PROT_STS
NVM Protection Status Register **($0E0_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES				NVMPROTSTSL_3	NVMPROTSTSL_2	NVMPROTSTSL_1	NVMPROTSTSL_0
r				rwp	rwp	rwp	rwp

Field	Bits	Type	Description
RES	7:4	r	**Reserved** This bit field is always read as zero.
NVMPROTSTSL_3	3	rwp	**NVM Protection: Code Flash Read Protection Status** 0_B Protection is set 1_B Protection is not set
NVMPROTSTSL_2	2	rwp	**NVM Protection: Data Flash Read Protection Status** 0_B Protection is set 1_B Protection is not set
NVMPROTSTSL_1	1	rwp	**NVM Protection: Code Flash Write Protection Status** 0_B Protection is set 1_B Protection is not set
NVMPROTSTSL_0	0	rwp	**NVM Protection: Data Flash Write Protection Status** 0_B Protection is set 1_B Protection is not set

Memory access status register

The MEM_ACC_STS register is reset by RESET_TYPE_4. It reflects the memory access status of the entire system memory. Software can only clear this register.

MEM_ACC_STS
Memory Access Status Register **($0E4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**RES**	**RES**	**ROM_PROT_ERR**	**RES**	**RES**	**NVM_ADDR_ERR**	**NVM_PROT_ERR**
r	r	r	rwh	r	r	rwh	rwh

Field	Bits	Type	Description
RES	7	r	**Reserved** This bit field is always read as zero.
RES	6	r	**Reserved** This bit field is always read as zero.
RES	5	r	**Reserved** This bit field is always read as zero.
ROM_PROT_ERR	4	rwh	**ROM Access Protection** This bit is set when an attempt is made to read the BootROM code area. 0_B No protection error 1_B Protection error
RES	3	r	**Reserved** This bit field is always read as zero.
RES	2	r	**Reserved** This bit field is always read as zero.
NVM_ADDR_ERR	1	rwh	**NVM Address Protection** This is set if an unmapped access to the data flash is performed. 0_B No Protection error 1_B Protection error
NVM_PROT_ERR	0	rwh	**NVM Access Protection** This bit is set when an attempt is made to access a read-protected NVM area, such as code flash or data flash. 0_B No protection error 1_B Protection error

Identity register

The Identity register identifies the product and the version.

ID
Identity Register **($0A8_H$)** **Reset Value: $8x_H$**

7	6	5	4	3	2	1	0
PRODID					VERID		
r					r		

Field	Bits	Type	Description
PRODID	7:3	r	**Product ID** 10000_B
VERID	2:0	r	**Version ID** The stepping code of the device. 001_B A-stepping 010_B B-stepping

Memory status register

The Memory Status register can be used in two ways. When the boot ROM startup after a reset is complete, the register contains the NVM initialization status. Subsequently, the register can be used by the user code to store the status of the NVM program and the emergency program operation status.

The following text describes the function of the register to indicate the NVM initialization status. (For information on its use to store NVM program and emergency operation statuses, see below the register table.)

The MEMSTAT register provides the return value of the service algorithm, which might get executed during startup. Because of the dual function of the MEMSTAT register, it is recommended to clear it after evaluating it and before any NVM firmware function is called. If the MEMSTAT register holds the value 0x00, the service algorithm was not executed at all. Otherwise, the SASTATUS bit field contains information about the result of the corrective actions performed by the service algorithm.

If the service algorithm was executed successfully (SASTATUS=0b01), you should assume that it did fix a data flash integrity issue by erasing an inconsistent data flash page. You should therefore check the availability of the expected data flash pages. If an expected page was erased by the service algorithm due to a failure in the data flash page control information, reading that page would trigger an NMIMAP error. You can use this information to reconstruct the content of the missing data flash page.

MEMSTAT
Memory Status Register **($0DC_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
SASTATUS		SECTORINFO					
rw		rw					

Field	Bits	Type	Description
SASTATUS	7:6	rw	**Service Algorithm Status** 00_B Depending on SECTORINFO, there are two possible outcomes: For SECTORINFO = 00_H, NVM initialization is successful and no service algorithm is executed. For SECTORINFO values other than 00_H, the service algorithm was successfully executed and up to one map error was fixed. 01_B The service algorithm was successfully executed and up to one map error was fixed. 10_B The execution of the service algorithm was not successful. A map error exists in one sector. 11_B The execution of the service algorithm was not successful. At least one sector failed. (This also includes cases where a sector is repaired but another sector is still failing.)

Field	Bits	Type	Description
SECTORINFO	5:0	rw	**Sector Information** The address of the data flash sector. The service algorithm stores this information in this bit field after every execution, regardless of the execution status. The stored values correspond to different flash sizes: 00_H Service algorithm was not executed, mapping consistent 09_H 36 KB flash derivative $0C_H$ 48 KB flash derivative 10_H 64 KB flash derivative 20_H 128 KB flash derivative 22_H 256 KB flash derivative

The MEMSTAT register is reset by RESET_TYPE_3.

Note: If MEMSTAT[7] is 1 the service algorithm was not able to resolve an existing data integrity issue in the data flash. It is recommended to perform a reset of the device in order to execute the service algorithm again. If this second attempt also fails with MEMSTAT[7] = 1, reinitializing the data flash by performing a SECTOR_ERASE is recommended.

The following text describes the function of the register to store NVM program and emergency operation statuses. (For information on its use to indicate the NVM initialization status after startup, see above.)

MEMSTAT
Memory Status Register **($0DC_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES						**EMPROP**	**NVMPROP**
rw						rw	rw

Field	Bits	Type	Description
RES	7:2	rw	**Reserved Bits** Should be written with 0.
EMPROP	1	rw	**Emergency Program Operation Status Bit** The status of the emergency program operation. 0_B No emergency program operation is requested. 1_B Emergency program operation is requested.
NVMPROP	0	rw	**NVM Program Operation Status Bit** The status of the NVM program operation. 0_B No NVM program operation has been started. 1_B NVM program operation has been started.

6.13 Register access notes

Some registers in the SCU module are clocked with MI_CLK. Write operations to them are delayed by one MI_CLK cycle. This issue applies to the **APCLK1**, **APCLK2**, **EDCSCLR**, and **NMICLR** registers.

Special considerations apply to clearing interrupt status flags. Under certain coding constellations, an extra write access to the interrupt status flag is necessary to add the extra MI_CLK cycle.

There are two different use cases when clearing interrupt status flags:

1. Between the code statements clearing an interrupt flag and reading that interrupt flag, other code is present.
2. There is no other code between the statements clearing and reading the interrupt status flag.

Use case 1: No extra write attempt to the interrupt status flag required

```
/* Clearing the RAM double bit error interrupt status flag */
SCU->EDCSCLR.bit.RDBEC = 1;
/* Some other code here */
...
/* Reading the RAM double bit error interrupt status flag */
if ( SCU->EDCSTAT.bit.RDBE == 1 )
{
 /* Some additional code here */
}
```

In this example, sufficient time between the commands to clear the register and to read the register is ensured because at least one CPU instruction is executed between the two.

Use case 2: An extra write attempt to the interrupt clear register required

```
/* Clearing the RAM double bit error interrupt status flag */
SCU->EDCSCLR.bit.RDBEC = 1;
/* Insert another command to clear the status flag in order to add an extra MI_CLK
cycle */
SCU->EDCSCLR.bit.RDBEC = 0;
/* Reading the RAM double bit error interrupt status flag */
if ( SCU->EDCSTAT.bit.RDBE == 1 )
{
 /* Some additional code here */
}
```

In this example, an additional write command to clear the register is inserted between the commands to clear the register and to read the register. This adds another MI_CLK cycle, ensuring that the first interrupt status flag write attempt has reached the interrupt status register before it is read again. Without this extra clear attempt, the following interrupt status flag read might still reflect the previous content of the flag (before it was cleared).

In principle, any instruction can be used to add the required MI_CLK cycle. However, the instruction should not cause unwanted side-effects and not be eliminated by compiler optimization. In most cases, declaring the register addresses in question as volatile to the compiler and then writing zero to the clear register achieves this.

We also recommend to write a zero in the second write command because that will not cause functional changes to the register.

7 System control unit – power modules (SCU-PM)

7.1 Features

- Clock watchdog unit (CWU): Supervises all clocks with NMI signaling relevant to power modules.
- Interrupt control unit (ICU): All interrupt flags and status flags with system relevance.
- Power control unit (PCU): Takes over control when device enters and exits Sleep and Stop mode.
- External watchdog (WDT1): Independent system watchdog for monitoring system activity.

7.2 Introduction

7.2.1 Block diagram

The system control unit of the power modules consists of the submodules in the figure shown below:

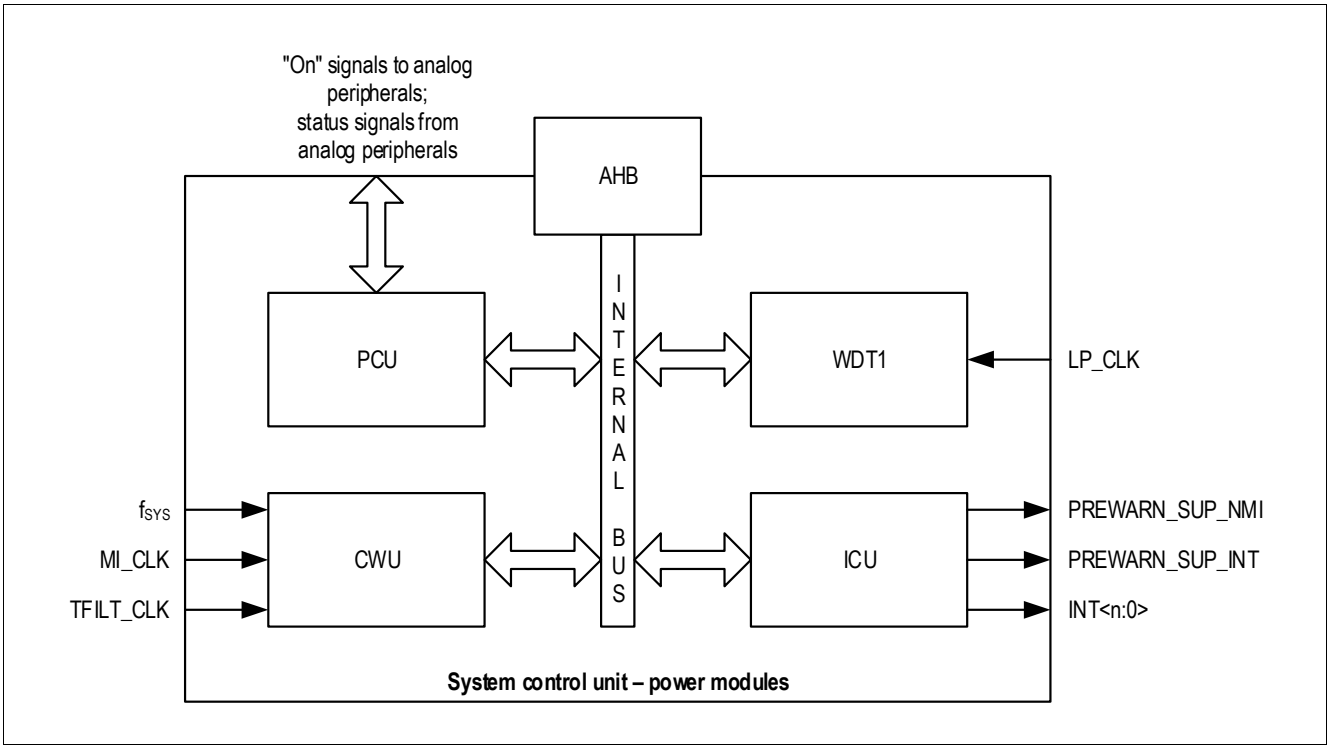


Figure 34 Block diagram of system control unit – power modules

AHB (Advanced High-performance Bus)

CWU (clock watchdog unit)

- f_{sys}: System frequency
- MI_CLK: Measurement interface clock (analog clock), derived from f_{sys} using division factors 1/2/3/4
- TFILT_CLK: Clock used for digital filters, derived from f_{sys} using configurable division factors

WDT1 (system watchdog)

- LP_CLK Clock source for all PMU submodules and WDT1

ICU (interrupt control unit)

- PREWARN_SUP_NMI Supply prewarning NMI request
- PREWARN_SUP_INT Supply prewarning interrupt
- Grouping of peripheral interrupts for external interrupt nodes:
  - Grouping single peripheral interrupts for interrupt node INT<2> (measurement unit (MU))
  - Grouping single peripheral interrupts for interrupt node INT<3> (ADC1-VAREF)
  - Grouping single peripheral interrupts for interrupt node INT<10> (UART1-LIN transceiver)
  - Grouping single peripheral interrupts for interrupt node INT<14> (bridge driver)

7.3 Clock Watchdog Unit (CWU)

There are two clock watchdogs available. One main purpose of them, is to monitor the derived switched capacitor clocks, which are used for analog module operation. If the clocks are not in the required range, a proper functionality of those modules is not given.

The following chapter describes the functionality and the configuration possibilities of these clock watchdogs.

7.3.1 Fail Safe Functionality of Clock Generation Unit (Clock Watchdog)

The Clock Generation Unit provides also fail safe functionalities, which are related to the input clock, the generated clocks and the clock settings. Those are:

- **MI_CLK** and **TFILT_CLK** are out of Range: MI Clock settings for f_{sys}, MI_CLK and TFILT_CLK Clock settings are out of required range and as a result the analog functionalities cannot be guaranteed. This failure triggers the clock watchdog NMI. The current status can be seen in the corresponding registers APCLK1 (in SCU) for the **MI_CLK** and APCLK2 (in SCU) for the **TFILT_CLK**.
- **Loss of clock:** When there is a loss of clock in the system, there is no possibility for the software to react upon this situation, like to enter a fail safe mode or switch to another backup clock source. For this purpose there is a clock watchdog implemented in the system which monitors the f_{sys} and in case of this emergency situation, disables all critical system functions, which are:
  - Bridge Driver and Charge Pump
  - LIN

As shown in **Figure 35** all analog clocks are derived from **MI_CLK**. This clock structure requires to place a monitor on this clock, because f_{sys} and therefore **MI_CLK** are adjustable in a wide range (see also Chapter **System Control Unit - CGU**). As an important clock, also the TFILT_CLK is monitored by a clock watchdog. This clock watchdogs have an adjustable lower and upper limits including hysteresis. The placement of the clock watchdogs in the clock structure is sketched below:

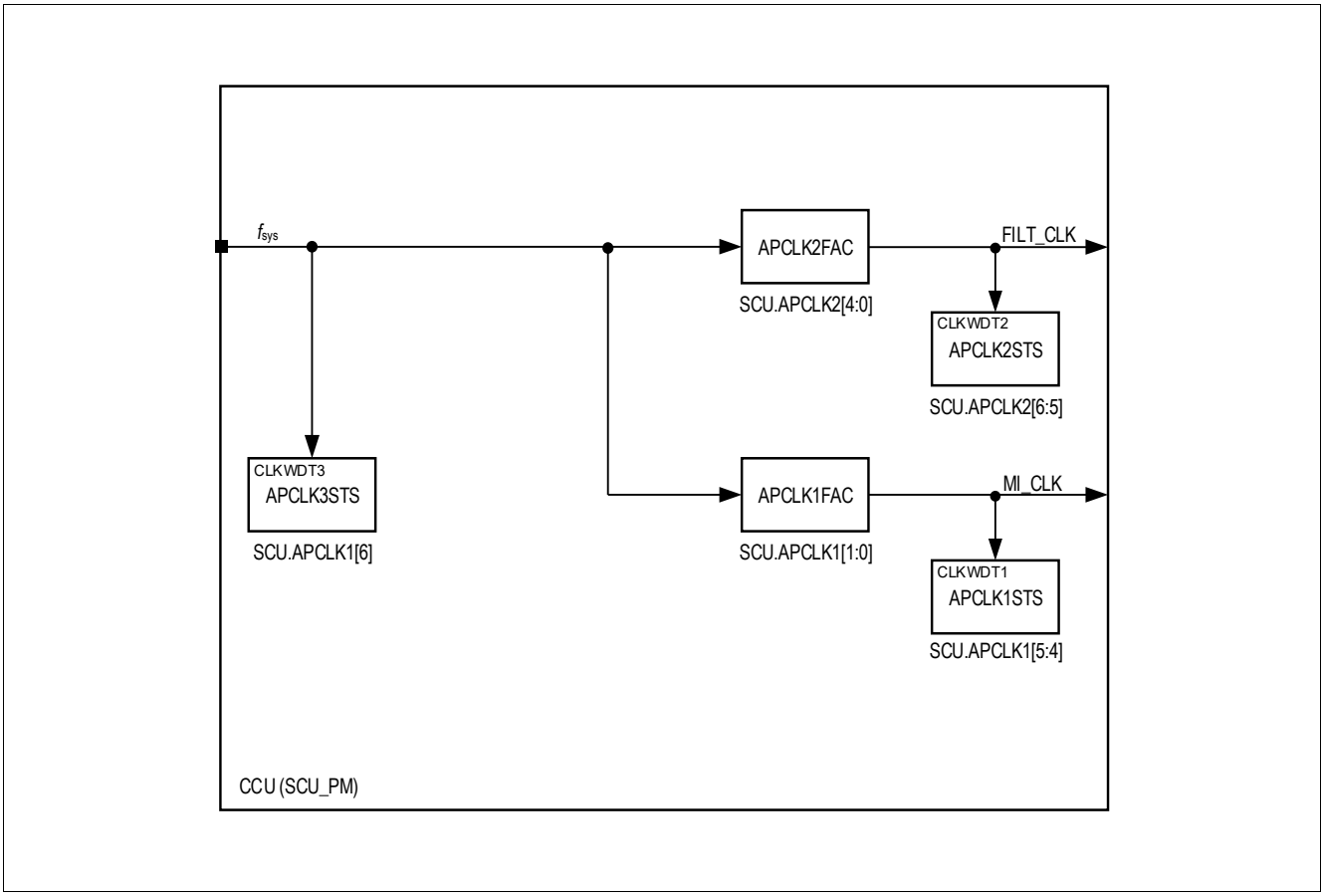


Figure 35 Block diagram of CGU including Clock Watchdogs

7.3.1.1 Functional Description of Clock Watchdog Module

The clock watchdog module consists of a counter. This counter monitors the number of system clocks within a defined time window. The duration of the time window is defined by a clock (**LP_CLK**), which is independent from the monitored system clock (**MI_CLK**). If the required number of clock cycles is not reached within this time window an clock watchdog NMI will be issued.

In case the clock watchdog NMI will be issued, indicating that the clock is not within the required frequency range, then the user has different options to overcome this situation:

- stay on mi_clk but reconfigure PLL to re-gain the required clock frequency. This would be the most time consuming measure to avoid emergency shutdown of the above listed modules.
- switch to divider factors 2, 3 and 4 to try to come back to specified frequency range.
- switch to LP_CLK, which also can be divided by factor 2, 3 and 4. This is the fastest option which allows the user to operate with a well defined backup clock rate. After this has been done the user can start investigating the root cause of the issued clock watchdog NMI, while operating on **LP_CLK**.

The register chapter below includes all necessary flags for setting up the analog module clock and monitoring its status during operation.

7.3.2 Clock Generation Unit Register

The analog module clock generation unit is fully controllable by the register described in this chapter.

Table 39 shows the module base addresses.

Table 39 Register Address Space

Module	Base Address	End Address	Note
SCUPM	50006000_H	$50006FFF_H$	SCU_PM

Table 40 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Clock Generation Unit Register			
AMCLK_FREQ_STS	Analog Module Clock Frequency Status Register	00_H	00xx xxxx $00xx\ xxxx_B$
AMCLK_CTRL	Analog Module Clock Control Register	04_H	$0000\ 0001_H$
AMCLK_TH_HYS	Analog Module Clock Limit Register	$0C_H$	$D4E1\ 94B3_H$
STCALIB	System Tick Calibration Register	$6C_H$	$0000\ 0000_H$

The registers are addressed wordwise.

Analog Clock Frequency Status Register

The register is reset by RESET_TYPE_4.

AMCLK_FREQ_STS

Analog Module Clock Frequency Status Register (00_H) **Reset Value: $00xx\ xxxx\ 00xx\ xxxx_B$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		AMCLK2_FREQ						RES		AMCLK1_FREQ					
r		r						r		r					

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
RES	15:14	r	**Reserved** Always read as 0
AMCLK2_FREQ	13:8	r	**Current frequency of Analog Module Clock 2 (TFILT_CLK)** AMCLK2_FREQ = (256 * TFILT_CLK) / LP_CLK
RES	7:6	r	**Reserved** Always read as 0
AMCLK1_FREQ	5:0	r	**Current frequency of Analog Module Clock System Clock (MI_CLK)** AMCLK1_FREQ = (32 * MI_CLK) / LP_CLK

Analog Clock Frequency Control Register

The register is reset by RESET_TYPE_4.

AMCLK_CTRL
Analog Module Clock Control Register **(04_H)** **Reset Value: $0000\ 0001_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES															CLKWDT_PD_N
r															rw

Field	Bits	Type	Description
RES	31:1	r	**Reserved** Always read as 0
CLKWDT_PD_N	0	rw	**Clock Watchdog Powerdown** 0_B **DISABLE**, Clock Watchdog disabled 1_B **ENABLE**, Clock Watchdog enabled

Analog Module Clock Limit Register

The register is reset by RESET_TYPE_4.

AMCLK_TH_HYS
Analog Module Clock Limit Register **($0C_H$)** **Reset Value: $D4E1\ 94B3_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AMCLK2_LOW_HYS		AMCLK2_LOW_TH						AMCLK2_UP_HYS		AMCLK2_UP_TH					
rw		rw						rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AMCLK1_LOW_HYS		AMCLK1_LOW_TH						AMCLK1_UP_HYS		AMCLK1_UP_TH					
rw		rw						rw		rw					

Field	Bits	Type	Description
AMCLK2_LOW_HYS	31:30	rw	**Analog Module Clock 2 (TFILT_CLK) Lower Hysteresis**
AMCLK2_LOW_TH	29:24	rw	**Analog Module Clock 2 (TFILT_CLK) Lower Limit Threshold** TFILT_CLK_LOW_TH = (AMCLK2_LOW_TH * LP_CLK) / 256
AMCLK2_UP_HYS	23:22	rw	**Analog Module Clock 2 (TFILT_CLK) Upper Hysteresis**
AMCLK2_UP_TH	21:16	rw	**Analog Module Clock 2 (TFILT_CLK) Upper Limit Threshold** TFILT_CLK_UP_TH = (AMCLK2_UP_TH * LP_CLK) / 256
AMCLK1_LOW_HYS	15:14	rw	**Analog Module Clock 1 (MI_CLK) Lower Hysteresis**
AMCLK1_LOW_TH	13:8	rw	**Analog Module Clock 1 (MI_CLK) Lower Limit Threshold** MI_CLK_LOW_TH = (AMCLK1_LOW_TH * LP_CLK) / 32
AMCLK1_UP_HYS	7:6	rw	**Analog Module Clock 1 (MI_CLK) Upper Hysteresis**
AMCLK1_UP_TH	5:0	rw	**Analog Module Clock 1 (MI_CLK) Upper Limit Threshold** MI_CLK_UP_TH = (AMCLK1_UP_TH * LP_CLK) / 32

System Tick Calibration Register

The register is reset by RESET_TYPE_4.

STCALIB
System Tick Calibration Register **($6C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES						STCALIB									
r						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STCALIB															
rw															

Field	Bits	Type	Description
RES	31:26	r	**Reserved** Always read as 0
STCALIB	25:0	rw	**System Tick Calibration** [25]: Noref [24] Skew [23:0] Reload value to use for 10ms (100 Hz) timing STCALIB[23:0] = HCLK (in Hz) / 100 Hz - 1, e.g. 0x7A11F

7.4 Interrupt Control Unit (ICU)

The Sub-block Interrupt Control Unit (ICU) of the System Control Unit - Power Modules (SCU_PM) is responsible for controlling and generating all analog peripheral relevant interrupts. Those analog interrupts are presented to the NVIC nodes 13-24 and NMI. Those are:

- **PREWARN_SUP_NMI:** combines all supply relevant interrupts to NMI.
- **Analog Module Interrupts:** combines all analog modules related interrupts.

The following two chapters describe the structure of the interrupt nodes.

7.4.1 Structure of PREWARN_SUP_NMI

This interrupt groups all system supply relevant interrupts. They can be divided into two groups:

- **voltages monitored by the Measurement Unit.** The supply voltages VS, VDDP and VDDC are monitored by the Measurement Unit. The Measurement Unit can be seen as an independent monitoring instance of the PMU, with independent reference voltage and supply voltage.
- **voltages monitored by measurement functions of the PMU:** The PMU itself is checking its output voltages. Here failures due to undervoltage (overload), overvoltage and overcurrent can be detected.

The following figure shows the structure of the **PREWARN_SUP**:

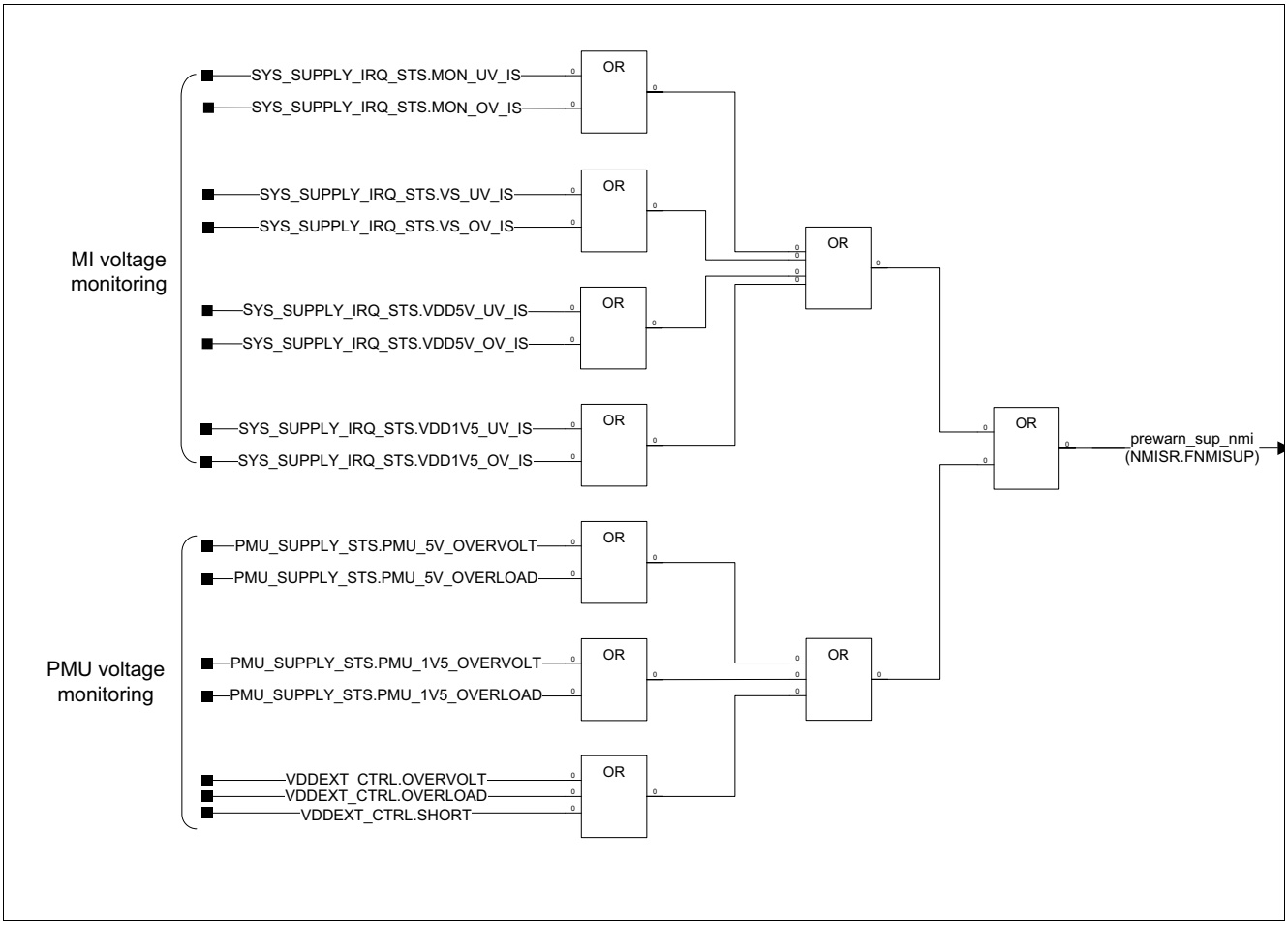


Figure 36 Structure of PREWARN_SUP

All PREWARN_SUP related flags are grouped in register **SYS_SUPPLY_IRQ_STS**. All measurement interface related flags are edge triggered. Therefore each IRQ_STS register has also an STS register where the current supply status can be monitored.

7.4.2 Interrupt Control Unit Status Register

All analog modules interrupt functionality is described in this chapter.

Table 41 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Interrupt Control Unit Status Register, Interrupt Control Unit Status Overview Register			
SYS_ISCLR	System Interrupt Status Clear	14_H	$0000\ 0000_H$
SYS_IS	System Interrupt Status	18_H	$0000\ 0000_H$
SYS_SUPPLY_IRQ_STS	System Supply Interrupt Status	$1C_H$	$0000\ 0000_H$
SYS_SUPPLY_IRQ_CLR	System Supply Interrupt Status Clear	24_H	$0000\ 0000_H$
BDRV_ISCLR	Bridge Driver Interrupt Status Clear	54_H	$0000\ 0000_H$
BDRV_IS	Bridge Driver Interrupt Status	58_H	$0000\ 0000_H$
Interrupt Control Unit Status Register, Interrupt Control Unit - Interrupt Enable Register			
SYS_SUPPLY_IRQ_CTRL	System Supply Interrupt Control	20_H	$0000\ 00FF_H$
SYS_IRQ_CTRL	System Interrupt Control	28_H	$0000\ 0000_H$
BDRV_IRQ_CTRL	Bridge Driver Interrupt Control	$5C_H$	$0000\ 0000_H$

The registers are addressed wordwise.

7.4.2.1 Interrupt Control Unit Status Overview Register

Due to the large variety of diagnosis possibilities of TLE987x, the system offers several overview registers, to help the user finding the right source of interrupt. Those registers are described in this sub-chapter.

Overview Register, Switches Interrupt Status Register and System Supply Interrupt Status Register

- **SYS_SUPPLY_IRQ_STS**: Flags for Under- and Overvoltage detection for all system relevant supplies. These Interrupts are edge triggered Interrupts.
- **SYS_IS**: Interrupts for Analog Modules.

System Interrupt Status

The register is reset by RESET_TYPE_4.

SYS_IS

System Interrupt Status **(18_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		PHW_ZCHI_STS	PHW_ZCLOW_STS	PHV_ZCHI_STS	PHV_ZCLOW_STS	PHU_ZCHI_STS	PHU_ZCLOW_STS	ADC4_EOC_IS	ADC3_EOC_IS	PHW_ZCHI_IS	PHW_ZCLOW_IS	PHV_ZCHI_IS	PHV_ZCLOW_IS	PHU_ZCHI_IS	PHU_ZCLOW_IS
r		r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2_ESM_IS	VREF5V_OVL_IS	VREF5V_UPTH_IS	VREF5V_LOWTH_IS	REFBG_UPTHWARN_IS	REFBG_LOTHWARN_IS	SYS_OT_IS	SYS_OTWARN_IS	PMU_OT_IS	PMU_OTWARN_IS	RES			LIN_TMOUT_IS	LIN_OT_IS	LIN_OC_IS
r	r	r	r	r	r	r	r	r	r		r		r	r	r

Field	Bits	Type	Description
RES	31:30	r	**Reserved** Always read as 0
PHW_ZCHI_STS	29	r	**Phase W Zero Crossing Comparator High Status** 0_B **INACTIVE**, no status set 1_B **ACTIVE**, at least one status set
PHW_ZCLOW_STS	28	r	**Phase W Zero Crossing Comparator Low Status** 0_B **INACTIVE**, no status set 1_B **ACTIVE**, at least one status set
PHV_ZCHI_STS	27	r	**Phase V Zero Crossing Comparator High Status** 0_B **INACTIVE**, no status set 1_B **ACTIVE**, at least one status set
PHV_ZCLOW_STS	26	r	**Phase V Zero Crossing Comparator Low Status** 0_B **INACTIVE**, no status set 1_B **ACTIVE**, at least one status set
PHU_ZCHI_STS	25	r	**Phase U Zero Crossing Comparator High Status** 0_B **INACTIVE**, no status set 1_B **ACTIVE**, at least one status set
PHU_ZCLOW_STS	24	r	**Phase U Zero Crossing Comparator Low Status** 0_B **INACTIVE**, no status set 1_B **ACTIVE**, at least one status set
ADC4_EOC_IS	23	r	**ADC4 EOC Interrupt Status**[1)] 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
ADC3_EOC_IS	22	r	**ADC3 EOC Interrupt Status**[1)] 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set

Field	Bits	Type	Description
PHW_ZCHI_IS	21	r	**Phase W Zero Crossing Comparator High Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PHW_ZCLOW_IS	20	r	**Phase W Zero Crossing Comparator Low Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PHV_ZCHI_IS	19	r	**Phase V Zero Crossing Comparator High Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PHV_ZCLOW_IS	18	r	**Phase V Zero Crossing Comparator Low Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PHU_ZCHI_IS	17	r	**Phase U Zero Crossing Comparator High Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PHU_ZCLOW_IS	16	r	**Phase U Zero Crossing Comparator Low Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
ADC2_ESM_IS	15	r	**ADC2 Exceptional Sequence Measurement Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
VREF5V_OVL_IS	14	r	**VREF5V Overload Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
VREF5V_UPTH_IS	13	r	**VREF5V ADC1 Reference Overvoltage (ADC2, Channel 6) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
VREF5V_LOWTH_IS	12	r	**VREF5V ADC1 Reference Undervoltage (ADC2, Channel 6) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
REFBG_UPTHWARN_IS	11	r	**8-Bit ADC2 Reference Overvoltage (ADC2, Channel 7) interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
REFBG_LOTHWARN_IS	10	r	**8-Bit ADC2 Reference Undervoltage (ADC2, Channel 7) interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
SYS_OT_IS	9	r	**System Overtemperature Shutdown (ADC2, Channel 9) interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set

Field	Bits	Type	Description
SYS_OTWARN_IS	8	r	**System Overtemperature Prewarning (ADC2, Channel 9) interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PMU_OT_IS	7	r	**PMU Regulator Overtemperature Shutdown (ADC2, Channel 9) interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
PMU_OTWARN_IS	6	r	**PMU Regulator Overtemperature Prewarning (ADC2, Channel 9) interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set
RES	5:3	r	**Reserved** Always read as 0
LIN_TMOUT_IS	2	r	**LIN TXD timeout** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, LIN TXD timeout occurred
LIN_OT_IS	1	r	**LIN Overtemperature interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, LIN Overtemperature occurred
LIN_OC_IS	0	r	**LIN Overcurrent interrupt status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, at least one interrupt status set

1) Available only for products with SDADC. See the datasheet for details.

System Interrupt Status Clear

The register is reset by RESET_TYPE_3.

SYS_ISCLR
System Interrupt Status Clear **(14_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		PHW_ZCHI_SCLR	PHW_ZCLOW_SCLR	PHV_ZCHI_SCLR	PHV_ZCLOW_SCLR	PHU_ZCHI_SCLR	PHU_ZCLOW_SCLR	ADC4_EOC_ICLR	ADC3_EOC_ICLR	PHW_ZCHI_ICLR	PHW_ZCLOW_ICLR	PHV_ZCHI_ICLR	PHV_ZCLOW_ICLR	PHU_ZCHI_ICLR	PHU_ZCLOW_ICLR
r		w	w	w	w	w	w	w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2_ESM_ICLR	VREF5V_OVL_ICLR	VREF5V_UPTH_ICLR	VREF5V_LOWTH_ICLR	REFBG_UPTHWARN_IC	REFBG_LOTHWARN_ICL	SYS_OT_ICLR	SYS_OTWARN_ICLR	PMU_OT_ICLR	PMU_OTWARN_ICLR	RES			LIN_TMOUT_ICLR	LIN_OT_ICLR	LIN_OC_ICLR
w	w	w	w	w	w	w	w	w	w		r		w	w	w

Field	Bits	Type	Description
RES	31:30	r	**Reserved** Always read as 0
PHW_ZCHI_SCLR	29	w	**Phase W Zero Crossing Comparator High Status** 0_B no operation 1_B clear status flag
PHW_ZCLOW_SCLR	28	w	**Phase W Zero Crossing Comparator Low Status** 0_B no operation 1_B clear status flag
PHV_ZCHI_SCLR	27	w	**Phase V Zero Crossing Comparator High Status** 0_B no operation 1_B clear status flag
PHV_ZCLOW_SCLR	26	w	**Phase V Zero Crossing Comparator Low Status** 0_B no operation 1_B clear status flag
PHU_ZCHI_SCLR	25	w	**Phase U Zero Crossing Comparator High Status** 0_B no operation 1_B clear status flag
PHU_ZCLOW_SCLR	24	w	**Phase U Zero Crossing Comparator Low Status** 0_B no operation 1_B clear status flag
ADC4_EOC_ICLR	23	w	**ADC4 EOC Interrupt Status**[1)] 0_B no operation 1_B clear status flag
ADC3_EOC_ICLR	22	w	**ADC3 EOC Interrupt Status**[1)] 0_B no operation 1_B clear status flag

Field	Bits	Type	Description
PHW_ZCHI_ICLR	21	w	**Phase W Zero Crossing Comparator High Interrupt Status** 0_B no operation 1_B clear status flag
PHW_ZCLOW_ICLR	20	w	**Phase W Zero Crossing Comparator Low Interrupt Status** 0_B no operation 1_B clear status flag
PHV_ZCHI_ICLR	19	w	**Phase V Zero Crossing Comparator High Interrupt Status** 0_B no operation 1_B clear status flag
PHV_ZCLOW_ICLR	18	w	**Phase V Zero Crossing Comparator Low Interrupt Status** 0_B no operation 1_B clear status flag
PHU_ZCHI_ICLR	17	w	**Phase U Zero Crossing Comparator High Interrupt Status** 0_B no operation 1_B clear status flag
PHU_ZCLOW_ICLR	16	w	**Phase U Zero Crossing Comparator Low Interrupt Status** 0_B no operation 1_B clear status flag
ADC2_ESM_ICLR	15	w	**ADC2 Exceptional Sequence Measurement Interrupt Status** 0_B no operation 1_B clear status flag
VREF5V_OVL_ICLR	14	w	**VREF5V Overload Interrupt Status** 0_B no operation 1_B clear status flag
VREF5V_UPTH_ICLR	13	w	**VREF5V ADC1 Reference Overvoltage (ADC2, Channel 6) Interrupt Status** 0_B no operation 1_B clear status flag
VREF5V_LOWTH_ICLR	12	w	**VREF5V ADC1 Reference Undervoltage (ADC2, Channel 6) Interrupt Status** 0_B no operation 1_B clear status flag
REFBG_UPTHWARN_ICLR	11	w	**8-Bit ADC2 Reference Overvoltage (ADC2, Channel 7) interrupt status** 0_B no operation 1_B clear status flag
REFBG_LOTHWARN_ICLR	10	w	**8-Bit ADC2 Reference Undervoltage (ADC2, Channel 7) interrupt status** 0_B no operation 1_B clear status flag
SYS_OT_ICLR	9	w	**System Overtemperature Shutdown (ADC2, Channel 9) interrupt status** 0_B no operation 1_B clear status flag

Field	Bits	Type	Description
SYS_OTWARN_ICLR	8	w	**System Overtemperature Prewarning (ADC2, Channel 9) interrupt status** 0_B no operation 1_B clear status flag
PMU_OT_ICLR	7	w	**PMU Regulator Overtemperature Shutdown (ADC2, Channel 9) interrupt status** 0_B no operation 1_B clear status flag
PMU_OTWARN_ICLR	6	w	**PMU Regulator Overtemperature Prewarning (ADC2, Channel 9) interrupt status** 0_B no operation 1_B clear status flag
RES	5:3	r	**Reserved** Always read as 0
LIN_TMOUT_ICLR	2	w	**LIN TXD timeout** 0_B no operation 1_B clear status flag
LIN_OT_ICLR	1	w	**LIN Overtemperature interrupt status** 0_B no operation 1_B clear status flag
LIN_OC_ICLR	0	w	**LIN Overcurrent interrupt status** 0_B no operation 1_B clear status flag

1) Available only for products with SDADC. See the datasheet for details.

Bridge Driver Interrupt Status

The register is reset by RESET_TYPE_4.

BDRV_IS
Bridge Driver Interrupt Status **(58_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			VSD_UPTH_STS	VSD_LOWTH_STS	VCP_UPTH_STS	VCP_LOWTH1_STS	VCP_LOWTH2_STS	RES			VSD_UPTH_IS	VSD_LOWTH_IS	VCP_UPTH_IS	VCP_LOWTH1_IS	VCP_LOWTH2_IS
	r		r	r	r	r	r		r		r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_IS	LS3_OC_IS	HS2_OC_IS	HS1_OC_IS	LS2_OC_IS	LS1_OC_IS	RES				HS3_DS_IS	LS3_DS_IS	HS2_DS_IS	HS1_DS_IS	LS2_DS_IS	LS1_DS_IS
r	r	r	r	r	r		r			r	r	r	r	r	r

Field	Bits	Type	Description
RES	31:29	r	**Reserved** Always read as 0
VSD_UPTH_STS	28	r	**Warning for VSD Upper Threshold Measurement (ADC2 channel 2) Status** 0_B **INACTIVE**, no overvoltage status set 1_B **ACTIVE**, overvoltage status set
VSD_LOWTH_STS	27	r	**Warning for VSD Lower Threshold Measurement (ADC2 channel 2) Status** 0_B **INACTIVE**, no undervoltage status set 1_B **ACTIVE**, undervoltage status set
VCP_UPTH_STS	26	r	**Warning for VCP Upper Threshold Measurement (ADC2 channel 3) Status** 0_B **INACTIVE**, no overvoltage status set 1_B **ACTIVE**, overvoltage status set
VCP_LOWTH1_STS	25	r	**Warning for VCP Lower Threshold 1 Measurement (ADC2 channel 3) Status** 0_B **INACTIVE**, no undervoltage status set 1_B **ACTIVE**, undervoltage status set
VCP_LOWTH2_STS	24	r	**Warning for VCP Lower Threshold 2 Measurement (VCP_LOW Signal from CP) Status** 0_B **INACTIVE**, no undervoltage status set 1_B **ACTIVE**, undervoltage status set
RES	23:21	r	**Reserved** Always read as 0
VSD_UPTH_IS	20	r	**Warning for VSD Upper Threshold Measurement (ADC2 channel 2) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set

Field	Bits	Type	Description
VSD_LOWTH_IS	19	r	**Warning for VSD Lower Threshold Measurement (ADC2 channel 2) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
VCP_UPTH_IS	18	r	**Warning for VCP Upper Threshold Measurement (ADC2 channel 3) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
VCP_LOWTH1_IS	17	r	**Warning for VCP Lower Threshold 1 Measurement (ADC2 channel 3) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
VCP_LOWTH2_IS	16	r	**Warning for VCP Lower Threshold 2 Measurement (VCP_LOW Signal from CP) Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
HS3_OC_IS	15	r	**External High Side 3 FET Over-current Status** 0_B **INACTIVE**, no interrupt status set. 1_B **ACTIVE**, at least one interrupt status set.
LS3_OC_IS	14	r	**External Low Side 3 FET Over-current Status** 0_B **INACTIVE**, no interrupt status set. 1_B **ACTIVE**, at least one interrupt status set.
HS2_OC_IS	13	r	**External High Side 2 FET Over-current Status** 0_B **INACTIVE**, no interrupt status set. 1_B **ACTIVE**, at least one interrupt status set.
HS1_OC_IS	12	r	**External High 1 FET Over-current Status** 0_B **INACTIVE**, no interrupt status set. 1_B **ACTIVE**, at least one interrupt status set.
LS2_OC_IS	11	r	**External Low Side 2 FET Over-current Status** 0_B **INACTIVE**, no interrupt status set. 1_B **ACTIVE**, at least one interrupt status set.
LS1_OC_IS	10	r	**External Low Side 1 FET Over-current Status** 0_B **INACTIVE**, no interrupt status set. 1_B **ACTIVE**, at least one interrupt status set.
RES	9:6	r	**Reserved** Always read as 0
HS3_DS_IS	5	r	**Bridge Driver High Side 3 Pre-Driver short Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
LS3_DS_IS	4	r	**Bridge Driver Low Side 3 Pre-Driver short Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
HS2_DS_IS	3	r	**Bridge Driver High Side 2 Pre-Driver short Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set

Field	Bits	Type	Description
HS1_DS_IS	2	r	**Bridge Driver High Side 1 Pre-Driver short Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
LS2_DS_IS	1	r	**Bridge Driver Low Side 2 Pre-Driver short Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set
LS1_DS_IS	0	r	**Bridge Driver Low Side 1 Pre-Driver short Interrupt Status** 0_B **INACTIVE**, no interrupt status set 1_B **ACTIVE**, interrupt status set

Bridge Driver Interrupt Status Clear

The register is reset by RESET_TYPE_3.

BDRV_ISCLR
Bridge Driver Interrupt Status Clear **(54_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			VSD_UPTH_SCLR	VSD_LOWTH_SCLR	VCP_UPTH_SCLR	VCP_LOWTH1_SCLR	VCP_LOWTH2_SCLR	RES			VSD_UPTH_ICLR	VSD_LOWTH_ICLR	VCP_UPTH_ICLR	VCP_LOWTH1_ICLR	VCP_LOWTH2_ICLR
	r		w	w	w	w	w		r		w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_ICLR	LS3_OC_ICLR	HS2_OC_ICLR	HS1_OC_ICLR	LS2_OC_ICLR	LS1_OC_ICLR	RES				HS3_DS_ICLR	LS3_DS_ICLR	HS2_DS_ICLR	HS1_DS_ICLR	LS2_DS_ICLR	LS1_DS_ICLR
w	w	w	w	w	w			r		w	w	w	w	w	w

Field	Bits	Type	Description
RES	31:29	r	**Reserved** Always read as 0
VSD_UPTH_SCLR	28	w	**Warning for VSD Upper Threshold Measurement (ADC2 channel 2) Status** 0_B no operation 1_B clear status flag
VSD_LOWTH_SCLR	27	w	**Warning for VSD Lower Threshold Measurement (ADC2 channel 2) Status** 0_B no operation 1_B clear status flag
VCP_UPTH_SCLR	26	w	**Warning for VCP Upper Threshold Measurement (ADC2 channel 3) Status** 0_B no operation 1_B clear status flag
VCP_LOWTH1_SCLR	25	w	**Warning for VCP Lower Threshold 1 Measurement (ADC2 channel 3) Status** 0_B no operation 1_B clear status flag
VCP_LOWTH2_SCLR	24	w	**Warning for VCP Lower Threshold 2 Measurement (VCP_LOW Signal from CP) Status** 0_B no operation 1_B clear status flag
RES	23:21	r	**Reserved** Always read as 0

Field	Bits	Type	Description
VSD_UPTH_ICLR	20	w	**Warning for VSD Upper Threshold Measurement (ADC2 channel 2) Interrupt Status** 0_B no operation 1_B clear status flag
VSD_LOWTH_ICLR	19	w	**Warning for VSD Lower Threshold Measurement (ADC2 channel 2) Interrupt Status** 0_B no operation 1_B clear status flag
VCP_UPTH_ICLR	18	w	**Warning for VCP Upper Threshold Measurement (ADC2 channel 3) Interrupt Status** 0_B no operation 1_B clear status flag
VCP_LOWTH1_ICLR	17	w	**Warning for VCP Lower Threshold 1 Measurement (ADC2 channel 3) Interrupt Status** 0_B no operation 1_B clear status flag
VCP_LOWTH2_ICLR	16	w	**Warning for VCP Lower Threshold 2 Measurement (VCP_LOW Signal from CP) Interrupt Status** 0_B no operation 1_B clear status flag
HS3_OC_ICLR	15	w	**External High Side 3 FET Over-current Status** 0_B no operation 1_B clear status flag
LS3_OC_ICLR	14	w	**External Low Side 3 FET Over-current Status** 0_B no operation 1_B clear status flag
HS2_OC_ICLR	13	w	**External High Side 2 FET Over-current Status** 0_B no operation 1_B clear status flag
HS1_OC_ICLR	12	w	**External High 1 FET Over-current Status** 0_B no operation 1_B clear status flag
LS2_OC_ICLR	11	w	**External Low Side 2 FET Over-current Status** 0_B no operation 1_B clear status flag
LS1_OC_ICLR	10	w	**External Low Side 1 FET Over-current Status** 0_B no operation 1_B clear status flag
RES	9:6	r	**Reserved** Always read as 0
HS3_DS_ICLR	5	w	**Bridge Driver High Side 3 Pre-Driver short Interrupt Status** 0_B no operation 1_B clear status flag

Field	Bits	Type	Description
LS3_DS_ICLR	4	w	**Bridge Driver Low Side 3 Pre-Driver short Interrupt Status** 0_B no operation 1_B clear status flag
HS2_DS_ICLR	3	w	**Bridge Driver High Side 2 Pre-Driver short Interrupt Status** 0_B no operation 1_B clear status flag
HS1_DS_ICLR	2	w	**Bridge Driver High Side 1 Pre-Driver short Interrupt Status** 0_B no operation 1_B clear status flag
LS2_DS_ICLR	1	w	**Bridge Driver Low Side 2 Pre-Driver short Interrupt Status** 0_B no operation 1_B clear status flag
LS1_DS_ICLR	0	w	**Bridge Driver Low Side 1 Pre-Driver short Interrupt Status** 0_B no operation 1_B clear status flag

System Supply Interrupt Status

The register is reset by RESET_TYPE_4.

SYS_SUPPLY_IRQ_STS
System Supply Interrupt Status **($1C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								VDD1V5_OV_STS	VDD5V_OV_STS	VS_OV_STS	MON_OV_STS	VDD1V5_UV_STS	VDD5V_UV_STS	VS_UV_STS	MON_UV_STS
r								r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								VDD1V5_OV_IS	VDD5V_OV_IS	VS_OV_IS	MON_OV_IS	VDD1V5_UV_IS	VDD5V_UV_IS	VS_UV_IS	MON_UV_IS
r								r	r	r	r	r	r	r	r

Field	Bits	Type	Description
RES	31:24	r	**Reserved** Always read as 0
VDD1V5_OV_STS	23	r	**VDDC Overvoltage (ADC2 channel 8) Status** 0_B **No Overvoltage**, occurred 1_B **Overvoltage**, occurred
VDD5V_OV_STS	22	r	**VDDP Overvoltage (ADC2 channel 5) Status** 0_B **No Overvoltage**, occurred 1_B **Overvoltage** , occurred
VS_OV_STS	21	r	**VS Overvoltage (ADC2 channel 1) Status** 0_B **No Overvoltage** , occurred 1_B **Overvoltage**, occurred
MON_OV_STS	20	r	**MON Overvoltage (ADC2 channel 4) Status** 0_B **No Overvoltage**, occurred 1_B **Overvoltage**, occurred
VDD1V5_UV_STS	19	r	**VDDC Undervoltage (ADC2 channel 8) Status** 0_B **No Undervoltage**, occurred 1_B **Undervoltage**, occurred
VDD5V_UV_STS	18	r	**VDDP Undervoltage (ADC2 channel 5) Status** 0_B **No Undervoltage**, occurred 1_B **Undervoltage**, occurred
VS_UV_STS	17	r	**VS Undervoltage (ADC2 channel 1) Status** 0_B **No Undervoltage**, occurred 1_B **Undervoltage**, occurred
MON_UV_STS	16	r	**MON Undervoltage (ADC2 channel 4) Status** 0_B **No Undervoltage**, occurred 1_B **Undervoltage**, occurred

Field	Bits	Type	Description
RES	15:8	r	**Reserved** Always read as 0
VDD1V5_OV_IS	7	r	**VDDC Overvoltage (ADC2 channel 8) Interrupt Status** 0_B **No Overvoltage Interrupt**, occurred 1_B **Overvoltage Interrupt**, occurred
VDD5V_OV_IS	6	r	**VDDP Overvoltage (ADC2 channel 5) Interrupt Status** 0_B **No Overvoltage Interrupt**, occurred 1_B **Overvoltage Interrupt**, occurred
VS_OV_IS	5	r	**VS Overvoltage (ADC2 channel 1) Interrupt Status** 0_B **No Overvoltage Interrupt**, occurred 1_B **Overvoltage Interrupt**, occurred
MON_OV_IS	4	r	**MON Overvoltage (ADC2 channel 4) Interrupt Status** 0_B **No Overvoltage Interrupt**, occurred 1_B **Overvoltage Interrupt**, occurred
VDD1V5_UV_IS	3	r	**VDDC Undervoltage (ADC2 channel 8) Interrupt Status** 0_B **No Undervoltage Interrupt**, occurred 1_B **Undervoltage Interrupt**, occurred
VDD5V_UV_IS	2	r	**VDDP Undervoltage (ADC2 channel 5) Interrupt Status** 0_B **No Undervoltage Interrupt**, occurred 1_B **Undervoltage Interrupt**, occurred
VS_UV_IS	1	r	**VS Undervoltage (ADC2 channel 1) Interrupt Status** 0_B **No Undervoltage Interrupt**, occurred 1_B **Undervoltage Interrupt**, occurred
MON_UV_IS	0	r	**MON Undervoltage (ADC2 channel 4) Interrupt Status** 0_B **No Undervoltage Interrupt**, occurred 1_B **Undervoltage Interrupt**, occurred

System Supply Interrupt Status Clear

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of 0101 $xxxx_H$

SYS_SUPPLY_IRQ_CLR

System Supply Interrupt Status Clear **(24_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								VDD1V5_OV_SCLR	VDD5V_OV_SCLR	VS_OV_SCLR	MON_OV_SCLR	VDD1V5_UV_SCLR	VDD5V_UV_SCLR	VS_UV_SCLR	MON_UV_SCLR
r								w	w	w	w	w	w	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								VDD1V5_OV_ICLR	VDD5V_OV_ICLR	VS_OV_ICLR	MON_OV_ICLR	VDD1V5_UV_ICLR	VDD5V_UV_ICLR	VS_UV_ICLR	MON_UV_ICLR
r								w	w	w	w	w	w	w	w

Field	Bits	Type	Description
RES	31:24	r	**Reserved** Always read as 0
VDD1V5_OV_SCLR	23	w	**VDDC Overvoltage (ADC2 channel 8) Status Clear** 0_B no operation 1_B clear status flag
VDD5V_OV_SCLR	22	w	**VDDP Overvoltage (ADC2 channel 5) Status Clear** 0_B no operation 1_B clear status flag
VS_OV_SCLR	21	w	**VS Overvoltage (ADC2 channel 1) Status Clear** 0_B no operation 1_B clear status flag
MON_OV_SCLR	20	w	**MON Overvoltage (ADC2 channel 4) Status Clear** 0_B no operation 1_B clear status flag
VDD1V5_UV_SCLR	19	w	**VDDC Undervoltage (ADC2 channel 8) Status Clear** 0_B no operation 1_B clear status flag
VDD5V_UV_SCLR	18	w	**VDDP Undervoltage (ADC2 channel 5) Status Clear** 0_B no operation 1_B clear status flag
VS_UV_SCLR	17	w	**VS Undervoltage (ADC2 channel 1) Status Clear** 0_B no operation 1_B clear status flag
MON_UV_SCLR	16	w	**MON Undervoltage (ADC2 channel 4) Status Clear** 0_B no operation 1_B clear status flag

Field	Bits	Type	Description
RES	15:8	r	**Reserved** Always read as 0
VDD1V5_OV_ICLR	7	w	**VDDC Overvoltage (ADC2 channel 8) Interrupt Status Clear** 0_B no operation 1_B clear status flag
VDD5V_OV_ICLR	6	w	**VDDP Overvoltage (ADC2 channel 5) Interrupt Status Clear** 0_B no operation 1_B clear status flag
VS_OV_ICLR	5	w	**VS Overvoltage (ADC2 channel 1) Interrupt Status Clear** 0_B no operation 1_B clear status flag
MON_OV_ICLR	4	w	**MON Overvoltage (ADC2 channel 4) Interrupt Status Clear** 0_B no operation 1_B clear status flag
VDD1V5_UV_ICLR	3	w	**VDDC Undervoltage (ADC2 channel 8) Interrupt Status Clear** 0_B no operation 1_B clear status flag
VDD5V_UV_ICLR	2	w	**VDDP Undervoltage (ADC2 channel 5) Interrupt Status Clear** 0_B no operation 1_B clear status flag
VS_UV_ICLR	1	w	**VS Undervoltage (ADC2 channel 1) Interrupt Status Clear** 0_B no operation 1_B clear status flag
MON_UV_ICLR	0	w	**MON Undervoltage (ADC2 channel 4) Interrupt Status Clear** 0_B no operation 1_B clear status flag

7.4.2.2 Interrupt Control Unit - Interrupt Enable Register

The Analog Module Interrupts can be enabled and disabled by the corresponding enable bits which are located in Registers:

- **SYS_SUPPLY_IRQ_CTRL**: Enable of Interrupts for Under- and Overvoltage detection for all system relevant supplies. These interrupts are edge triggered interrupts to reduce interrupt load of the µC.
- **SYS_IRQ_CTRL**: Enable of interrupts for Analog Modules.

System Interrupt Control

The register is reset by RESET_TYPE_3.

SYS_IRQ_CTRL

System Interrupt Control **(28_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								ADC4_EOC_IE	ADC3_EOC_IE	PHW_ZCHI_IE	PHW_ZCLOW_IE	PHV_ZCHI_IE	PHV_ZCLOW_IE	PHU_ZCHI_IE	PHU_ZCLOW_IE
r								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC2_ESM_IE	VREF5V_OVL_IE	VREF5V_UPTH_IE	VREF5V_LOWTH_IE	REFBG_UPTHWARN_IE	REFBG_LOTHWARN_IE	SYS_OT_IE	SYS_OTWARN_IE	PMU_OT_IE	PMU_OTWARN_IE	RES			LIN_TMOUT_IE	LIN_OT_IE	LIN_OC_IE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r			rw	rw	rw

Field	Bits	Type	Description
RES	31:24	r	**Reserved** Always read as 0
ADC4_EOC_IE	23	rw	**ADC4 EOC Interrupt Enable**[1)] 0_B Interrupt is disabled 1_B Interrupt is enabled
ADC3_EOC_IE	22	rw	**ADC3 EOC Interrupt Enable**[1)] 0_B Interrupt is disabled 1_B Interrupt is enabled
PHW_ZCHI_IE	21	rw	**Phase W Zero Crossing Comparator High Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
PHW_ZCLOW_IE	20	rw	**Phase W Zero Crossing Comparator Low Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
PHV_ZCHI_IE	19	rw	**Phase V Zero Crossing Comparator High Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
PHV_ZCLOW_IE	18	rw	**Phase V Zero Crossing Comparator Low Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

Field	Bits	Type	Description
PHU_ZCHI_IE	17	rw	**Phase U Zero Crossing Comparator High Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
PHU_ZCLOW_IE	16	rw	**Phase U Zero Crossing Comparator Low Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
ADC2_ESM_IE	15	rw	**ADC2 Exceptional Sequence Measurement Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VREF5V_OVL_IE	14	rw	**VREF5V Overload Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VREF5V_UPTH_IE	13	rw	**VREF5V ADC1 Reference Overvoltage (ADC2, Channel 6) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VREF5V_LOWTH_IE	12	rw	**VREF5V ADC1 Reference Undervoltage (ADC2, Channel 6) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
REFBG_UPTHWARN_IE	11	rw	**Reference Voltage Overvoltage (ADC2 channel 7) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
REFBG_LOTHWARN_IE	10	rw	**Reference Voltage Undervoltage (ADC2 channel 7) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
SYS_OT_IE	9	rw	**System Overtemperature Shutdown (ADC2 channel 9) Interrupt Enable (leads to shutdown of System)** 0_B Interrupt is disabled 1_B Interrupt is enabled
SYS_OTWARN_IE	8	rw	**System Overtemperature Warning (ADC2 channel 9) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
PMU_OT_IE	7	rw	**PMU Regulator Overtemperature Shutdown (ADC2 channel 9) Interrupt Enable (leads to shutdown of System)** 0_B Interrupt is disabled 1_B Interrupt is enabled
PMU_OTWARN_IE	6	rw	**PMU Regulator Overtemperature Warning (ADC2 channel 9) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

Field	Bits	Type	Description
RES	5:3	r	**Reserved** Always read as 0
LIN_TMOUT_IE	2	rw	**LIN TXD timeout Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LIN_OT_IE	1	rw	**LIN Overtemperature Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LIN_OC_IE	0	rw	**LIN Overcurrent Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

1) Available only for products with SDADC. See the datasheet for details.

Bridge Driver Interrupt Control

The register is reset by RESET_TYPE_3.

BDRV_IRQ_CTRL
Bridge Driver Interrupt Control **($5C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											VSD_UPTH_IE	VSD_LOWTH_IE	VCP_UPTH_IE	VCP_LOWTH1_IE	VCP_LOWTH2_IE
r											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_IE	LS3_OC_IE	HS2_OC_IE	HS1_OC_IE	LS2_OC_IE	LS1_OC_IE	RES				HS3_DS_IE	LS3_DS_IE	HS2_DS_IE	HS1_DS_IE	LS2_DS_IE	LS1_DS_IE
rw	rw	rw	rw	rw	rw	r				rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:21	r	**Reserved** Always read as 0
VSD_UPTH_IE	20	rw	**VSD Measurement Upper Threshold (ADC2 channel 2) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VSD_LOWTH_IE	19	rw	**VSD Measurement Lower Threshold (ADC2 channel 2) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VCP_UPTH_IE	18	rw	**VCP Measurement Upper Threshold (ADC2 channel 2) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VCP_LOWTH1_IE	17	rw	**VCP Measurement Lower Threshold 1 (ADC2 channel 3) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VCP_LOWTH2_IE	16	rw	**VCP Measurement Lower Threshold 2 Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
HS3_OC_IE	15	rw	**External High Side 3 FET Over-current Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LS3_OC_IE	14	rw	**External Low Side 3 FET Over-current Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

Field	Bits	Type	Description
HS2_OC_IE	13	rw	**External High Side 2 FET Over-current Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
HS1_OC_IE	12	rw	**External High Side 1 FET Over-current Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LS2_OC_IE	11	rw	**External Low Side 2 FET Over-current Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LS1_OC_IE	10	rw	**External Low Side 1 FET Over-current Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
RES	9:6	r	**Reserved**
HS3_DS_IE	5	rw	**Bridge Driver High Side 3 Pre-Driver Short Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LS3_DS_IE	4	rw	**Bridge Driver Low Side 3 Pre-Driver Short Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
HS2_DS_IE	3	rw	**Bridge Driver High Side 2 Pre-Driver Short Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
HS1_DS_IE	2	rw	**Bridge Driver High Side 1 Pre-Driver Short Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LS2_DS_IE	1	rw	**Bridge Driver Low Side 2 Pre-Driver Short Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
LS1_DS_IE	0	rw	**Bridge Driver Low Side 1 Pre-Driver Short Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

System Supply Interrupt Control

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of $0000\ 0000_H$

SYS_SUPPLY_IRQ_CTRL

System Supply Interrupt Control **(20_H)** **Reset Value: $0000\ 00FF_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								VDD1V5_OV_IE	VDD5V_OV_IE	VS_OV_IE	MON_OV_IE	VDD1V5_UV_IE	VDD5V_UV_IE	VS_UV_IE	MON_UV_IE
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
VDD1V5_OV_IE	7	rw	**VDD1V5 Overvoltage (ADC2 channel 8) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VDD5V_OV_IE	6	rw	**VDD5V Overvoltage (ADC2 channel 5) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VS_OV_IE	5	rw	**VS Overvoltage (ADC2 channel 1) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
MON_OV_IE	4	rw	**MON Overvoltage (ADC2 channel 4) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VDD1V5_UV_IE	3	rw	**VDD1V5 Undervoltage (ADC2 channel 8) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VDD5V_UV_IE	2	rw	**VDD5V Undervoltage (ADC2 channel 5) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled
VS_UV_IE	1	rw	**VS Undervoltage (ADC2 channel 1) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

Field	Bits	Type	Description
MON_UV_IE	0	rw	**MON Undervoltage (ADC2 channel 4) Interrupt Enable** 0_B Interrupt is disabled 1_B Interrupt is enabled

7.5 Power Control Unit for Power Modules (PCU_PM)

The chapter describes the implementation of the power modules state machine. This state machine is responsible for powering up and powering down the on-board power modules. It takes care about the interaction between the Measurement Unit and the modules which are evaluated by the Unit. The following modules are controlled by this state machine:

Analog Modules controlled by Power Control Unit:

- Central Reference Voltage Generation
- Central Bias Current Generation
- 8-Bit ADC Core
- Supply Voltage Attenuators
- Monitoring Inputs Voltage Attenuators
- LIN Transceiver
- MOSFET Drivers

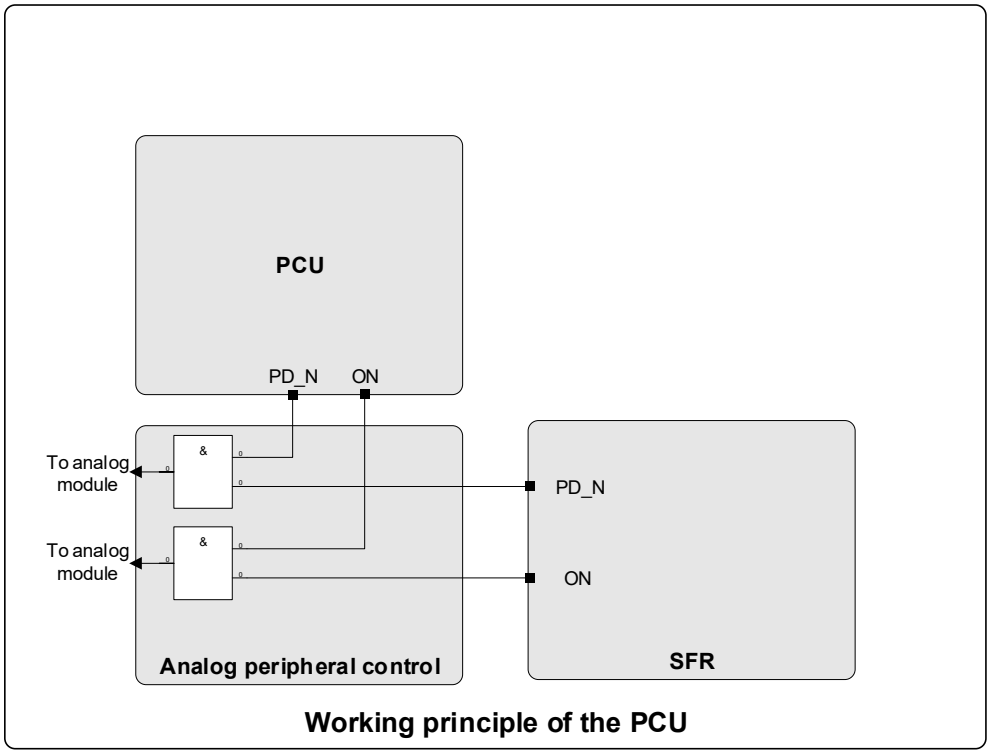


Figure 37 Function of AP_SUB_CTRL

If the device will power up the analog modules state machine will startup all analog modules. First of all, the reference voltage will be enabled. After that the biasing module will be enabled. If this step is completed the analog modules will be enabled step by step. After this is done the measurement interface will start-up.

When leaving Stop Mode, this sequence restores the SFR register contents with the values written before entering Stop Mode.

The Sleep and Stop Mode entry is as well controlled by this state machine. This ensures a smooth shutdown of the modules avoiding disturbances (like load jumps) on the supplies.

The power control unit also handles system failures indicated by the analog measurement interface. They are:

System failures handled by SCU_PM:

- Automatic shutdown of power modules in case of system overtemperature
- Automatic shutdown of power modules in case of loss of clock
- Automatic shutdown of system in case of system overtemperature
- Automatic shutdown of system in case of internal supply fail
- Automatic shutdown of LIN module in case of VS undervoltage

How to configure this actions on the above described system failures will be described in the following chapters.

7.5.1 VSD-Overvoltage System Shutdown

The PCU provides the possibility of an system shutdown in case of VSD Overvoltage. The system shutdown means the device enters Sleep Mode. This feature can be used to reduce power dissipation in case of an increased supply voltage VSD. This feature can be enabled by bit **SYS_VSD_OV_SLM_DIS**. **This bit is low active!**. When there is an overvoltage, the system will be set in system shutdown and all power modules as:

- Bridge driver
- Charge pump
- Supply voltage attenuators

are switched off automatically. The power modules will be switched on when the VSD-Overvoltage condition is left again.

7.5.2 Overtemperature System Shutdown

In case of overtemperature ($T_j > T_{j,max}$) the system will be set to Sleep Mode. This functionality is intended to protect the system from thermal overstress. One possibility to avoid this thermal shutdown is to actively reduce the power dissipation of the system, by clocking down the microcontroller subsystem, or reducing the PWM frequency of a motor control, which helps to reduce the power dissipation in the system. This procedure has to be implemented in user software, and should be triggered by the overtemperature prewarning (ADC2.Ch9 lower-threshold).

System Overtemperature Power Stage Disable

Another possibility is to use the implemented hardware shutdown procedure. This procedure can be activated by the flag **PCU_CTRL_STS**.SYS_OT_PS_DIS. **This flag is low active!** When this flag is set enabled (SYS_OT_PS_DIS=0) the following modules will be automatically shutdown (based on ADC2.Ch9 upper-threshold):

- Charge pump
- Bridge driver

After the ADC2.Ch9 overtemperture gets triggered (upper threshold exceeded), the modules mentioned above are shutdown after 64µs (based on TFILT_CLK). If the overtemperature condition remains for 1ms (based on TFILT_CLK), the device enters Sleep Mode.

Fail Power Stage Disable

If **PCU_CTRL_STS**.FAIL_PS_DIS = 0 (enabled) then the modules:

- Charge pump
- Bridge driver

are shutdown immediately after system overtemperature is detected (ADC2.Ch9 upper-threshold exceeded).

7.5.3 Power Control Unit Register

The PCU is fully controllable by the below listed SFR Registers.

Table 42 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Power Control Unit Register			
PCU_CTRL_STS	Power Control Unit Control Status Register	30_H	$82E3_7EF3_H$

The registers are addressed wordwise.

Power Control Unit Register

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of $82E3\ 7EF3_H$

PCU_CTRL_STS

Power Control Unit Control Status Register (30_H) Reset Value: $82E3\ 7EF3_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**RES**				**CLKWDT_RES_SD_DIS**	**CLKLOSS_SD_DIS**	**SYS_OT_PS_DIS**	**RES**							
r	r				rw	rw	rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**SYS_VSD_OV_SLM_DIS**	**RES**					**LIN_VS_UV_SD_DIS**	**FAIL_PS_DIS**	**RES**					**CLKWDT_SD_DIS**	**RES**
r	rw	r					rw	rw	r					rw	r

Field	Bits	Type	Description
RES	31	r	**Reserved** Always read as 1
RES	30:27	r	**Reserved** Always read as 0
CLKWDT_RES_SD_DIS	26	rw	**Clock Watchdog Reset Disable** 0_B **Enable**, Clock Watchdog Reset Enable 1_B **Disable**, Clock Watchdog Reset Disable
CLKLOSS_SD_DIS	25	rw	**Power Switches Loss of Clock Shutdown Disable (APCLK3)** 0_B **Enable**, Automatic Shutdown Signal for Power Switches in case of loss of clock 1_B **Disable**, Automatic Shutdown Signal for Power Switches in case of loss of clock

Field	Bits	Type	Description
SYS_OT_PS_DIS	24	rw	**System Overtemperature Power Switches Shutdown Disable (ADC2.Ch9 upper-threshold)** 0_B **Enable**, Automatic Shutdown Signal for Power Switches in case of system overtemperature enabled 1_B **Disable**, Automatic Shutdown Signal for Power Switches in case of system overtemperature disabled
RES	23:15	r	**Reserved** Always read as 0b111000110
SYS_VSD_OV_SLM_DIS	14	rw	**VSD Overvoltage Shutdown for Peripherals Disable** 0_B **Enable**, Automatic Shutdown for Power modules in case of VSD Overvoltage enabled 1_B **Disable**, Automatic Shutdown for Power modules in case of VSD Overvoltage disabled
RES	13:9	r	**Reserved** Always read as 0 0b11111
LIN_VS_UV_SD_DIS	8	rw	**LIN Module VS Undervoltage Transmitter Shutdown** 0_B **Enable**, Automatic Shutdown for Power modules in case of VS Undervoltage enabled 1_B **Disable**, Automatic Shutdown for Power modules in case of VS Undervoltage disabled
FAIL_PS_DIS	7	rw	**Disable BDRV and CP because of Overtemperature (ADC2.Ch9 upper-threshold)** 0_B **Switch off Enabled**, BDRV and CP will be turned off when Overtemperature occurs 1_B **Switch off Disabled**, BDRV and CP will not be turned off when Overtemperature occurs
RES	6:2	r	**Reserved** Always read as 0b11100
CLKWDT_SD_DIS	1	rw	**Power Modules Clock Watchdog Shutdown Disable** 0_B **Shutdown Enable**, Power Devices will be switched off when Clock Watchdog occurs 1_B **Shutdown Disable**, Power Devices will not be shutdown when Clock Watchdog occurs
RES	0	r	**Reserved** Always read as 1

7.6 MISC Control

7.6.1 Bus Faults

Due to the pipeline structure of the Arm® Cortex® M3 core, it is difficult to determine the fault address of a nonexistent or protected memory region. For this reason, the following register allows the user to determine the address of the memory access that caused the fault. The registers model the processor's instruction pipeline.

7.6.1.1 Bus Fault Control Register

For the determining the address of a bus fault exception the following registers should be used.

Table 43 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Bus Fault Control Register			
BFSTS	Bus Fault Status Register	80_H	$0000\ 0000_H$
DBFA	Data Bus Fault Address Register	84_H	$xxxx\ xxxx_H$
SBFA	System Bus Fault Address Register	88_H	$xxxx\ xxxx_H$
BFSTS_CLR	Bus Fault Status Clear Register	$8C_H$	$0000\ 0000_H$

The registers are addressed wordwise.

BUS_FAULT_STATUS

The register is reset by RESET_TYPE_3.

BFSTS

Bus Fault Status Register **(80_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES														SBFSTS	DBFSTS
r														r	r

Field	Bits	Type	Description
RES	31:2	r	**Reserved** Always read as 0
SBFSTS	1	r	**System Bus Fault Status Valid Flag** Address of System Bus Fault is valid and can be red out of SBFA Register 1_B **Valid**, Address is valid 0_B **not valid**, Address is not valid
DBFSTS	0	r	**Data Bus Fault Status Valid Flag** Address of Data Bus Fault is valid and can be red out of DBFA Register 1_B **Valid**, Address is valid 0_B **not valid**, Address is not valid

BUS_FAULT_STATUS_CLEAR

The register is reset by RESET_TYPE_3.

BFSTS_CLR
Bus Fault Status Clear Register **($8C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES														SBFSTSCLR	DBFSTSCLR
r														w	w

Field	Bits	Type	Description
RES	31:2	r	**Reserved** Always read as 0
SBFSTSCLR	1	w	**System Bus Fault Status Clear Flag** System Bus Fault valid flag is cleared. 1_B **Cleared**, Valid Address is cleared 0_B **not cleared**, Valid Address is not cleared
DBFSTSCLR	0	w	**Data Bus Fault Status Clear Flag** Data Bus Fault valid flag is cleared. 1_B **Cleared**, Address is cleared 0_B **not cleared**, Address is not cleared

DATA_BUS_FAULT_ADDRESS

The register is reset by RESET_TYPE_3.

DBFA
Data Bus Fault Address Register **(84_H)** **Reset Value: xxxx $xxxx_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DBFA															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBFA															
r															

Field	Bits	Type	Description
DBFA	31:0	r	**Reserved** Always read as 0

SYSTEM_BUS_FAULT_ADDRESS

The register is reset by RESET_TYPE_3.

SBFA
System Bus Fault Address Register **(88_H)** **Reset Value: xxxx $xxxx_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SBFA															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SBFA															
r															

Field	Bits	Type	Description
SBFA	31:0	r	**Reserved** Always read as 0

7.7 Register Access Hints

Registers in the SCUPM module are clocked with the MI_CLK. Write operations will be delayed by one MI_CLK cycle.

For handling the clearing if interrupt status flags special care might be necessary. Under certain coding constellations an extra write access to the interrupt clear flag will be necessary to add the extra MI_CLK cycle.

In case of interrupt status clear, two use cases in the software can be considered:

1. the writing to the interrupt clear flag and the reading of the corresponding interrupt status flag are having some other code in between
2. the writing to the interrupt clear flag and the reading of the corresponding interrupt status flag are placed in code right behind each other, with no code in between

Use Case 1 - Example, no extra write attempt to the interrupt clear register required

```
/* clearing of VAREF overvoltage interrupt status flag */
SCUPM->SYS_ISCLR.bit.VREF5V_UPTH_ICLR = 1;
/* some other code is placed here */
...
/* reading of VAREF overvoltage interrupt status flag */
if ( SCUPM->SYS_IS.bit.VREF5V_UPTH_IS == 1 )
{
 /* some user code here */
}
```

In this example, between the interrupt clear register write and the corresponding status register read, sufficient time is ensured, as some C code (at least one C instruction) is placed in between.

Use Case 2- Example, one extra write attempt to the interrupt clear register required

```
/* clearing of VAREF overvoltage interrupt status flag */
SCUPM->SYS_ISCLR.bit.VREF5V_UPTH_ICLR = 1;
/* place another interrupt status flag clear attempt to add an extra MI_CLK cycle*/
SCUPM->SYS_ISCLR.bit.VREF5V_UPTH_ICLR = 0;
/* reading of VAREF overvoltage interrupt status flag */
if ( SCUPM->SYS_IS.bit.VREF5V_UPTH_IS == 1 )
{
 /* some user code here */
}
```

In this example, between the interrupt clear register write and the corresponding status register read, another interrupt clear register write is placed. In order to add another MI_CLK cycle and ensure that the first interrupt status flag write attempt has reach the interrupt status register before it will be read again in the if clause right after it. Without this extra clear attempt the following interrupt status flag read might still reflect the interrupt status set.

An extra instruction gives the first clear attempt at least one MI_CLK cycle of time. Any other C instruction might be sufficient as well. Using the same interrupt clear write again is proposed. This access cannot be eliminated by the compiler by optimization.

Executing the second interrupt clear write with a ‘0’ is recommended, as so, no functional effect to the register takes place.

8 Arm® Cortex®-M3 core

8.1 Features

The key features of the Arm® Cortex®-M3 implemented are listed below.

Processor core: a low-gate-count core, with low-latency interrupt processing

- A subset of the Thumb®-2 instruction set
- Banked stack pointer (SP) only
- 32-bit hardware divide instructions, SDIV and UDIV (Thumb-2 instructions)
- Handler and thread modes
- Thumb and debug states
- Interruptible-continued instructions LDM/STM, push/pop for low interrupt latency
- Automatic processor state saving and restoration for low-latency interrupt service routine (ISR) entry and exit
- Arm® architecture v7-M Style BE8/LE support
- Arm®v6 unaligned accesses

Nested vectored interrupt controller (NVIC) closely integrated with the processor core to achieve low latency interrupt processing

- Interrupts, configurable from 1 to 16
- Bits of priority (4)
- Dynamic reprioritization of interrupts
- Priority grouping. This enables selection of preemptive interrupt levels and non-preemptive interrupt levels.
- Support for tail-chaining and late arrival of interrupts. This enables back-to-back interrupt processing without the overhead of state-saving and restoration between interrupts.
- Processor state automatically saved on interrupt entry, and restored on interrupt exit, with no instruction overhead.

Bus interfaces

- Advanced High-performance Bus-Lite (AHB-Lite) interfaces: ICode, DCode, and system bus interface
- Memory access alignment
- Write buffer for buffering of write data

8.2 Introduction

The Arm® Cortex®-M3 processor is a leading 32-bit processor and provides a high-performance and cost-optimized platform for a broad range of applications including microcontrollers, automotive body systems and industrial control systems. Like the other Arm® Cortex® family processors, the Arm® Cortex®-M3 processor implements the Thumb®-2 instruction set architecture. With the optimized feature set the Arm® Cortex®-M3 delivers 32-bit performance in an application space that is usually associated with 8- and 16-bit microcontrollers.

8.2.1 Block diagram

Figure 38 shows the functional blocks of the Arm® Cortex®-M3.

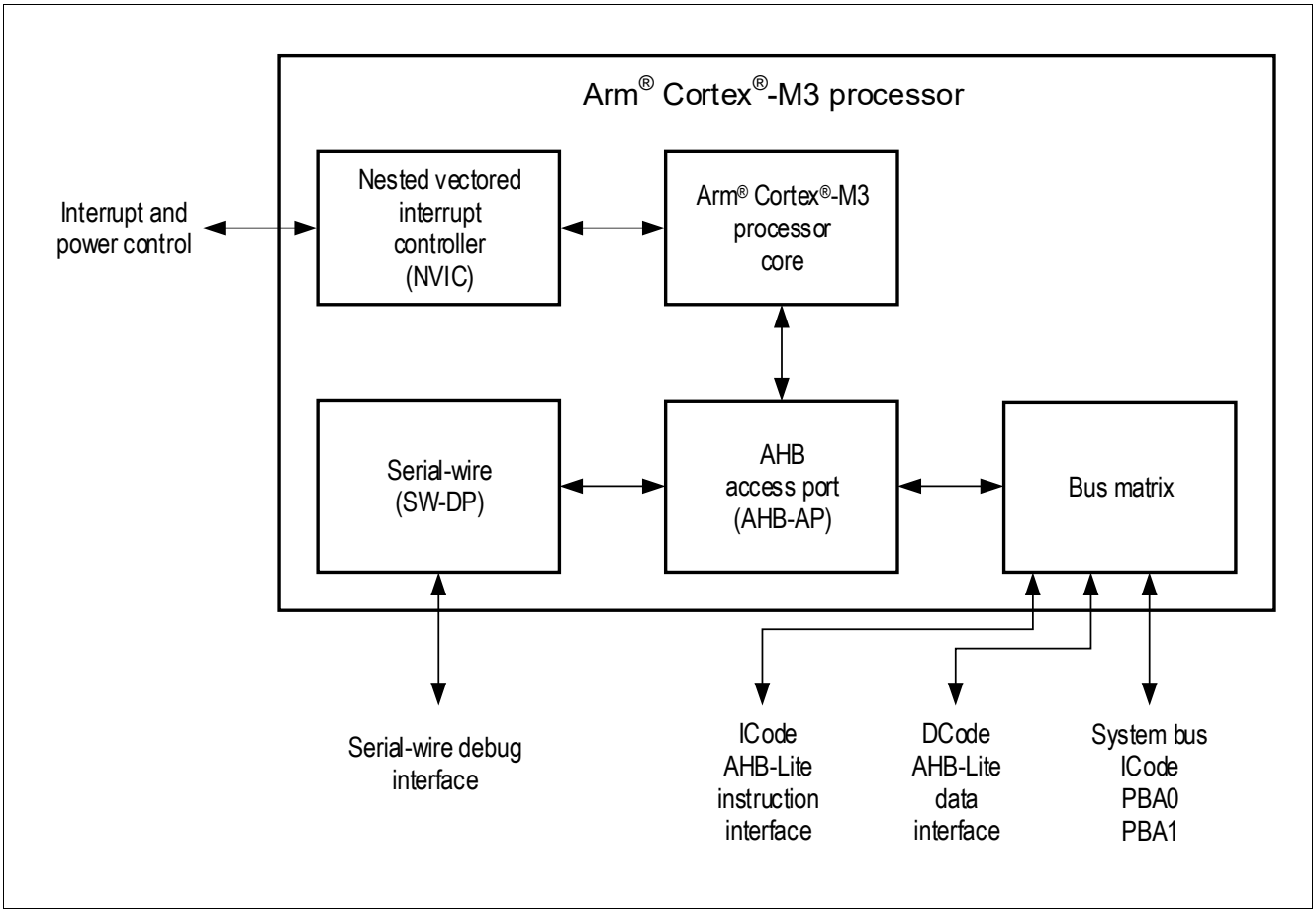


Figure 38 Arm® Cortex®-M3 block diagram

8.3 Functional Description

8.3.1 Registers

The processor has the following 32-bit registers:

- 13 general-purpose registers, R0-R12
- Stack pointer (SP), R13 alias of banked registers, SP_process and SP_main
- Link register (LR), R14
- Program counter (PC), R15
- Special-purpose program status registers (xPSR)

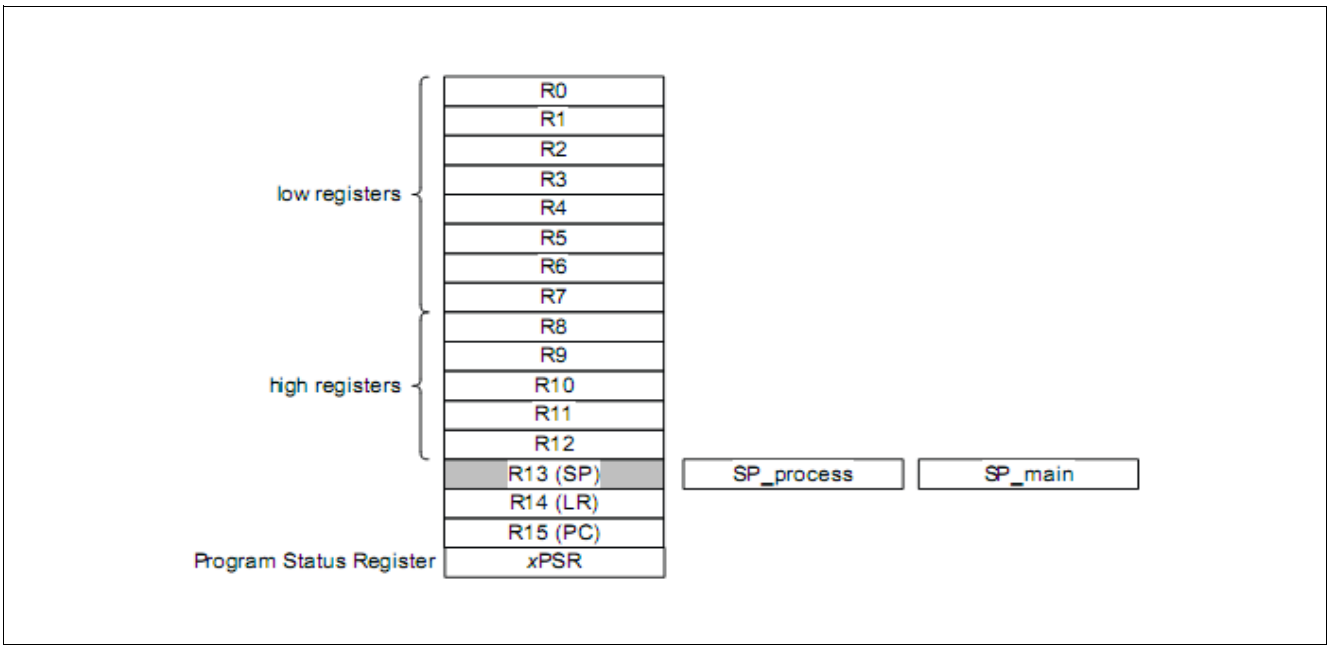


Figure 39 Processor Register Set

8.3.1.1 General-Purpose Registers

The general-purpose registers R0-R12 have no special architecturally-defined uses. Most instructions that can specify a general-purpose register can specify R0-R12.

Low Registers

Registers R0-R7 are accessible by all instructions that specify a general-purpose register.

High Registers

Registers R8-R12 are accessible by all 32-bit instructions that specify a general-purpose register.

Registers R8-R12 are not accessible by all 16-bit instructions.

Registers R13, R14, and R15 have the following special functions:

Stack Pointer

Register R13 is used as Stack Pointer (SP).

Link Register

Register R14 is the subroutine Link Register (LR).

Program counter

Register R15 is the Program Counter (PC).

8.3.1.2 Special-Purpose Program Status Registers (xPSR)

Processor status at the system level breaks down into three categories:

- Application PSR
- Interrupt PSR
- Execution PSR

They can be accessed as individual registers, a combination of any two from three, or a combination of all three using the Move to Register from Status (MRS) and MSR instructions.

8.4 Summary of Processor Registers

The processor has the following 32-bit registers that control functionality:

Table 44 Register Address Space for Processor Registers

Module	Base Address	End Address	Note
CPU	$E000E000_H$	$E000EFFF_H$	Arm® Cortex®-M3 Core SCS (System Control Space), Systick, NVICProcessor Registers

Table 45 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Summary of Processor Registers			
ICT	Interrupt Controller Type Register	004_H	$0000\ 0000_H$
SYSTICK_CS	SysTick Control and Status Register	010_H	$0000\ 0000_H$
SYSTICK_RL	SysTick Reload Value Register	014_H	$00XX\ XXXX_H$
SYSTICK_CUR	SysTick Current Value Register	018_H	$00XX\ XXXX_H$
SYSTICK_CAL	SysTick Calibration Value Register	$01C_H$	XX00 0000 XXXX XXXX XXXX XXXX XXXX_B
NVIC_ISER0	Interrupt Set-Enable Register	100_H	$0000\ 0000_H$
NVIC_ICER0	Interrupt Clear-Enable Register	180_H	$0000\ 0000_H$
NVIC_ISPR0	Interrupt Set-Pending Register	200_H	$0000\ 0000_H$
NVIC_ICPR0	Interrupt Clear-Pending Register	280_H	$0000\ 0000_H$
NVIC_IABR0	Active Bit Register	300_H	$0000\ 0000_H$
NVIC_IPR0	Interrupt Priority Register	400_H	$0000\ 0000_H$
NVIC_IPR1	Interrupt Priority Register	404_H	$0000\ 0000_H$
NVIC_IPR2	Interrupt Priority Register	408_H	$0000\ 0000_H$
NVIC_IPR3	Interrupt Priority Register	$40C_H$	$0000\ 0000_H$
CPUID	CPU ID Base Register	$D00_H$	$412F\ C231_H$
ICSR	Interrupt Control State Register	$D04_H$	$0000\ 0000_H$
VTOR	Vector Table Offset Register	$D08_H$	$0000\ 0000_H$
AIRCR	Application Interrupt/Reset Control Register	$D0C_H$	$0000\ 0000_H$
SCR	System Control Register	$D10_H$	$0000\ 0000_H$
CCR	Configuration Control Register	$D14_H$	$0000\ 0200_H$
SHPR1	System Handler Priority Register 1	$D18_H$	$0000\ 0000_H$
SHPR2	System Handler Priority Register 2	$D1C_H$	$0000\ 0000_H$
SHPR3	System Handler Priority Register 3	$D20_H$	$0000\ 0000_H$
SHCSR	System Handler Control and State Register	$D24_H$	$0000\ 0000_H$
CFSR	Configurable Fault Status Register	$D28_H$	$0000\ 0000_H$

Table 45 Register Overview (cont'd)

Register Short Name	Register Long Name	Offset Address	Reset Value
HFSR	Hard Fault Status Register	$D2C_H$	$0000\ 0000_H$
DFSR	Debug Fault Status Register	$D30_H$	$0000\ 0000_H$
MMFAR	MemManage Fault Status Register	$D34_H$	$XXXX\ XXXX_H$
BFAR	Bus Fault Status Register	$D38_H$	$XXXX\ XXXX_H$
AFSR	Auxiliary Fault Status Register	$D3C_H$	$0000\ 0000_H$

The registers are addressed wordwise.

Interrupt Controller Type Register

ICT
Interrupt Controller Type **(004_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											**INTLINESNUM**				
r											r				

Field	Bits	Type	Description
RES	31:5	r	**Reserved**
INTLINESNUM	4:0	r	**Interrupt Lines** Total number of interrupt lines in groups of 32: 00000_B , 0 to 32[1)] 00001_B , 33 to 64 00010_B , 65 to 95 00011_B , 97 to 128 00100_B , 129 to 160 00101_B , 161 to 192 00110_B , 193 to 224 00111_B , 225 to 256[1)]

1) The processor only supports between 1 and 240 interrupts.

SysTick Control and Status Register

SYSTICK_CS
SysTick Control and Status **(010_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															COUNTFLAG
							r								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES													CLKSOURCE	TICKINT	ENABLE
						r							rw	rw	rw

Field	Bits	Type	Description
RES	31:17	r	**Reserved**
COUNTFLAG	16	rw	**Count Flag** Returns 1 if timer counted to 0 since last time this was read. Clears on read by application of any part of the SysTick Control and Status Register. If read by the debugger using the SWD, this bit is cleared on read-only if the MasterType bit in the AHB-AP Control Register is set to 0. Otherwise, the COUNTFLAG bit is not changed by the debugger read.
RES	15:3	r	**Reserved**
CLKSOURCE	2	rw	**CLK Source** If no reference clock is provided, it is held at 1 and gives the same time as the core clock. The core clock must be at least 2.5 times faster than the reference clock. If it is not, the count values are unpredictable. 0_B **EXTCLK**, external reference clock (STCLK: from $f_{sys}/4$) 1_B **HCLK**, core clock (HCLK)
TICKINT	1	rw	**SysTick underflow interrupt** 0_B **DISABLE** , counting down to 0 does not pend the SysTick handler. Software can use the COUNTFLAG to determine if ever counted to 0. 1_B **ENABLE** , counting down to 0 pends the SysTick handler.
ENABLE	0	rw	**SysTick enable** 0_B **DISABLE** , counter disabled. 1_B **ENABLE**, counter operates in a multi-shot way. That is, counter loads with the Reload value and then begins counting down. On reaching 0, it sets the COUNTFLAG to 1 and optionally pends the SysTick handler, based on TICKINT. It then loads the Reload value again, and begins counting.

SysTick Reload Value Register

SYSTICK_RL
SysTick Reload Value **(014_H)** **Reset Value: $00XX\ XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								RELOAD							
r								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RELOAD															
rw															

Field	Bits	Type	Description
RES	31:24	r	**Reserved**
RELOAD	23:0	rw	**Reload** Value to load into the SysTick Current Value Register when the counter reaches 0.

SysTick Current Value Registers

SYSTICK_CUR
SysTick Current Value **(018_H)** **Reset Value: $00XX\ XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								CURRENT							
r								rwc							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENT															
rwc															

Field	Bits	Type	Description
RES	31:24	r	**Reserved**
CURRENT	23:0	rwc	**Current** Current value at the time the register is accessed. No read-modify-write protection is provided, so change with care. This register is write-clear. Writing to it with any value clears the register to 0. Clearing this register also clears the COUNTFLAG bit of the SysTick Control and Status Register.

SysTick Calibration Value Registers

SYSTICK_CAL
SysTick Calibration Value **($01C_H$)** **Reset Value:**
XX00 0000 XXXX XXXX XXXX XXXX XXXX XXXX$_B$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOREF	**SKEW**	**RES**						**TENMS**							
r	r			r							r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TENMS															
								r							

Field	Bits	Type	Description
NOREF	31	r	**No Reference Clock** 0_B **reserved** , n.u. 1_B **NOREF** , the reference clock is not provided
SKEW	30	r	**Skew** 0_B **reserved** , n.u. 1_B **SKEW** , the calibration value is not exactly 10 ms because of clock frequency. This could affect its suitability as a software real time clock.
RES	29:24	r	**Reserved**
TENMS	23:0	r	**10 ms reload value** This value is the Reload value to use for 10ms timing. Depending on the value of SKEW, this might be exactly 10ms or might be the closest value. If this reads as 0, then the calibration value is not known. This is probably because the reference clock is an unknown input from the system or scalable dynamically.

Interrupt Set-Enable Registers

NVIC_ISER0

Interrupt Set-Enable **(100_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int_DMA	Int_BDRV	Int_EXINT1	Int_EXINT0	Int_UART2	Int_UART1	Int_SSC2	Int_SSC1	Int_CCU6SR3	Int_CCU6SR2	Int_CCU6SR1	Int_CCU6SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:16	r	**Reserved**
Int_DMA	15	rw	**Interrupt Set for DMA** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_BDRV	14	rw	**Interrupt Set for Bridge Driver** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_EXINT1	13	rw	**Interrupt Set for External Int 1** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_EXINT0	12	rw	**Interrupt Set for External Int 0** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_UART2	11	rw	**Interrupt Set for UART2** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_UART1	10	rw	**Interrupt Set for UART1** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_SSC2	9	rw	**Interrupt Set for SSC2** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_SSC1	8	rw	**Interrupt Set for SSC1** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_CCU6SR3	7	rw	**Interrupt Set for CCU6 SR3** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt

Field	Bits	Type	Description
Int_CCU6SR2	6	rw	**Interrupt Set for CCU6 SR2** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_CCU6SR1	5	rw	**Interrupt Set for CCU6 SR1** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_CCU6SR0	4	rw	**Interrupt Set for CCU6 SR0** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_ADC1	3	rw	**Interrupt Set for ADC1** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_ADC2	2	rw	**Interrupt Set for MU, ADC2** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_GPT2	1	rw	**Interrupt Set for GPT2** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt
Int_GPT1	0	rw	**Interrupt Set for GPT1** 0_B **DISABLED**, no effect on write 1_B **ENABLE**, enables the associated interrupt

Interrupt Clear-Enable Registers

NVIC_ICER0
Interrupt Clear-Enable **(180_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int_DMA	Int_BDRV	Int_EXINT1	Int_EXINT0	Int_UART2	Int_UART1	Int_SSC2	Int_SSC1	Int_CCU6SR3	Int_CCU6SR2	Int_CCU6SR1	Int_CCU6SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:16	r	**Reserved**
Int_DMA	15	rw	**Interrupt Clear for DMA** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_BDRV	14	rw	**Interrupt Clear for Bridge Driver** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_EXINT1	13	rw	**Interrupt Clear for External Int 1** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_EXINT0	12	rw	**Interrupt Clear for External Int 0** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_UART2	11	rw	**Interrupt Clear for UART2** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_UART1	10	rw	**Interrupt Clear for UART1** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled

Field	Bits	Type	Description
Int_SSC2	9	rw	**Interrupt Clear for SSC2** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_SSC1	8	rw	**Interrupt Clear for SSC1** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_CCU6SR3	7	rw	**Interrupt Clear for CCU6 SR3** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_CCU6SR2	6	rw	**Interrupt Clear for CCU6 SR2** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_CCU6SR1	5	rw	**Interrupt Clear for CCU6 SR1** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_CCU6SR0	4	rw	**Interrupt Clear for CCU6 SR0** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_ADC1	3	rw	**Interrupt Clear for ADC1** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_ADC2	2	rw	**Interrupt Clear for MU, ADC2** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled
Int_GPT2	1	rw	**Interrupt Clear for GPT2** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled

Field	Bits	Type	Description
Int_GPT1	0	rw	**Interrupt Clear for GPT1** 0_B **DISABLE**, on reads the associated interrupt is disabled, no effect on write 1_B **ENABLE**, on reads the associated interrupt is enabled, on writes the associated interrupt is disabled

Interrupt Set-Pending Registers

NVIC_ISPR0
Interrupt Set-Pending **(200_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int_DMA	Int_BDRV	Int_EXINT1	Int_EXINT0	Int_UART2	Int_UART1	Int_SSC2	Int_SSC1	Int_CCU6SR3	Int_CCU6SR2	Int_CCU6SR1	Int_CCU6SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:16	r	**Reserved**
Int_DMA	15	rw	**Interrupt Set Pending for DMA** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_BDRV	14	rw	**Interrupt Set Pending for Bridge Driver** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_EXINT1	13	rw	**Interrupt Set Pending for External Int 1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_EXINT0	12	rw	**Interrupt Set Pending for External Int 0** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_UART2	11	rw	**Interrupt Set Pending for UART2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_UART1	10	rw	**Interrupt Set Pending for UART1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_SSC2	9	rw	**Interrupt Set Pending for SSC2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending

Field	Bits	Type	Description
Int_SSC1	8	rw	**Interrupt Set Pending for SSC1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_CCU6SR3	7	rw	**Interrupt Set Pending for CCU6 SR3** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_CCU6SR2	6	rw	**Interrupt Set Pending for CCU6 SR2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_CCU6SR1	5	rw	**Interrupt Set Pending for CCU6 SR1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_CCU6SR0	4	rw	**Interrupt Set Pending for CCU6 SR0** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_ADC1	3	rw	**Interrupt Set Pending for ADC1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_ADC2	2	rw	**Interrupt Set Pending for MU, ADC2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_GPT2	1	rw	**Interrupt Set Pending for GPT2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending
Int_GPT1	0	rw	**Interrupt Set Pending for GPT1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, the associated interrupt is pending

Interrupt Clear-Pending Registers

NVIC_ICPR0
Interrupt Clear-Pending **(280_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int_DMA	Int_BDRV	Int_EXINT1	Int_EXINT0	Int_UART2	Int_UART1	Int_SSC2	Int_SSC1	Int_CCU6SR3	Int_CCU6SR2	Int_CCU6SR1	Int_CCU6SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:16	r	**Reserved**
Int_DMA	15	rw	**Interrupt Clear Pending for DMA** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_BDRV	14	rw	**Interrupt Clear Pending for Bridge Driver** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_EXINT1	13	rw	**Interrupt Clear Pending for External Int 1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_EXINT0	12	rw	**Interrupt Clear Pending for External Int 0** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_UART2	11	rw	**Interrupt Clear Pending for UART2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_UART1	10	rw	**Interrupt Clear Pending for UART1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending

Field	Bits	Type	Description
Int_SSC2	9	rw	**Interrupt Clear Pending for SSC2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_SSC1	8	rw	**Interrupt Clear Pending for SSC1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_CCU6SR3	7	rw	**Interrupt Clear Pending for CCU6 SR3** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_CCU6SR2	6	rw	**Interrupt Clear Pending for CCU6 SR2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_CCU6SR1	5	rw	**Interrupt Clear Pending for CCU6 SR1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_CCU6SR0	4	rw	**Interrupt Clear Pending for CCU6 SR0** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_ADC1	3	rw	**Interrupt Clear Pending for ADC1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_ADC2	2	rw	**Interrupt Clear Pending for MU, ADC2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending
Int_GPT2	1	rw	**Interrupt Clear Pending for GPT2** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending

Field	Bits	Type	Description
Int_GPT1	0	rw	**Interrupt Clear Pending for GPT1** 0_B **Not Pending**, on reads the associated interrupt is not pending, no effect on writes 1_B **Pending**, on reads the associated interrupt is pending, on writes the status of the associated interrupt is changed to not pending

Active Bit Register

NVIC_IABR0
Active Bit Register Interrupt Active Flags **(300_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Int_DMA	Int_BDRV	Int_EXINT1	Int_EXINT0	Int_UART2	Int_UART1	Int_SSC2	Int_SSC1	Int_CCU6SR3	Int_CCU6SR2	Int_CCU6SR1	Int_CCU6SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Field	Bits	Type	Description
RES	31:16	r	**Reserved**
Int_DMA	15	r	**Interrupt Active for DMA** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_BDRV	14	r	**Interrupt Active for Bridge Driver** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_EXINT1	13	r	**Interrupt Active for External Int 1** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_EXINT0	12	r	**Interrupt Active for External Int 0** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_UART2	11	r	**Interrupt Active for UART2** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_UART1	10	r	**Interrupt Active for UART1** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_SSC2	9	r	**Interrupt Active for SSC2** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation

Field	Bits	Type	Description
Int_SSC1	8	r	**Interrupt Active for SSC1** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_CCU6SR3	7	r	**Interrupt Active for CCU6 SR3** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_CCU6SR2	6	r	**Interrupt Active for CCU6 SR2** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_CCU6SR1	5	r	**Interrupt Active for CCU6 SR1** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_CCU6SR0	4	r	**Interrupt Active for CCU6 SR0** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_ADC1	3	r	**Interrupt Active for ADC1** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_ADC2	2	r	**Interrupt Active for MU, ADC2** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_GPT2	1	r	**Interrupt Active for GPT2** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation
Int_GPT1	0	r	**Interrupt Active for GPT1** 0_B **DISABLE**, disables interrupt for read operation, no effect for write operation 1_B **ENABLE**, enables interrupt for read and write operation

Interrupt Priority Registers

For the Interrupt Priority Registers, only the upper nibble of each priority bytes is significant.

NVIC_IPR0
Interrupt Priority **(400_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_ADC1				RES				PRI_ADC2				RES			
rw				r				rw				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_GPT2				RES				PRI_GPT1				RES			
rw				r				rw				r			

Field	Bits	Type	Description
PRI_ADC1	31:28	rw	**Priority for ADC1** 0: highest priority, .., 15: lowest priority
RES	27:24	r	**Reserved**
PRI_ADC2	23:20	rw	**Priority for MU, ADC2** 0: highest priority, .., 15: lowest priority
RES	19:16	r	**Reserved**
PRI_GPT2	15:12	rw	**Priority for GPT2** 0: highest priority, .., 15: lowest priority
RES	11:8	r	**Reserved**
PRI_GPT1	7:4	rw	**Priority for GPT1** 0: highest priority, .., 15: lowest priority
RES	3:0	r	**Reserved**

Interrupt Priority Registers

For the Interrupt Priority Registers, only the upper nibble of each priority bytes is significant.

NVIC_IPR1
Interrupt Priority **(404_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_CCU6SR3				RES				PRI_CCU6SR2				RES			
rw				r				rw				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_CCU6SR1				RES				PRI_CCU6SR0				RES			
rw				r				rw				r			

Field	Bits	Type	Description
PRI_CCU6SR3	31:28	rw	**Priority for CCU6 SR3** 0: highest priority, .., 15: lowest priority
RES	27:24	r	**Reserved**
PRI_CCU6SR2	23:20	rw	**Priority for CCU6 SR2** 0: highest priority, .., 15: lowest priority
RES	19:16	r	**Reserved**
PRI_CCU6SR1	15:12	rw	**Priority for CCU6 SR1** 0: highest priority, .., 15: lowest priority
RES	11:8	r	**Reserved**
PRI_CCU6SR0	7:4	rw	**Priority for CCU6 SR0** 0: highest priority, .., 15: lowest priority
RES	3:0	r	**Reserved**

Interrupt Priority Registers

For the Interrupt Priority Registers, only the upper nibble of each priority bytes is significant.

NVIC_IPR2
Interrupt Priority **(408_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_UART2				RES				PRI_UART1				RES			
rw				r				rw				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_SSC2				RES				PRI_SSC1				RES			
rw				r				rw				r			

Field	Bits	Type	Description
PRI_UART2	31:28	rw	**Priority for UART2** 0: highest priority, .., 15: lowest priority
RES	27:24	r	**Reserved**
PRI_UART1	23:20	rw	**Priority for UART1** 0: highest priority, .., 15: lowest priority
RES	19:16	r	**Reserved**
PRI_SSC2	15:12	rw	**Priority for SSC2** 0: highest priority, .., 15: lowest priority
RES	11:8	r	**Reserved**
PRI_SSC1	7:4	rw	**Priority for SSC1** 0: highest priority, .., 15: lowest priority
RES	3:0	r	**Reserved**

Interrupt Priority Registers

For the Interrupt Priority Registers, only the upper nibble of each priority bytes is significant.

NVIC_IPR3
Interrupt Priority **($40C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_DMA				RES				PRI_BDRV				RES			
rw				r				rw				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_EXINT1				RES				PRI_EXINT0				RES			
rw				r				rw				r			

Field	Bits	Type	Description
PRI_DMA	31:28	rw	**Priority for DMA** 0: highest priority, .., 15: lowest priority
RES	27:24	r	**Reserved**
PRI_BDRV	23:20	rw	**Priority for Bridge Driver** 0: highest priority, .., 15: lowest priority
RES	19:16	r	**Reserved**
PRI_EXINT1	15:12	rw	**Priority for Ext. Int 1** 0: highest priority, .., 15: lowest priority
RES	11:8	r	**Reserved**
PRI_EXINT0	7:4	rw	**Priority for Ext. Int 0** 0: highest priority, .., 15: lowest priority
RES	3:0	r	**Reserved**

CPU ID Base Register

CPUID
CPU ID Base Register **($D00_H$)** **Reset Value: $412F\ C231_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER								VARIANT				ARCHITECTURE			
r								r				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARTNO												REVISION			
r												r			

Field	Bits	Type	Description
IMPLEMENTER	31:24	r	**Implementer Code** Assigned by Arm®. Read as 41_H for a processor implemented by Arm®.
VARIANT	23:20	r	**Variant Number** Implementation defined.
ARCHITECTURE	19:16	r	**Architecture** Read as F_H.
PARTNO	15:4	r	**Part Number** Implementation defined.
REVISION	3:0	r	**Revision Number** Implementation defined.

Interrupt Control State Register

ICSR
Interrupt Control State Register **($D04_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NMIPENDSET	RES		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	RES	ISRPREEMPT	ISRPENDING	RES				VECTPENDING	
rw	r		rw	w	rw	w	r	r	r	r				r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VECTPENDING				RETTOBASE	RES		VECTACTIVE								
r				r	r		r								

Field	Bits	Type	Description
NMIPENDSET	31	rw	**NMI set pending bit** On writes, makes the NMI exception active. On reads, indicates the state of the exception. *Note: Because NMI is higher priority than other exceptions, if the processor is not already executing the NMI handler, it enters the NMI exception handler as soon as it recognizes the write to this bit.* 0_B **value0**, write has no effect. On reads, NMI is not pending. 1_B **value1**, on writes, makes the NMI exception pending. On reads, NMI is pending.
RES	30:29	r	**Reserved**
PENDSVSET	28	rw	**PendSV set pending bit** On writes, sets the PendSV exception as pending. On reads, indicates the current state of the exception. *Note: Normally, software writes 1 to this bit to request a context switch.* 0_B **value0** , write has no effect. On reads, PendSV is not pending. 1_B **value1** , on writes, makes PendSV exception pending. On reads, PendSV is pending.
PENDSVCLR	27	w	**PendSV clear pending bit** Removes the pending status of the PendSV exception 0_B **value0**, no effect 1_B **value1**, remove pending status

Field	Bits	Type	Description
PENDSTSET	26	rw	**SysTick exception set pending bit** On writes, sets the SysTick exception as pending. On reads, indicates the current state of the exception. 0_B **value0**, write has no effect. On reads, SysTick is not pending. 1_B **value1**, on writes, makes SysTick exception pending. On reads, SysTick is pending.
PENDSTCLR	25	w	**SysTick exception clear pending bit** Removes the pending status of the SysTick exception. 0_B **value0**, no effect 1_B **value1**, remove pending status
RES	24	r	**Reserved**
ISRPREEMPT	23	r	**ISRPREEMT (only used at debug time)** Indicates whether a pending interrupt will be serviced on exit from debug halt state. 0_B **value0**, will not service 1_B **value1**, will service a pending exception
ISRPENDING	22	r	**Interrupt pending flag, excluding NMI and Faults** Indicates whether an interrupt, generated by the NVIC, is pending. 0_B **value0**, interrupt is not pending 1_B **value1**, interrupt is pending
RES	21:18	r	**Reserved**
VECTPENDING	17:12	r	**Vector of the pending exception** The exception number of the highest priority pending exception. A value of 0 indicates that there is no pending exception. 0_B **value0**, no pending exceptions
RETTOBASE	11	r	**RETTOBASE** In Handler Mode, indicates whether there is an active exception other than the exception indicated by the current value of the ISPR. *Note: In Thread Mode the value of this bit is undefined.* 0_B **value0**, There is an active exception other than the exception shown by IPSR. 1_B **value1**, There is no active exception other than any exception shown by IPSR.
RES	10:9	r	**Reserved**
VECTACTIVE	8:0	r	**Active vector** The exception number of the current executing exception. A value of 0 indicates that the processor is in Thread Mode. 0_B **value0**, Thread Mode

Vector Table Offset Register

VTOR
Vector Table Offset Register **($D08_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TBLOFF															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBLOFF									**RES**						
rw									r						

Field	Bits	Type	Description
TBLOFF	31:7	rw	**Vector Table Offset** Bits[31:7] of the vector table address.
RES	6:0	r	**Reserved**

Application Interrupt/Reset Control Register

AIRCR
Application Interrupt/Reset Control Register (D0C$_H$) **Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
VECTKEY															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANNESS	RES				PRIGROUP			RES					SYSRESETREQ	VECTCLRACTIVE	VECTRESET
r	r				rw			r					rh	w	w

Field	Bits	Type	Description
VECTKEY	31:16	rw	**Vector Key** Register writes must write 05FA$_H$ to this field, otherwise the write is ignored. On reads, returns FA05$_H$.
ENDIANNESS	15	r	**Memory System Endianness** This bit is static or configured by hardware input on reset. 0_B **value0**, little endian 1_B **value1**, big endian
RES	14:11	r	**Reserved**
PRIGROUP	10:8	rw	**Priority Grouping** Indicates the binary point position. This bit resets to 000_B
RES	7:3	r	**Reserved**
SYSRESETREQ	2	rh	**System Reset Request** Writing 1 to this bit asserts a signal to the external system to request a local reset. A Local or Power-on reset clears this bit to 0. 0_B **value0**, do not request a reset 1_B **value1**, request a reset
VECTCLRACTIVE	1	w	**VECTCLRACTIVE (for debug use only)** When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable. *Note: The effect of writing a 1 to this bit, if the processor is not halted in Debug state, is* **unpredictable**.

Field	Bits	Type	Description
VECTRESET	0	w	**VECTRESET (reserved for debug use)** When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable. *Note: The effect of writing a 1 to this bit, if the processor is not halted in Debug state, is* **unpredictable**. *When the processor is halted in Debug state, if a write to the register writes a 1 to both VECTRESET and SYSRESETREQ, the behavior is* **unpredictable**.

System Control Register

SCR
System Control Register **($D10_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											SEVONPEND	RES	SLEEPDEEP	SLEEPONEXIT	RES
r											rw	r	rw	rw	r

Field	Bits	Type	Description
RES	31:5	r	**Reserved**
SEVONPEND	4	rw	**Send Event on Pending bit** Determines whether an interrupt transition from inactive state to pending state is a wake-up event. 0_B **value0**, transitions from inactive to pending are not wake-up events 1_B **value1**, transitions from inactive to pending are wake-up events
RES	3	r	**Reserved**
SLEEPDEEP	2	rw	**Sleep Deep** Provides a qualifying hint that waking from sleep might take longer. An implementation can use this bit to select between two alternative sleep states. *Note: Details of the implemented sleep states, if any, and details of the use of this bit, are implementation defined. If the processor does not implement a deep sleep state then this bit can be RAZ/WI.* 0_B **value0**, selected sleep state is not deep sleep 1_B **value1**, selected sleep state is deep sleep
SLEEPONEXIT	1	rw	**Sleep on Exit** Indicates sleep-on-exit when returning from Handler mode to Thread mode 0_B **value0**, do not sleep when returning to Thread mode 1_B **value1**, enter sleep, or deep sleep, on return from an ISR
RES	0	r	**Reserved**

Configuration Control Register

CCR
Configuration Control Register **($D14_H$)** **Reset Value: $0000\ 0200_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						STKALIGN	BFHFMIGN	RES			DIV_0_TRP	UNALIGN_TRP	RES	USERSETMPEND	NONBASETHRDENA
r						rw	rw	r			rw	rw	r	rw	rw

Field	Bits	Type	Description
RES	31:10	r	**Reserved**
STKALIGN	9	rw	**Stack alignment** Determines whether the exception entry sequence guarantees 8-byte stack frame alignment, adjusting the SP if necessary before saving state. 0_B **value0** , 4-byte aligned 1_B **value1**, 8-byte aligned
BFHFMIGN	8	rw	**BFHFMIGN** Enables handlers with priority -1 or -2 to ignore data BusFaults caused by load and store instructions. This applies to the hard fault, NMI, and FAULTMASK escalated handlers. 0_B **value0**, data bus faults caused by load and store instructions cause a lock-up 1_B **value1**, handlers running at priority -1 and -2 ignore data bus faults caused by load and store instructions
RES	7:5	r	**Reserved**
DIV_0_TRP	4	rw	**Division on Zero trap** Enables faulting or halting when the processor executes an SDIV or UDIV instruction with a divisor of 0. When this bit is set to 0, a divide by zero returns a quotient of 0. 0_B **value0**, trapping disabled 1_B **value1**, trapping enabled
UNALIGN_TRP	3	rw	**Unalignment trap** Controls the trapping of unaligned word or halfword accesses. *Note: Unaligned load-store multiples and word or halfword exclusive accesses always fault.* 0_B **value0**, trapping disabled 1_B **value1**, trapping enabled
RES	2	r	**Reserved**

Field	Bits	Type	Description
USERSETMPEND	1	rw	**USERSETMPEND** Controls whether unprivileged software can access the STIR. 0_B **value0**, unprivileged software cannot access the STIR. 1_B **value1**, unprivileged software can access the STIR.
NONBASETHRDENA	0	rw	**NONBASETHRDENA** Controls whether the processor can enter Thread Mode at an execution priority level other than base level. 0_B **value0**, any attempt to enter Thread Mode at an execution priority level of other than base level faults. 1_B **value1**, the processor can enter Thread Mode at any execution priority level because of a controlled return value.

System Handler Priority Register 1

SHPR1
System Handler Priority Register 1 **($D18_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_7								**PRI_6**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_5								**PRI_4**							
rw								rw							

Field	Bits	Type	Description
PRI_7	31:24	rw	**Reserved for Priority of System Handler 7**
PRI_6	23:16	rw	**Priority of System Handler 6, UsageFault**
PRI_5	15:8	rw	**Priority of System Handler 5, BusFault**
PRI_4	7:0	rw	**Priority of System Handler 4, MemManage**

System Handler Priority Register 2

SHPR2
System Handler Priority Register 2 **($D1C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_11								PRI_10							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_9								PRI_8							
rw								rw							

Field	Bits	Type	Description
PRI_11	31:24	rw	**Priority of System Handler 11, SVCall**
PRI_10	23:16	rw	**Reserved for Priority of System Handler 10**
PRI_9	15:8	rw	**Reserved for Priority of System Handler 9**
PRI_8	7:0	rw	**Reserved for Priority of System Handler 8**

System Handler Priority Register 3

SHPR3
System Handler Priority Register 3 **($D20_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_15								PRI_14							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_13								PRI_12							
rw								rw							

Field	Bits	Type	Description
PRI_15	31:24	rw	**Priority of System Handler 15, SysTick**
PRI_14	23:16	rw	**Priority of System Handler 14, PendSV**
PRI_13	15:8	rw	**Reserved for Priority of System Handler 13**
PRI_12	7:0	rw	**Priority of System Handler 12, DebugMonitor**

System Handler Control and State Register

SHCSR
System Handler Control and State Register **($D24_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													USGFAULTENA	BUSFAULTENA	MEMFAULTENA
						r							rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SVCALLPENDED	BUSFAULTPENDED	MEMFAULTPENDED	USGFAULTPENDED	SYSTICKACT	PENDSVACT	RES	MONITORACT	SVCALLACT	RES			USGFAULTACT	RES	BUSFAULTACT	MEMFAULTACT
rw	rw	rw	rw	rw	rw	r	rw	rw	r			rw	r	rw	rw

Field	Bits	Type	Description
RES	31:19	r	**Reserved**
USGFAULTENA	18	rw	**UsageFault enable bit** 0_B **disable**, Disable UsageFault 1_B **enable**, Enable UsageFault
BUSFAULTENA	17	rw	**BusFault enable bit** 0_B **disable**, Disable BusFault 1_B **enable**, Enable BusFault
MEMFAULTENA	16	rw	**MemManage enable bit** 0_B **disable**, Disable MemManage fault 1_B **enable**, Enable MemManage fault
SVCALLPENDED	15	rw	**SVCall pending bit**[1)] 0_B **notpending**, SVCall is not pending 1_B **pending**, SVCall is pending
BUSFAULTPENDED	14	rw	**BusFault exception pending bit**[1)] 0_B **notpending**, BusFault is not pending 1_B **pending**, BusFault is pending
MEMFAULTPENDED	13	rw	**MemManage exception pending bit**[1)] 0_B **notpending**, MemManage is not pending 1_B **pending**, MemManage is pending
USGFAULTPENDED	12	rw	**UsageFault exception pending bit**[1)] 0_B **notpending**, UsageFault is not pending 1_B **pending**, UsageFault is pending
SYSTICKACT	11	rw	**SysTick exception active bit**[2)] 0_B **inactive**, SysTick is not active 1_B **active**, SysTick is active
PENDSVACT	10	rw	**PendSV exception active bit**[2)] 0_B **inactive**, PendSV is not active 1_B **active**, PendSV is active

Field	Bits	Type	Description
RES	9	r	**Reserved**
MONITORACT	8	rw	**Debug monitor active bit**[2)] 0_B **inactive**, Monitor is not active 1_B **active**, Monitor is active
SVCALLACT	7	rw	**SVCall active bit**[2)] 0_B **inactive**, SVCall is not active 1_B **active**, SVCall is active
RES	6:4	r	**Reserved**
USGFAULTACT	3	rw	**UsageFault exception active bit**[2)] 0_B **inactive**, UsageFault is not active 1_B **active**, UsageFault is active
RES	2	r	**Reserved**
BUSFAULTACT	1	rw	**BusFault exception active bit**[2)] 0_B **inactive**, BusFault is not active 1_B **active**, BusFault is active
MEMFAULTACT	0	rw	**MemManage exception active bit**[2)] 0_B **inactive**, MemManage is not active 1_B **active**, MemManage is active

1) Pending state bits are set to 1 when an exception occurs, and are cleared to 0 when the exception becomes active.
2) Active state bits are set to 1 if the associated exception is the current exception or an exception that is nested because of preemption.

Configurable Fault Status Register

CFSR
Configurable Fault Status Register **($D28_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES						DIVBYZERO	UNALIGNED	RES				NOCP	INVPC	INVSTATE	UNDEFINSTR
r						rw	rw	r				rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BFARVALID	RES		STKERR	UNSTKERR	IMPRECISERR	PRECISERR	IBUSERR	MMARVALID	RES		MSTKERR	MUNSTKERR	RES	DACCVIOL	IACCVIOL
rw	r		rw	rw	rw	rw	rw	rw	r		rw	rw	r	rw	rw

Field	Bits	Type	Description
RES	31:26	r	**Reserved**
DIVBYZERO	25	rw	**Divide by zero UsageFault** *Note: When SDIV or UDIV instruction is used with a divisor of 0, this fault occurs if DIV_0_TRP is enabled in the CCR.* 0_B **value0**, No Divide by zero error has occurred. 1_B **value1**, A divide by zero error has occurred.
UNALIGNED	24	rw	**Unaligned access UsageFault** *Note: Multi-word accesses always fault if not word aligned. Software can configure unaligned word and halfword accesses to fault.* 0_B **value0**, No unaligned access error has occurred. 1_B **value1**, A unaligned access error has occurred.
RES	23:20	r	**Reserved**
NOCP	19	rw	**No Coprocessor UsageFault (not implemented)** 0_B **value0**, No coprocessor access error has occurred. 1_B **value1**, A coprocessor access error has occurred.
INVPC	18	rw	**Invalid PC load UsageFault** 0_B **value0**, No integrity check error has occurred. 1_B **value1**, the processor has attempted an illegal load of EXC_RETURN to the PC, as a result of an invalid context, or an invalid EXC_RETURN value.
INVSTATE	17	rw	**Invalid state UsageFault** 0_B **value0**, EPSR.T bit and EPSR.IT bits are valid for instruction execution. 1_B **value1**, Instruction executed with invalid EPSR.T or EPSR.IT field.

Field	Bits	Type	Description
UNDEFINSTR	16	rw	**Undefined Instruction UsageFault** 0_B **value0**, No Undefined Instruction Usage fault has occurred. 1_B **value1**, The processor hat attempted to execute an undefined instruction.
BFARVALID	15	rw	**BusFault Address Register (BFAR) valid flag** 0_B **invalid**, BFAR does not have valid contents. 1_B **valid**, BFAR has valid contents.
RES	14:13	r	**Reserved**
STKERR	12	rw	**BusFault on stacking for exception entry** 0_B **value0**, No derived bus fault occurred 1_B **value1**, A derived bus fault occurred on exception entry
UNSTKERR	11	rw	**BusFault on unstacking for a return from exception** 0_B **value0**, No derived bus fault occurred 1_B **value1**, A derived bus fault occurred on exception return
IMPRECISERR	10	rw	**Imprecise data bus error** 0_B **value0**, No precise data access error has occurred 1_B **value1**, An imprecise data access error has occurred.
PRECISERR	9	rw	**Precise data bus error** 0_B **value0**, No precise data access error has occurred 1_B **value1**, An imprecise data access error has occurred, and the processor has written the faulting address to the BFAR.
IBUSERR	8	rw	**Instruction bus error** 0_B **value0**, No bus fault on an instruction prefetch has occurred. 1_B **value1**, A bus fault on an instruction prefetch has occurred. The fault is signalled only if the instruction is issued.
MMARVALID	7	rw	**MemManage Fault Address Register (MMFAR) valid flag** 0_B **not valid**, MMAR does not have valid contents. 1_B **valid**, MMAR has valid contents.
RES	6:5	r	**Reserved**
MSTKERR	4	rw	**MemManage fault on stacking for exception entry** 0_B **value0**, No derived MemManage fault occurred 1_B **value1**, A derived MemManage fault occurred on exception entry
MUNSTKERR	3	rw	**MemManage fault on unstacking for a return from exception** 0_B **value0**, No derived MemManage fault occurred 1_B **value1**, A derived MemManage fault occurred on exception return
RES	2	r	**Reserved**
DACCVIOL	1	rw	**Data access violation flag** 0_B **value0**, No data access violation has occurred. 1_B **value1**, Data access violation. The MMAR shows the data address that the load or store tried to access.

Field	Bits	Type	Description
IACCVIOL	0	rw	**Instruction access violation flag** 0_B **value0**, no instruction access violation fault 1_B **value1**, MPU or Execute Never (XN) default memory map access violation on an instruction fetch has occurred. The fault is signalled only if the instruction is issued.

Notes

1. *USAGEFAULT contains bits 31:16 of CFSR and provides information on UsageFault exceptions*
2. *BUSFAULT contains bits 15:8 of CFSR and provides information on BusFault exceptions*
3. *MemManage contains bits 7:0 of CFSR and provides information on MemManage exceptions*

Hard Fault Status Register

HFSR
Hard Fault Status Register **($D2C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DEBUGEVT	FORCED	RES													
rw	rw								r						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES														VECTTBL	RES
							r							rw	r

Field	Bits	Type	Description
DEBUGEVT	31	rw	**Debug Event (reserved for debug use)** Indicates when a Debug event has occurred. When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable. *Note: The processor sets this bit to 1 only when halting debug is disabled and a Debug event occurs.* 0_B **value0**, No Debug event has occurred 1_B **value1**, Debug event has occurred. The Debug Fault Status Register has been updated.
FORCED	30	rw	**Forced Hard Fault** Indicates that a fault with configurable priority has been escalated to a HardFault exception, because it could not be made active, because of priority or because it was disabled. 0_B **value0**, No priority escalation has occurred 1_B **value1**, Processor has escalated a configurable priority exception to HardFault
RES	29:2	r	**Reserved**
VECTTBL	1	rw	**Vector Table Fault** Indicates when a fault has occurred because of a vector table read error on exception processing. 0_B **value0**, No vector table read fault has occurred 1_B **value1**, Vector table read fault has occurred
RES	0	r	**Reserved**

Debug Fault Status Register

DFSR
Debug Fault Status Register **($D30_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											EXTERNAL	VCATCH	DWTTRAP	BKPT	HALTED
r											rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:5	r	**Reserved**
EXTERNAL	4	rw	**External** Indicates a debug event generated because of the assertion of **EDBGRQ**. 0_B **value0**, No EDBGRQ debug event 1_B **value1**, EDBGRQ debug event
VCATCH	3	rw	**Vector Catch** Indicates triggering of a vector catch. The corresponding FSR shows the primary cause of exception. 0_B **value0**, No vector catch triggered 1_B **value1**, Vector catch triggered
DWTTRAP	2	rw	**DWTTRAP** Indicates a debug event generated by the DWT. 0_B **value0**, No current debug event generated by the DWT 1_B **value1**, At least one current debug event generated by the DWT
BKPT	1	rw	**BKPT** Indicates a debug event generated by BKPT instruction execution or a breakpoint match in FPB. 0_B **value0**, No current breakpoint debug event 1_B **value1**, At least one current breakpoint debug event
HALTED	0	rw	**HALTED** Indicates a debug event generated by either a C_HALT or C_STEP request, triggered by a write to the DHCSR, or a step request triggered by setting DEMCR.MON_STEP to 1. 0_B **value0**, No active halt request debug event 1_B **value1**, Halt request debug event active

MemManage Fault Status Register

MMFAR
MemManage Fault Status Register **($D34_H$)** **Reset Value: $XXXX\ XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS															
rw															

Field	Bits	Type	Description
ADDRESS	31:0	rw	**Data Address for an MPU Fault** This is the location addressed by an attempted load or store access that was faulted. The MemManage Status Register shows the cause of the fault, and whether MMFAR.ADDRESS is valid. When an unaligned access faults, the address is the actual address that faulted. Because an access might be split into multiple parts, each aligned, this address can be any offset in the range of the requested size.

Bus Fault Status Register

BFAR
Bus Fault Status Register **($D38_H$)** **Reset Value: $XXXX\ XXXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRESS															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS															
rw															

Field	Bits	Type	Description
ADDRESS	31:0	rw	**Data Address for a precise BusFault** This is the location addressed by an attempted data access that was faulted. The BFSR shows the reason for the fault, and whether BFAR.ADDRESS is valid. For unaligned access faults, the address returned is the address requested by the instruction. This might not be the address that faulted.

Auxiliary Fault Status Register

AFSR
Auxiliary Fault Status Register **($D3C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AUXFAULT															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXFAULT															
r															

Field	Bits	Type	Description
AUXFAULT	31:0	r	**not used, always read as 0**

8.5 Instruction Set Summary

This chapter provides the Instruction set. **Table 46** shows the instructions and their cycle counts. The cycle counts are based on a system with zero wait states.

Within the assembler syntax, depending on the operation, the <op2> field can be replaced with one of the following options:

- a simple register
- an immediate shifted register
- a register shifted register
- an immediate value

For brevity, not all load and store addressing modes are shown.

Table 46 uses the following abbreviations in the cycles column:

- P for the number of cycles required for a pipeline refill.
- B for the number of cycles required to perform the barrier operation.
- N for the number of registers in the register list to be loaded or stored, including PC or LR.
- W for the number of cycles spent waiting for an appropriate event.

Table 46 Instruction Set Summary

Operation	Description	Mnemonic	Cycles (without wait states)
Move	Register	MOV Rd, <op2>	1
	16-bit immediate	MOVW Rd, #<imm>	1
	Immediate into top	MOVT Rd, #<imm>	1
	To PC	MOV PC, Rm	1 + P
Add	Add	ADD Rd, Rn, <op2>	1
	Add to PC	Add PC, PC, Rm	1 + P
	Add with carry	ADC Rd, Rn, <op2>	1
	Form address	ADR Rd, <label>	1
Subtract	Subtract	SUB Rd, Rn, <op2>	1
	Subtract with borrow	SBC Rd, Rn, <op2>	1
	Reverse	RSB Rd, Rn, <op2>	1
Multiply	Multiply	MUL Rd, Rn, Rm	1
	Multiply accumulate	MLA Rd, Rn, Rm	2
	Multiply subtract	MLS Rd, Rn, Rm	2
	Long signed	SMULL RdLo, RDHi, Rn, Rm	3 to 5[1)]
	Long unsigned	UMULL RdLo, RdHi, Rn, Rm	3 to 5[1)]
	Long singed accumulate	SMLAL RdLo, RdHi, Rn, Rm	4 to 7[1)]
	Long unsigned accumulate	UMLAL RdLo, RdHi, Rn, Rm	4 to 7[1)]
Divide	Signed	SDIV Rd, Rn, Rm	2 to 12[2)]
	Unsigned	UDIV Rd, Rn, Rm	2 to 12[2)]
Saturate	Signed	SSAT Rd, #i<imm>, <op2>	1
	Unsigned	USAT Rd, #<imm>, <op2>	1

Table 46 Instruction Set Summary (cont'd)

Operation	Description	Mnemonic	Cycles (without wait states)
Compare	Compare	CMP Rn, <op2>	1
	Negative	CMN Rn, <op2>	1
Logical	AND	AND Rd, Rn, <op2>	1
	Exclusive OR	EOR Rd, Rn, <op2>	1
	OR	ORR Rd, Rn, <op2>	1
	OR NOT	ORN Rd, Rn, <op2>	1
	Bit clear	BIC Rd, Rn, <op2>	1
	Move NOT	MVN Rd, <op2>	1
	AND test	TST Rn, <op2>	1
	Exclusive OR test	TEQ Rn, <op1>	
Shift	Logical shift left	LSL Rd, Rn, #<imm>	1
	Logical shift left	LSL Rd, Rn, Rs	1
	Logical shift right	LSR Rd, Rn, #<imm>	1
	Logical shift right	LSR Rd, Rn, Rs	1
	Arithmetic shift right	ASR Rd, Rn, #<imm>	1
	Arithmetic shift right	ASR Rd, Rn, Rs	1
Rotate	Rotate right	ROR Rd, Rn, #<imm>	1
	Rotate right	ROR Rd, Rn, Rs	1
	With extension	RRX Rd, Rn	1
Count	Leading zeroes	CLZ Rd, Rn	1
Load	Word	LDR Rd, [Rn, <op2>]	2[3)]
	To PC	LDR PC, [Rn, <op2>]	2[3)] + P
	Halfword	LDRH Rd, [Rn, <op2>]	2[3)]
	Byte	LDRB Rd, [Rn, <op2>]	2[3)]
	Signed halfword	LDRSH Rd, [Rn, <op2>]	2[3)]
	Signed byte	LDRSB Rd, [Rn, <op2>]	2[3)]
	User word	LDRT Rd, [Rn, #<imm>]	2[3)]
	User halfword	LDRHT Rd, [Rn, #<imm>]	2[3)]
	User byte	LDRBT Rd, [Rn, #<imm>]	2[3)]
	User signed halfword	LDRSHT Rd, [Rn, #<imm>]	2[3)]
	User signed byte	LDRSBT Rd, [Rn, #<imm>]	2[3)]
	PC relative	LDR Rd, [PC, #<imm>]	2[3)]
	Doubleword	LDRD Rd, Rd, [Rn, #<imm>]	1 + N
	Multiple	LDM Rn, {<reglist>}	1 + N
	Multiple including PC	LDM Rn, {<reglist>, PC}	1 + N + P

Table 46 Instruction Set Summary (cont'd)

Operation	Description	Mnemonic	Cycles (without wait states)
Store	Word	STR Rd, [Rn, <op2>]	2[3)]
	Halfword	STRH Rd, [Rn, <op2>]	2[3)]
	Byte	STRB Rd, [Rn, <op2>]	2[3)]
	Signed halfword	STRSH Rd, [Rn, <op2>]	2[3)]
	Signed byte	STRSB Rd, [Rn, <op2>]	2[3)]
	User word	STRT Rd, [Rn, #<imm>]	2[3)]
	User halfword	STRHT Rd, [Rn, #<imm>]	2[3)]
	User byte	STRBT Rd, [Rn, #<imm>]	2[3)]
	User signed halfword	STRSHT Rd, [Rn, #<imm>]	2[3)]
	User signed byte	STRSBT Rd, [Rn, #<imm>]	2[3)]
	Doubleword	STRD RD, Rd, [Rn, #<imm>]	1 + N
	Multiple	STM Rn, {<reglist>}	1 + N
Push	Push	PUSH {<reglist>}	1 + N
	Push with link register	PUSH {<reglist>, LR}	1 + N
Pop	Pop	POP {<reglist>}	1 + N
	Pop and return	POP {<reglist>, PC}	1 + N + P
Semaphore	Load exclusive	LDREX Rd, [Rn, #imm>]	2
	Load exclusive half	LDREXH Rd, [Rn]	2
	Load exclusive byte	LDREXB Rd, [Rn]	2
	Store exclusive	STREX Rd, Rt, [Rn, #<imm>]	2
	Store exclusive half	STREXH Rd, Rt, [Rn]	2
	Store exclusive byte	STREXB Rd, Rt, [Rn]	2
	Clear exclusive monitor	CLREX	1
Branch	Conditional	B <cc> <label>	1 or 1 + P[4)]
	Unconditional	B <label>	1 + P
	With link	BL <label>	1 + P
	With exchange	BX Rm	1 + P
	With link and exchange	BLX Rm	1 + P
	Branch if zero	CBZ Rn, <label>	1 or 1 + P[4)]
	Branch if non-zero	CBZN Rn, <label>	1 or 1 + P[4)]
	Byte table branch	TBB [Rn, Rm]	2 + P
	Halfword table branch	TBH [Rn, Rm, LSL#1]	2 + P

Table 46 Instruction Set Summary (cont'd)

Operation	Description	Mnemonic	Cycles (without wait states)
State change	Supervisor call	SVC #<imm>	–
	If-then-else	IT... <cond>	1[5)]
	Disable interrupts	CPSID <flags>	1 or 2
	Enable interrupts	CPSIE <flags>	1 or 2
	Read special register	MRS Rd, <specreg>	1 or 2
	Write special register	MSR <specreg>, Rn	1 or 2
	Breakpoint	BKPT #<imm>	–
Extend	Signed halfword to word	SXTH Rd, <op2>	1
	Signed byte to word	SXTB Rd, <op2>	1
	Unsigned halfword	UXTH Rd, <op2>	1
	Unsigned byte	UXTB Rd, <op2>	1
Bit field	Extract unsigned	UBFX Rd, Rn, #<imm>, #<imm>	1
	Extract signed	SBFX Rd, Rn, #<imm>, #<imm>	1
	Clear	BFC Rd, Rn, #<imm>, #<imm>	1
	Insert	BFI Rd, Rn, #<imm>, #<imm>	1
Reverse	Bytes in word	REV Rd, Rm	1
	Bytes in both halfwords	REV16 Rd, Rm	1
	Signed bottom halfword	REVSH Rd, Rm	1
	Bits in word	RBIT Rd, Rm	1
Hint	Send event	SEV	1
	Wait for event	WFE	1 + W
	Wait for interrupt	WFI	1 + W
	No operation	NOP	1
Barriers	Instruction synchronization	ISB	1 + B
	Data memory	DMB	1 + B
	Data synchronization	DSB <flags>	1 + B

1) UMULL, SMULL, UMLAL, and SMLAL instructions use early termination depending on the size of the source values. These are interruptible, that is abandoned and restarted, with worst case latency of one cycle.
2) Division operations use early termination to minimize the number of cycles required based on the number of leading ones and zeros in the input operands.
3) Neighboring load and store single instructions can pipeline their address and data phases. This enables these instructions to complete in a single execution cycle.
4) Conditional branch completes in a single cycle if the branch is not taken.
5) An IT instruction can be folded onto a preceding 16-bit Thumb instruction, enabling execution in zero cycles.

9 DMA controller

Figure 40 shows the top level block diagram of the TLE987x.

The bus matrix allows the µDMA to access the PBA0, PBA1, and RAM.

9.1 Features

The principal features of the DMA Controller are:

- It is compatible with AHB-Lite for DMA transfers.
- It is compatible with APB for register programming.
- It has a single AHB-Lite master for transferring data using a 32-bit address bus and a 32-bit data bus.
- It supports up to 14 DMA channels.
- Each DMA channel has dedicated handshake signals.
- Each DMA channel has a programmable priority level.
- Each priority level arbitrates using a fixed priority that is determined by the DMA channel number. The DMA also supports multiple transfer types:
  - Memory-to-memory
  - Memory-to-peripheral
  - Peripheral-to-memory
- It supports multiple DMA cycle types.
- It supports multiple DMA transfer data widths.
- Each DMA channel can access a primary and an alternate channel control data structure.
- All the channel control data is stored in system memory (RAM) in little-endian format.
- It performs all DMA transfers using the single AHB-Lite burst type. The destination data width is equal to the source data width.
- The number of transfers in a single DMA cycle can be programmed from 1 to 1024.
- The transfer address increment can be greater than the data width.

9.2 Introduction

Please also refer to **Chapter 9.3**, **Functional description**.

9.2.1 Block diagram

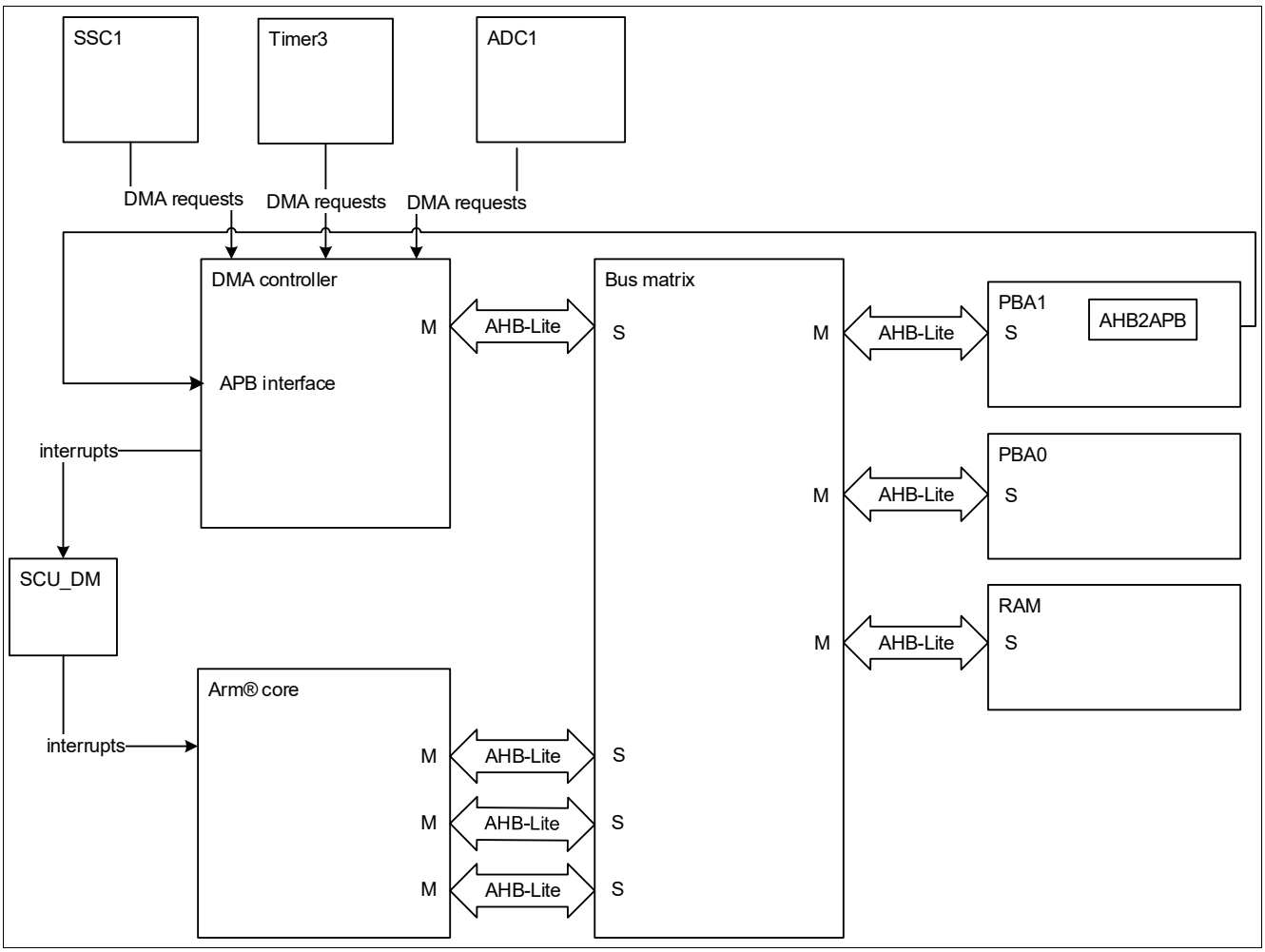


Figure 40 DMA controller top level block diagram

9.3 Functional description

9.3.1 DMA mode overview

The DMA controller implements the following up to 14 hardware DMA requests, see **Figure 41**:

- ADC1 complete sequence 1 done: DMA transfer is requested on completion of the ADC1 channel conversion sequence.
- ADC1 exceptional sequence 2 (ESM) done: DMA transfer is requested on completion of the ADC1 conversion sequence triggered by an exceptional measurement request.
- SSC1/2 transmit byte: DMA transfer is requested upon the completion of data transmission via SSC1/2.
- SSC1/2: receive byte: DMA transfer is requested upon the completion of data reception via SSC1/2.
- ADC1 channel 0 conversion done: DMA transfer is requested on completion of the ADC1 channel 0 conversion.
- ADC1 channel 1 conversion done: DMA transfer is requested on completion of the ADC1 channel 1 conversion.
- ADC1 channel 2 conversion done: DMA transfer is requested on completion of the ADC1 channel 2 conversion.
- ADC1 channel 3 conversion done: DMA transfer is requested on completion of the ADC1 channel 3 conversion.
- ADC1 channel 4 conversion done: DMA transfer is requested on completion of the ADC1 channel 4 conversion.
- ADC1 channel 5 conversion done: DMA transfer is requested on completion of the ADC1 channel 5 conversion.
- ADC1 channel 6 conversion done: DMA transfer is requested on completion of the ADC1 channel 6 conversion.
- ADC1 channel 7 conversion done: DMA transfer is requested on completion of the ADC1 channel 7 conversion.
- Timer3 ccu6_int: DMA transfer is requested following a timer trigger.
- SDADC, conversion done: DMA transfer is requested on completion of the SDADC (ADC3/4) conversion.[1)]

Note: Channel 0 has the highest priority, when the priority setting for all channels is the same.

1) Available only for products with SDADC. See the datasheet for details.

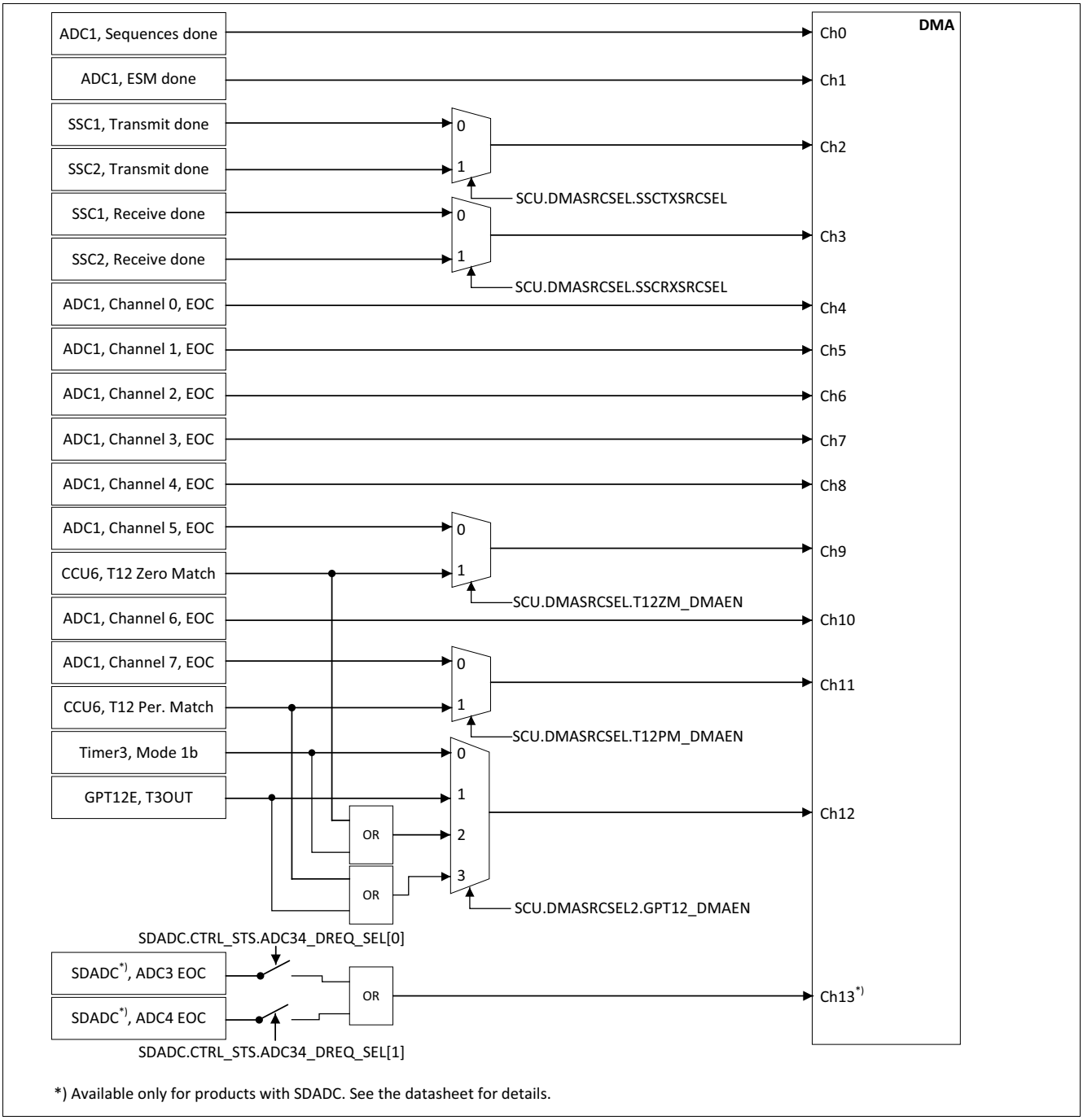


Figure 41 DMA transfer request sources

Burst transfers via signal transfer

Single transfers:

The DMA Controller should be programmed for single transfers, see **Table 60 "channel_cfg bit assignments" on Page 294**,

where R = 0, n = number of single transfer +1, transfer_type = 1 (basic).

For every dma_req, one transfer is done; if the number of transfers reaches n, dma_done (interrupt) is generated and the DMA channel is deactivated.

Burst transfers:

The DMA Controller should be programmed for burst transfers like this

n = number of transfers, transfer_type = 2 (auto)

Then for one dma_req, n +1 transfers are done, dma_done (interrupt) is generated and the DMA channel is deactivated.

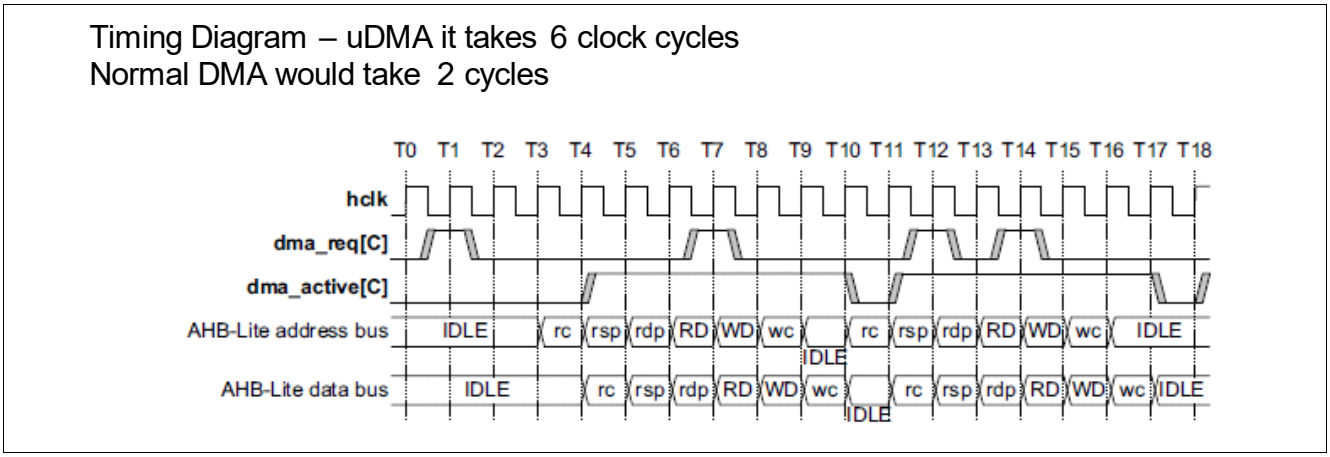


Figure 42 DMA Controller Timing Diagram

Table 47 DMA Controller Timing Diagram - Names and Abbreviations

Phrase	Definition
rc	read channel config
rsp	read src. ptr.
rdp	read dest. ptr
RD	read data
WD	write data
wc	write channel config

9.3.2 DMA Arbitration Rate

You can configure when the controller arbitrates during a DMA transfer. This enables you to reduce the latency to service a higher priority channel.

The controller provides four bits that configure how many AHB bus transfers occur before it rearbitrates. These bits are known as the R_power bits because the value you enter, R, is raised to the power of two and this determines the arbitration rate. For example, if R = 4 then the arbitration rate is 2^4, that is, the controller arbitrates every 16 DMA transfers.

Table 48 lists the arbitration rates.

Table 48 AHB-Bus Transfer Arbitration Interval

R_power	Arbitrate after x DMA Transfers
0000_B	x = 1
0001_B	x = 2
0010_B	x = 4

Table 48 AHB-Bus Transfer Arbitration Interval (cont'd)

R_power	Arbitrate after x DMA Transfers
0011_B	x = 8
0100_B	x = 16
0101_B	x = 32
0110_B	x = 64
0111_B	x = 128
1000_B	x = 256
1001_B	x = 512
$1010\text{-}1111_B$	x = 1024

Note: Take care not to assign a low-priority channel with a large R_power because this prevents the controller from servicing high-priority requests, until it rearbitrates.

When $N > 2^R$ and is not an integer multiple of 2^R then the controller always performs sequences of 2^R transfers until $N < 2^R$ remain to be transferred. The controller performs the remaining N transfers at the end of the DMA cycle.

You store the value of the R_power bits in the channel control data structure. See **Control Data Configuration** on **Page 293** for more information about the location of the R_power bits in the data structure.

Priority

When the controller arbitrates, it determines the next channel to service by using the following information:

- The channel number
- The priority level, default or high, that is assigned to the channel.

You can configure each channel to use either the default priority level or a high priority level by setting the chnl_priority_set Register. See Channel Priority set on page 3-23.

Channel number zero has the highest priority and as the channel number increases, the priority of a channel decreases. **Table 49** lists the DMA channel priority levels in descending order of priority.

Table 49 DMA Channel Priority

Channel Number	Priority Level Setting	Descending Order of Channel Priority
0	High	Highest-priority DMA channel
1	High	–
2	High	–
–	High	–
–	High	–
–	High	–
12	High	–
13	High	–
0	Default	–
1	Default	–
2	Default	–
–	Default	–

Table 49 DMA Channel Priority (cont'd)

Channel Number	Priority Level Setting	Descending Order of Channel Priority
–	Default	–
–	Default	–
12	Default	–
13	Default	Lowest-priority DMA channel

After a DMA transfer completes, the controller polls all the DMA channels that are available. **Figure 43 "Polling Flowchart" on Page 279** shows the process it uses to determine which DMA transfer to perform next.

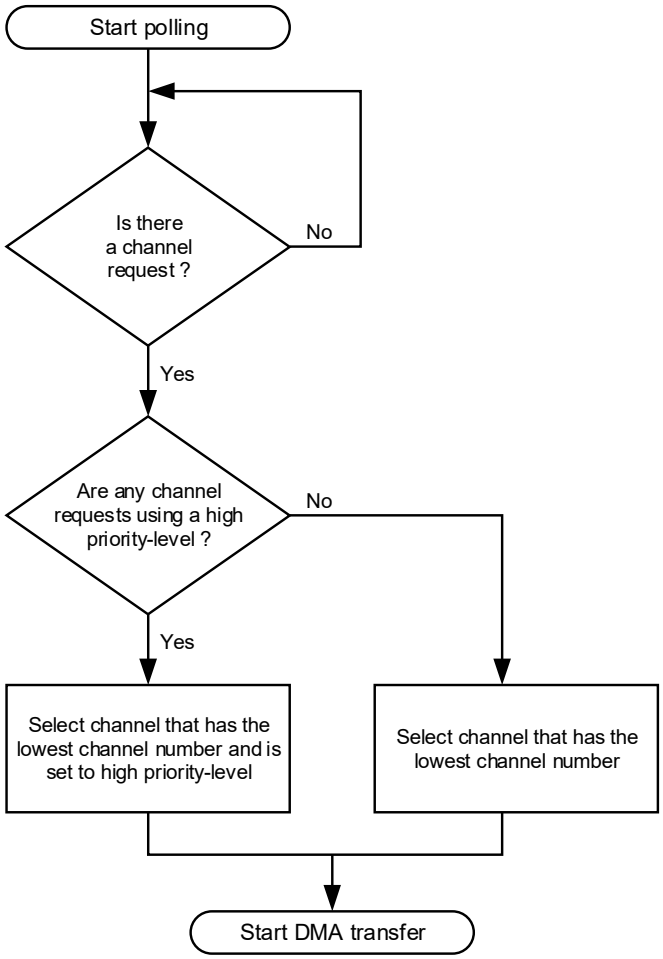


Figure 43 Polling Flowchart

Note: When dma_req[C] is HIGH, the controller sets dma_done[C] HIGH. This enables the controller to alert the host processor to a request, even when the channel is disabled.

DMA Cycle Types

The cycle_ctrl bits control how the controller performs a DMA cycle. You can set the cycle_ctrl bits as **Table 50** lists.

Table 50 DMA Cycle Types

cycle_ctrl	Description
000_B	Channel control data structure is invalid
001_B	Basic DMA transfer
010_B	Auto-request
011_B	Ping-pong
100_B	Memory scatter-gather using the primary data structure
101_B	Memory scatter-gather using the alternate data structure
110	Peripheral scatter-gather using the primary data structure
111	Peripheral scatter-gather using the alternate data structure

Note: The cycle_ctrl bits are located in the channel_cfg memory location that ***Control Data Configuration*** *on* ***Page 293*** *describes.*

For all cycle types, the controller arbitrates after 2^R DMA transfers. If you set a low-priority channel with a large 2^R value then it prevents all other channels from performing a DMA transfer, until the low-priority DMA transfer completes. Therefore, you must take care when setting the R_power, that you do not significantly increase the latency for high-priority channels.

The following sections describe the cycle types:

- **Invalid**
- **Basic**
- **Auto-Request** on **Page 281**
- **Ping-Pong** on **Page 281**
- **Memory Scatter-Gather** on **Page 284**
- **Peripheral Scatter-Gather** on **Page 287**

Invalid

After the controller completes a DMA cycle it sets the cycle type to invalid, to prevent it from repeating the same DMA cycle.

Basic

In this mode, you configure the controller to use either the primary, or alternate, data structure. After you enable the channel, and the controller receives a request then the flow for this DMA cycle is:

- The controller performs 2^R transfers. If the number of transfers remaining is zero the flow continues at step 3.
- The controller arbitrates:
  - If a higher-priority channel is requesting service then the controller services that channel.
  - If the peripheral or software signals a request to the controller then it continues at step 1.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle. This indicates to the host processor that the DMA cycle is complete.

Auto-Request

When the controller operates in this mode, it is only necessary for it to receive a single request to enable it to complete the entire DMA cycle. This enables a large data transfer to occur, without significantly increasing the latency for servicing higher priority requests, or requiring multiple requests from the processor or peripheral.

You can configure the controller to use the primary, or alternate, data structure. After you enable the channel, and the controller receives a request for this channel, then the flow for this DMA cycle is:

- The controller performs 2^R transfers for channel C. If the number of transfers remaining is zero the flow continues at step 3.
- The controller arbitrates. When channel C has the highest priority then the DMA cycle continues at step 1.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle. This indicates to the host processor that the DMA cycle is complete.

Ping-Pong

In ping-pong mode, the controller performs a DMA cycle using one of the data structures and it then performs a DMA cycle using the other data structure. The controller continues to switch from primary to alternate to primary… until it reads a data structure that is invalid, or until the host processor disables the channel.

Figure 44 “Ping-Pong Example” on Page 282 shows an example of a ping-pong DMA transaction.

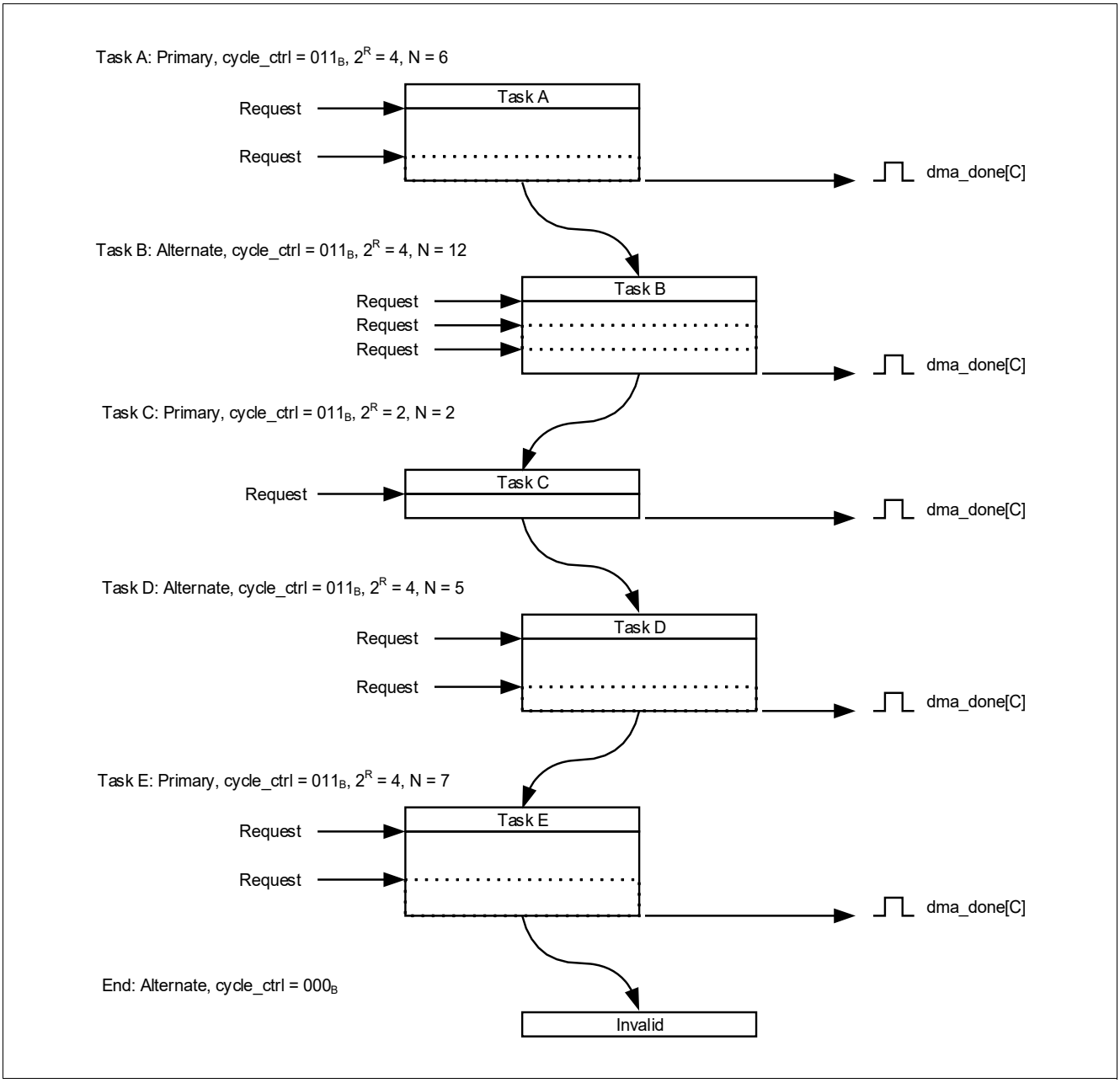


Figure 44 Ping-Pong Example

In **Figure 44 "Ping-Pong Example" on Page 282**:

Task A

- The host processor configures the primary data structure for task A.
- The host processor configures the alternate data structure for task B. This enables the controller to immediately switch to task B after task A completes, provided that a higher priority channel does not require servicing.
- The controller receives a request and performs four DMA transfers.
- The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- The controller performs the remaining two DMA transfers.

- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

After task A completes, the host processor can configure the primary data structure for task C. This enables the controller to immediately switch to task C after task B completes, provided that a higher priority channel does not require servicing.

After the controller receives a new request for the channel and it has the highest priority then task B commences:

Task B

- The controller performs four DMA transfers.
- The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- The controller performs four DMA transfers.
- The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- The controller performs the remaining four DMA transfers.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

After task B completes, the host processor can configure the alternate data structure for task D.

After the controller receives a new request for the channel and it has the highest priority then task C commences:

Task C

- The controller performs two DMA transfers.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

After task C completes, the host processor can configure the primary data structure for task E.After the controller receives a new request for the channel and it has the highest priority then task D commences:

Task D

- The controller performs four DMA transfers.
- The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- The controller performs the remaining DMA transfer.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

After the controller receives a new request for the channel and it has the highest priority then task E commences:

Task E

- The controller performs four DMA transfers.
- The controller arbitrates. After the controller receives a request for this channel, the flow continues if the channel has the highest priority.
- The controller performs the remaining three DMA transfers.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

If the controller receives a new request for the channel and it has the highest priority then it attempts to start the next task. However, because the host processor has not configured the alternate data structure, and on completion of task D the controller set the cycle_ctrl bits to 000_B, then the ping-pong DMA transaction completes.

Note: You can also terminate the ping-pong DMA cycle in ***Figure 44 "Ping-Pong Example" on Page 282****, if you configure task E to be a basic DMA cycle by setting the cycle_ctrl field to 3'001$_B$.*

Memory Scatter-Gather

In memory scatter-gather mode the controller receives an initial request and then performs four DMA transfers using the primary data structure. After this transfer completes, it starts a DMA cycle using the alternate data structure. After this cycle completes, the controller performs another four DMA transfers using the primary data structure. The controller continues to switch from primary to alternate to primary… until either:

- The host processor configures the alternate data structure for a basic cycle.
- It reads an invalid data structure.

Note: After the controller completes the N primary transfers it invalidates the primary data structure by setting the cycle_ctrl field to 000$_B$.

The controller only asserts **dma_done[C]** when the scatter-gather transaction completes using a basic cycle.

In scatter-gather mode, the controller uses the primary data structure to program the alternate data structure. **Table 51** lists the fields of the channel_cfg memory location for the primary data structure, that you must program with constant values and those that can be user defined.

Table 51 channel_cfg for a Primary Data Structure, in Memory Scatter-Gather Mode

Bit	Field	Value	Description
Constant-Value Fields:			
[31:30]	dst_inc	10_B	Configures the controller to use word increments for the address
[29:28]	dst_size	10_B	Configures the controller to use word transfers
[27:26]	src_inc	10_B	Configures the controller to use word increments for the address
[25:24]	src_size	10_B	Configures the controller to use word transfers
[17:14]	R_power	0010_B	Configures the controller to perform four DMA transfers
[3]	next_useburst	0_B	For a memory scatter-gather DMA cycle, this bit must be set to zero
[2:0]	cycle_ctrl	100_B	Configures the controller to perform a memory scatter-gather DMA cycle
User Defined Values:			
[23:21]	dst_prot_ctrl	–	Configures the state of **HPROT** when the controller writes the destination data
[20:18]	src_prot_ctrl	–	Configures the state of **HPROT** when the controller reads the source data
[13:4]	n_minus_1	N[1)]	Configures the controller to perform N DMA transfers, where N is a multiple of four

1) Because the R_power field is set to four, you must set N to be a multiple of four. The value given by N/4 is the number of times that you must configure the alternate data structure.

See **Control Data Configuration** on **Page 293** for more information.

Figure 45 "Memory Scatter-Gather Example" on Page 285 shows a memory scatter-gather example.

Initialization

- Configure primary to enable the copy A, B, C and D operations: cycle_ctrl = 100_B, $2^R = 4$, N = 16
- Write the primary source data to memory, using the structure shown in the following table.

Table 52

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	$00A000000_H$	$00AE00000_H$	cycle_ctrl = 101_B, $2^R = 4$, N = 3	$0XXXXXXXX_H$
Data for Task B	$00B000000_H$	$00BE00000_H$	cycle_ctrl = 101_B, $2^R = 2$, N = 8	$0XXXXXXXX_H$
Data for Task C	$00C000000_H$	$00CE00000_H$	cycle_ctrl = 101_B, $2^R = 8$, N = 5	$0XXXXXXXX_H$
Data for Task D	$00D000000_H$	$00DE00000_H$	cycle_ctrl = 001_B, $2^R = 4$, N = 4	$0XXXXXXXX_H$

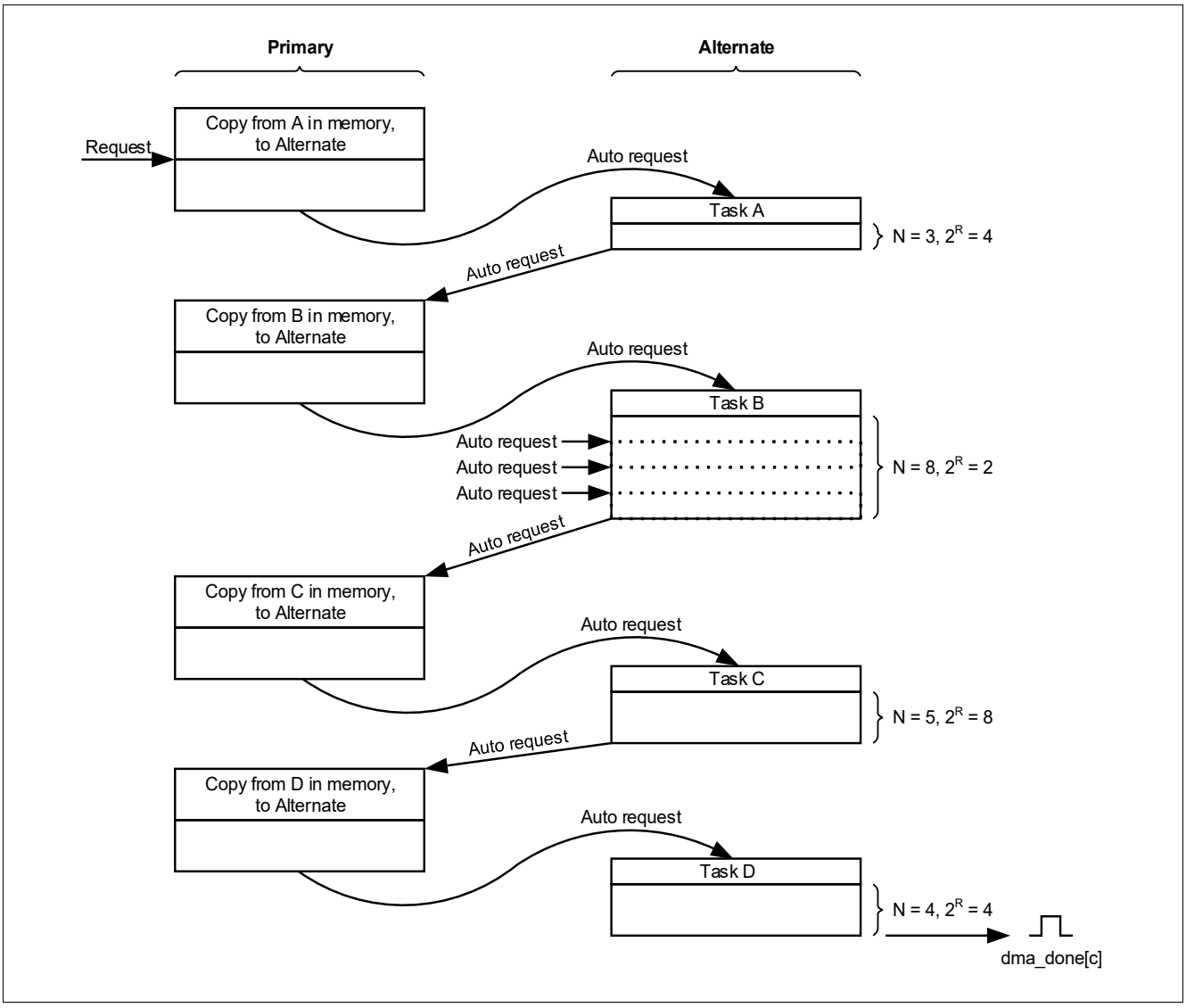


Figure 45 Memory Scatter-Gather Example

In **Figure 45 "Memory Scatter-Gather Example" on Page 285**:

Initialization

- The host processor configures the primary data structure to operate in memory scatter-gather mode by setting cycle_ctrl to 100_B. Because a data structure for a single channel consists of four words then you must set 2^R to 4. In this example, there are four tasks and therefore N is set to 16.
- The host processor writes the data structure for tasks A, B, C, and D to the memory locations that the primary src_data_end_ptr specifies.
- The host processor enables the channel.

The memory scatter-gather transaction commences when the controller receives a request on **dma_req[]** or a manual request from the host processor. The transaction continues as follows:

Primary, copy A

- After receiving a request, the controller performs four DMA transfers. These transfers write the alternate data structure for task A.
- The controller generates an auto-request for the channel and then arbitrates.

Task A

- The controller performs task A. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy B

- The controller performs four DMA transfers. These transfers write the alternate data structure for task B.
- The controller generates an auto-request for the channel and then arbitrates.

Task B

- The controller performs task B. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy C

- The controller performs four DMA transfers. These transfers write the alternate data structure for task C.
- The controller generates an auto-request for the channel and then arbitrates.

Task C

- The controller performs task C. After it completes the task, it generates an auto-request for the channel and then arbitrates.

Primary, copy D

- The controller performs four DMA transfers. These transfers write the alternate data structure for task D.
- The controller sets the cycle_ctrl bits of the primary data structure to 000_B, to indicate that this data structure is now invalid.
- The controller generates an auto-request for the channel and then arbitrates.

Task D

- The controller performs task D using a basic cycle.
- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

Peripheral Scatter-Gather

In peripheral scatter-gather mode the controller receives an initial request from a peripheral and then it performs four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure, without rearbitrating or **dma_active[C]** going LOW.

Note: These are the only circumstances, where the controller does not enter the arbitration process after completing a transfer using the primary data structure.

After this cycle completes, the controller rearbitrates and if the controller receives a request from the peripheral that has the highest priority then it performs another four DMA transfers using the primary data structure. It then immediately starts a DMA cycle using the alternate data structure, without re-arbitrating or **dma_active[C]** going LOW. The controller continues to switch from primary to alternate to primary… until either:

- The host processor configures the alternate data structure for a basic cycle.
- It reads an invalid data structure.

Note: After the controller completes the N primary transfers it invalidates the primary data structure by setting the cycle_ctrl field to 000_B.

The controller asserts **dma_done[C]** when the scatter-gather transaction completes using a basic cycle.

In scatter-gather mode, the controller uses the primary data structure to program the alternate data structure. **Table 53** lists the fields of the channel_cfg memory location for the primary data structure, that you must program with constant values and those that can be user defined.

Table 53 channel_cfg for a Primary Data Structure, in Peripheral Scatter-Gather Mode

Bit	Field	Value	Description
Constant-Value Fields:			
[31:30]	dst_inc	10_B	Configures the controller to use word increments for the address
[29:28]	dst_size	10_B	Configures the controller to use word transfers
[27:26]	src_inc	10_B	Configures the controller to use word increments for the address
[25:24]	src_size	10_B	Configures the controller to use word transfers
[17:14]	R_power	0010_B	Configures the controller to perform four DMA transfers
[2:0]	cycle_ctrl	110_B	Configures the controller to perform a peripheral scatter-gather DMA cycle
User Defined Values:			
[23:21]	dst_prot_ctrl	–	Configures the state of **HPROT** when the controller writes the destination data
[20:18]	src_prot_ctrl	–	Configures the state of **HPROT** when the controller reads the source data
[13:4]	n_minus_1	N[1)]	Configures the controller to perform N DMA transfers, where N is a multiple of four
[3]	next_useburst	–	When set to 1, the controller sets the chnl_useburst_set [C] bit to 1 after the alternate transfer completes

1) Because the R_power field is set to four, you must set N to be a multiple of four. The value given by N/4 is the number of times that you must configure the alternate data structure.

See **Control Data Configuration** on **Page 293** for more information.

Figure 46 "Peripheral Scatter-Gather Example" on Page 288 shows a peripheral scatter-gather example.

Initialization

- Configure primary to enable the copy A, B, C and D operations: cycle_ctrl = 110_B, $2^R = 4$, N = 16.
- Write the primary source data to memory, using the structure shown in the following table.

Table 54

	src_data_end_ptr	dst_data_end_ptr	channel_cfg	Unused
Data for Task A	$00A000000_H$	$00AE00000_H$	cycle_ctrl = 111_B, $2^R = 4$, N = 3	$0XXXXXXXX_H$
Data for Task B	$00B000000_H$	$00BE00000_H$	cycle_ctrl = 111_B, $2^R = 2$, N = 8	$0XXXXXXXX_H$
Data for Task C	$00C000000_H$	$00CE00000_H$	cycle_ctrl = 111_B, $2^R = 8$, N = 5	$0XXXXXXXX_H$
Data for Task D	$00D000000_H$	$00DE00000_H$	cycle_ctrl = 001_B, $2^R = 4$, N = 4	$0XXXXXXXX_H$

Peripheral scatter-gather transaction:

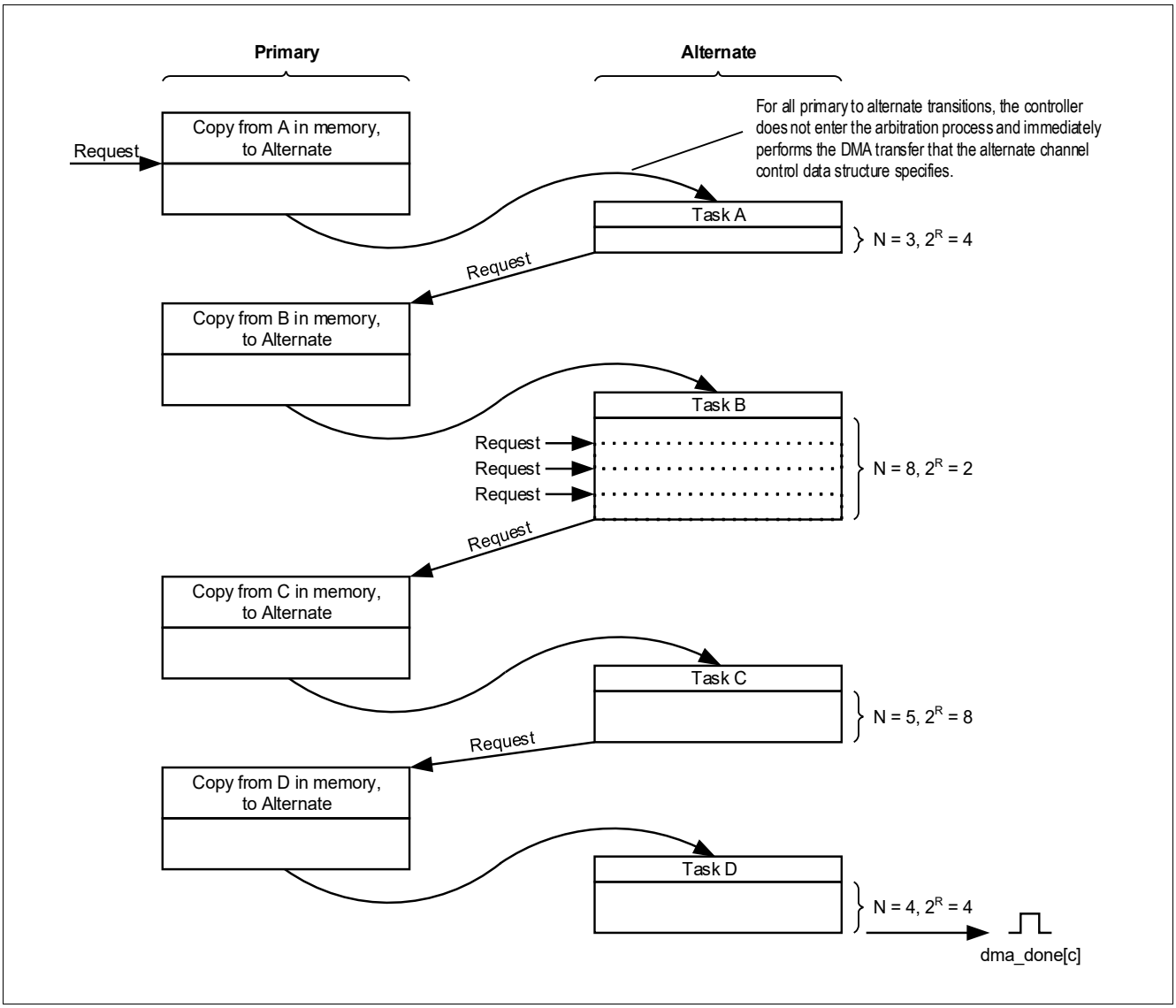


Figure 46 Peripheral Scatter-Gather Example

In Figure 46 “Peripheral Scatter-Gather Example” on Page 288:

Initialization

- The host processor configures the primary data structure to operate in peripheral scatter-gather mode by setting cycle_ctrl to 110_B. Because a data structure for a single channel consists of four words then you must set 2^R to 4. In this example, there are four tasks and therefore N is set to 16.
- The host processor writes the data structure for tasks A, B, C, and D to the memory locations that the primary src_data_end_ptr specifies.
- The host processor enables the channel.

The peripheral scatter-gather transaction commences when the controller receives a request on **dma_req[]**. The transaction continues as follows:

Primary, copy A

- After receiving a request, the controller performs four DMA transfers. These transfers write the alternate data structure for task A.

Task A

- The controller performs task A.
- After the controller completes the task it enters the arbitration process.

After the peripheral issues a new request and it has the highest priority then the process continues with:

Primary, copy B

- The controller performs four DMA transfers. These transfers write the alternate data structure for task B.

Task B

- The controller performs task B. To enable the controller to complete the task, the peripheral must issue a further three requests.
- After the controller completes the task it enters the arbitration process.

After the peripheral issues a new request and it has the highest priority then the process continues with:

Primary, copy C

- The controller performs four DMA transfers. These transfers write the alternate data structure for task C.

Task C

- The controller performs task C.
- After the controller completes the task it enters the arbitration process.

After the peripheral issues a new request and it has the highest priority then the process continues with:

Primary, copy D

- The controller performs four DMA transfers. These transfers write the alternate data structure for task D.
- The controller sets the cycle_ctrl bits of the primary data structure to 000_B, to indicate that this data structure is now invalid.

Task D

- The controller performs task D using a basic cycle.

- The controller sets **dma_done[C]** HIGH for one **hclk** cycle and enters the arbitration process.

Error Signaling

If the controller detects an ERROR response on the AHB-Lite master interface, it:

- Disables the channel that corresponds to the ERROR.
- Sets **dma_err** HIGH.

After the host processor detects that **dma_err** is HIGH, it must check which channel was active when the ERROR occurred. It can do this by:

- Reading the chnl_enable_set Register to create a list of disabled channels.

When a channel asserts **dma_done[]** then the controller disables the channel. The program running on the host processor must always keep a record of which channels have recently asserted their **dma_done[]** outputs.

- It must compare the disabled channels list from step 1, with the record of the channels that have recently set their **dma_done[]** outputs. The channel with no record of **dma_done[C]** being set is the channel that the ERROR occurred on.

9.3.3 Channel Control Data Structure

You must provide an area of system memory to contain the channel control data structure. This system memory must:

- Provide a contiguous area of system memory that the controller and host processor can access.
- Have a base address that is an integer multiple of the total size of the channel control data structure.

Figure 47 shows the memory that the controller requires for the channel control data structure, when it uses all 14 channels and the optional alternate data structure.

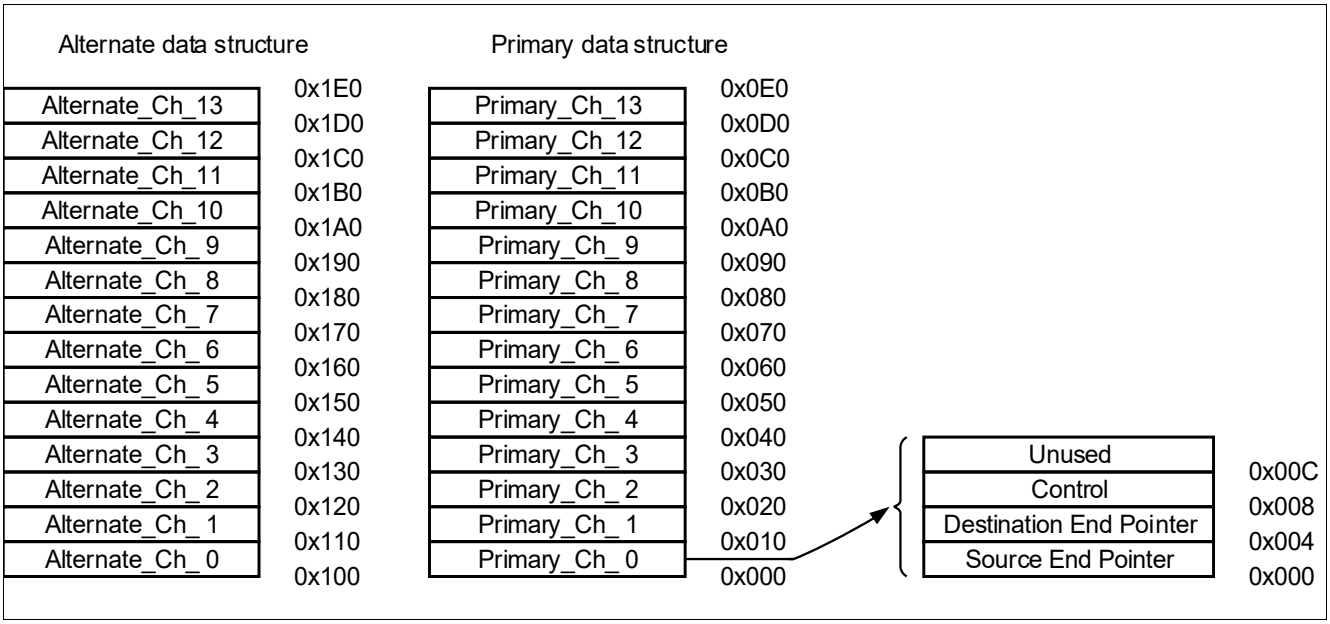


Figure 47 Memory Map for 32 Channels, Including the Alternate Data Structure

The example structure in **Figure 47** uses 1KB of system memory. In this example, the controller uses the lower 10 address bits to enable it to access all of the elements in the structure and therefore the base address must be at $0XXXXX000_H$, $0XXXXX400_H$, $0XXXXX800_H$, or $0XXXXXC00_H$. You can configure the base address for the

primary data structure by writing the appropriate value in the ctrl_base_ptr Register. See Channel control data base pointer on page 3-8.

The amount of system memory you require depends on:

- The number of DMA channels you configure the controller to use.
- If you configure a DMA channel to use the alternate data structure. See Channel primary-alternate set.

Table 55 lists the address bits that the controller uses when it accesses the elements of the channel control data structure, depending on the number of channels that the controller contains.

Table 55 Address Bit Settings for the Channel Control Data Structure

Address Bits							
Number of DMA Channels Implemented	**[9]**	**[8]**	**[7]**	**[6]**	**[5]**	**[4]**	**[3:0]**
0-13		A	C[3]	C[2]	C[1]	C[0]	00_H, 04_H or 08_H

Where:

Table 56

A	Selects one of the channel control data structures:	
	A = 0	Selects the primary data structure.
	A = 1	Selects the alternate data structure.
C[x:0]	Selects the DMA channel.	
Address[3:0]	Selects one of the control elements:	
	00_H	Selects the source data end pointer.
	04_H	Selects the destination data end pointer.
	08_H	Selects the control data configuration.
	$0C_H$	The controller does not access this address location. If required, you can enable the host processor to use this memory location as system memory.

Note: It is not necessary for you to calculate the base address of the alternate data structure because the alt_ctrl_base_ptr Register provides this information. See Channel alternate control data base pointer.

Figure 47 shows an example implementation where the controller uses three DMA channels and the alternate data structure.

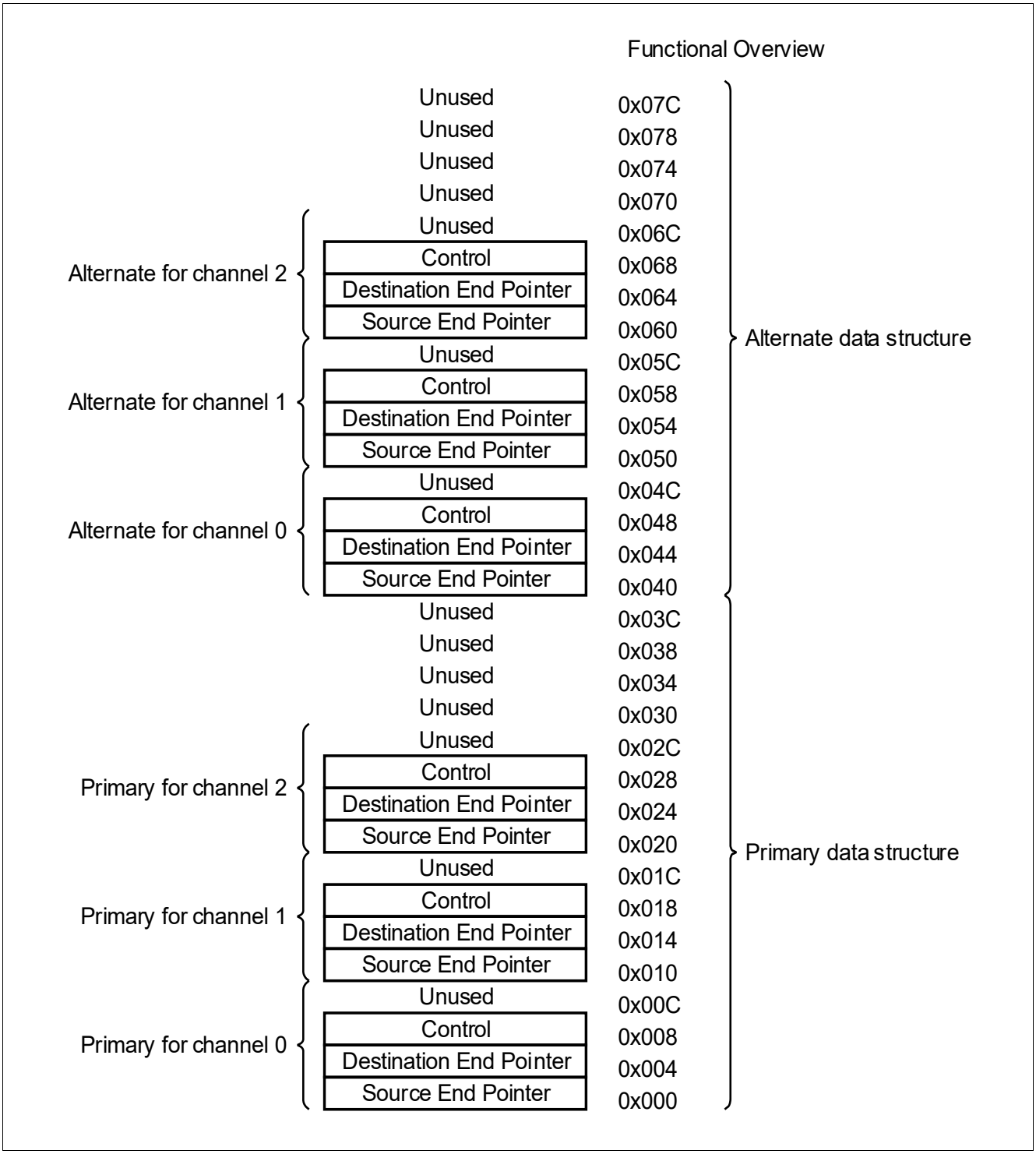


Figure 48 Memory Map for Three DMA Channels, Including the Alternate Data Structure

The example structure in **Figure 47** uses 128 bytes of system memory. In this example, the controller uses the lower six address bits to enable it to access all of the elements in the structure and therefore the base address must be at $0XXXXXX00_H$ or $0XXXXXX80_H$.

Table 57 lists the permitted base address values that you can assign for the primary data structure, depending on the number of channels that the controller contains.

Table 57 Permitted Base Addresses

Number of DMA Channels	Permitted Base Addresses[1)] for the Primary Data Structure
0-13	$0XXXXX000_H$, $0XXXXX200_H$, $0XXXXX400_H$, $0XXXXX600_H$, $0XXXXX800_H$, $0XXXXXA00_H$, $0XXXXXC00_H$, $0XXXXXE00_H$

1) Where X is a hexadecimal.

The controller uses the system memory to enable it to access two pointers and the control information that it requires for each channel. The following subsections describe these 32-bit memory locations and how the controller calculates the DMA transfer address:

- **Source Data End Pointer**
- **Destination Data End Pointer**
- **Control Data Configuration**
- **Address Calculation**

Source Data End Pointer

The src_data_end_ptr memory location contains a pointer to the end address of the source data. **Table 58** lists the bit assignments for this memory location.

Table 58 src_data_end_ptr bit assignments

Bit	Name	Description
[31:0]	src_data_end_ptr	Pointer to the end address of the source data

Before the controller can perform a DMA transfer, you must program this memory location with the end address of the source data. The controller reads this memory location when it starts a 2^R DMA transfer.

Note: The controller does not write to this memory location.

Destination Data End Pointer

The dst_data_end_ptr memory location contains a pointer to the end address of the destination data. **Table 59** lists the bit assignments for this memory location.

Table 59 dst_data_end_ptr bit assignments

Bit	Name	Description
[31:0]	dst_data_end_ptr	Pointer to the end address of the destination data

Before the controller can perform a DMA transfer, you must program this memory location with the end address of the destination data. The controller reads this memory location when it starts a 2^R DMA transfer.

Note: The controller does not write to this memory location.

Control Data Configuration

For each DMA transfer, the channel_cfg memory location provides the control information for the controller. **Figure 49** shows the bit assignments for this memory location.

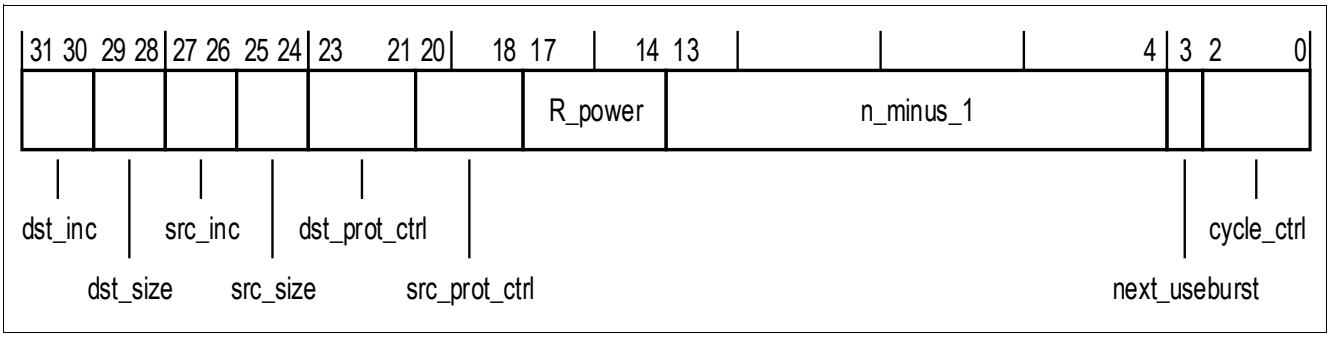


Figure 49 channel_cfg bit assignments

Table 60 lists the bit assignments for this memory location.

Table 60 channel_cfg bit assignments

Bit	Name	Description
[31:30]	dst_inc	Destination address increment. The address increment depends on the source data width as follows: **Source data width = byte** 00_B = byte. 01_B = halfword. 10_B = word. 11_B = no increment. Address remains set to the value that the dst_data_end_ptr memory location contains. **Source data width = halfword** 00_B = reserved. 01_B = halfword. 10_B = word. 11_B = no increment. Address remains set to the value that the dst_data_end_ptr memory location contains. **Source data width = word** 00_B = reserved. 01_B = reserved. 10_B = word. 11_B = no increment. Address remains set to the value that the dst_data_end_ptr memory location contains.
[29:28]	dst_size	Destination data size. *Note: You must set dst_size to contain the same value that src_size contains.*

Table 60 channel_cfg bit assignments (cont'd)

Bit	Name	Description
[27:26]	src_inc	Set the bits to control the source address increment. The address increment depends on the source data width as follows: **Source data width = byte** 00_B = byte. 01_B = halfword. 10_B = word. 11_B = no increment. Address remains set to the value that the src_data_end_ptr memory location contains. **Source data width = halfword** 00_B = reserved. 01_B = halfword. 10_B = word. 11_B = no increment. Address remains set to the value that the src_data_end_ptr memory location contains. **Source data width = word** 00_B = reserved. 01_B = reserved. 10_B = word. 11_B = no increment. Address remains set to the value that the src_data_end_ptr memory location contains.
[25:24]	src_size	Set the bits to match the size of the source data: 00_B = byte 01_B = halfword 10_B = word 11_B = reserved.
[23:21]	dst_prot_ctrl	Set the bits to control the state of **HPROT[3:1]** when the controller writes the destination data. **Bit [23]** Controls the state of **HPROT[3]** as follows: 0_B = **HPROT[3]** is LOW and the access is non-cacheable. 1_B = **HPROT[3]** is HIGH and the access is cacheable. **Bit [22]** Controls the state of **HPROT[2]** as follows: 0_B = **HPROT[2]** is LOW and the access is non-bufferable. 1_B = **HPROT[2]** is HIGH and the access is bufferable. **Bit [21]** Controls the state of **HPROT[1]** as follows: 0_B = **HPROT[1]** is LOW and the access is non-privileged. 1_B = **HPROT[1]** is HIGH and the access is privileged.

Table 60 channel_cfg bit assignments (cont'd)

Bit	Name	Description
[20:18]	src_prot_ctrl	Set the bits to control the state of **HPROT[3:1]** when the controller reads the source data. **Bit [20]** Controls the state of **HPROT[3]** as follows: 0_B = **HPROT[3]** is LOW and the access is non-cacheable. 1_B = **HPROT[3]** is HIGH and the access is cacheable. **Bit [19]** Controls the state of **HPROT[2]** as follows:0 = **HPROT[2]** is LOW and the access is non-bufferable.1 = **HPROT[2]** is HIGH and the access is bufferable. **Bit [18]** Controls the state of **HPROT[1]** as follows: 0_B = **HPROT[1]** is LOW and the access is non-privileged. 1_B = **HPROT[1]** is HIGH and the access is privileged.
[17:14]	R_power	Set these bits to control how many DMA transfers can occur before the controller rearbitrates. The possible arbitration rate settings are: **0000_B** Arbitrates after each DMA transfer. **0001_B** Arbitrates after 2 DMA transfers. **0010_B** Arbitrates after 4 DMA transfers. **0011_B** Arbitrates after 8 DMA transfers. **0100_B** Arbitrates after 16 DMA transfers. **0101_B** Arbitrates after 32 DMA transfers. **0110_B** Arbitrates after 64 DMA transfers. **0111_B** Arbitrates after 128 DMA transfers. **1000_B** Arbitrates after 256 DMA transfers. **1001_B** Arbitrates after 512 DMA transfers. **1010_B-1111_B** Arbitrates after 1024 DMA transfers. This means that no arbitration occurs during the DMA transfer because the maximum transfer size is 1024.
[13:4]	n_minus_1	Prior to the DMA cycle commencing, these bits represent the total number of DMA transfers that the DMA cycle contains. You must set these bits according to the size of DMA cycle that you require. The 10-bit value indicates the number of DMA transfers, minus one. The possible values are: 000000000_B = 1 DMA transfer 000000001_B = 2 DMA transfers 000000010_B = 3 DMA transfers 000000011_B = 4 DMA transfers 000000100_B = 5 DMA transfers.. . 111111111_B = 1024 DMA transfers. The controller updates this field immediately prior to it entering the arbitration process. This enables the controller to store the number of outstanding DMA transfers that are necessary to complete the DMA cycle.

Table 60 channel_cfg bit assignments (cont'd)

Bit	Name	Description
[3]	next_useburst	Controls if the chnl_useburst_set [C] bit is set to a 1, when the controller is performing a peripheral scatter-gather and is completing a DMA cycle that uses the alternate data structure. *Note: Immediately prior to completion of the DMA cycle that the alternate data structure specifies, the controller sets the chnl_useburst_set [C] bit to 0 if the number of remaining transfers is less than 2R. The setting of the next_useburst bit controls if the controller performs an additional modification of the chnl_useburst_set [C] bit.* In peripheral scatter-gather DMA cycle then after the DMA cycle that uses the alternate data structure completes, either:0 = The controller does not change the value of the chnl_useburst_set [C] bit. If the chnl_useburst_set [C] bit is 0 then for all the remaining DMA cycles in the peripheral scatter-gather transaction, the controller responds to requests on **dma_req[]** and **dma_sreq[]**, when it performs a DMA cycle that uses an alternate data structure.1 = The controller sets the chnl_useburst_set [C] bit to a 1. Therefore, for the remaining DMA cycles in the peripheral scatter-gather transaction, the controller only responds to requests on **dma_req[]**, when it performs a DMA cycle that uses an alternate data structure.

Table 60 channel_cfg bit assignments (cont'd)

Bit	Name	Description	
[2:0]	cycle_ctrl	The operating mode of the DMA cycle. The modes are:	
		000_B	Stop. Indicates that the data structure is invalid.
		001_B	Basic. The controller must receive a new request, prior to it entering the arbitration process, to enable the DMA cycle to complete.
		010_B	Auto-request. The controller automatically inserts a request for the appropriate channel during the arbitration process. This means that the initial request is sufficient to enable the DMA cycle to complete.
		011_B	Ping-pong. The controller performs a DMA cycle using one of the data structures. After the DMA cycle completes, it performs a DMA cycle using the other data structure. After the DMA cycle completes and provided that the host processor has updated the original data structure, it performs a DMA cycle using the original data structure. The controller continues to perform DMA cycles until it either reads an invalid data structure or the host processor changes the cycle_ctrl bits to 001_B or 010_B. See **Ping-Pong** on **Page 281**.
		100_B	Memory scatter/gather. See **Memory Scatter-Gather** on **Page 284**.When the controller operates in memory scatter-gather mode, you must only use this value in the primary data structure.
		101_B	Memory scatter/gather. See **Memory Scatter-Gather** on **Page 284**.When the controller operates in memory scatter-gather mode, you must only use this value in the alternate data structure.
		110_B	Peripheral scatter/gather. See **Peripheral Scatter-Gather** on **Page 287**. When the controller operates in peripheral scatter-gather mode, you must only use this value in the primary data structure.
		111_B	Peripheral scatter/gather. See **Peripheral Scatter-Gather** on **Page 287**. When the controller operates in peripheral scatter-gather mode, you must only use this value in the alternate data structure.

At the start of a DMA cycle, or 2^R DMA transfer, the controller fetches the channel_cfg from system memory. After it performs 2^R, or N, transfers it stores the updated channel_cfg in system memory.

The controller does not support a dst_size value that is different to the src_size value. If it detects a mismatch in these values, it uses the src_size value for source and destination and when it next updates the n_minus_1 field, it also sets the dst_size field to the same as the src_size field.

After the controller completes the N transfers it sets the cycle_ctrl field to 000_B, to indicate that the channel_cfg data is invalid. This prevents it from repeating the same DMA transfer.

Address Calculation

To calculate the source address of a DMA transfer, the controller performs a left shift operation on the n_minus_1 value by a shift amount that src_inc specifies, and then subtracts the resulting value from the source data end pointer. Similarly, to calculate the destination address of a DMA transfer, it performs a left

shift operation on the n_minus_1 value by a shift amount that dst_inc specifies, and then subtracts the resulting value from the destination end pointer.

Depending on the value of src_inc and dst_inc, the source address and destination address can be calculated using the equations:

src_inc = 00_B and dst_inc = 00_B

- source address = src_data_end_ptr - n_minus_1
- destination address = dst_data_end_ptr - n_minus_1

src_inc = 01_B and dst_inc = 01_B

- source address = src_data_end_ptr - (n_minus_1 << 1)
- destination address = dst_data_end_ptr - (n_minus_1 << 1)

src_inc = 10_B and dst_inc = 10_B

- source address = src_data_end_ptr - (n_minus_1 << 2)
- destination address = dst_data_end_ptr - (n_minus_1 << 2)

src_inc = 11_B and dst_inc = 11_B

- source address = src_data_end_ptr
- destination address = dst_data_end_ptr

Table 62 lists the destination addresses for a DMA cycle of six words.

Table 61 DMA Cycle of Six Words Using a Word Increment

Initial values of channel_cfg, prior to the DMA cycle				
src_size = 10_B, dst_inc = 10_B, n_minus_1 = 101_B, cycle_ctrl = 1_B				
DMA Transfers	**End Pointer**	**Count**	**Difference[1)]**	**Address**
	$02AC_H$	5	014_H	0298_H
	$02AC_H$	4	010_H	$029C_H$
	$02AC_H$	3	$0C_H$	$02A0_H$
	$02AC_H$	2	08_H	$02A4_H$
	$02AC_H$	1	04_H	$02A8_H$
	$02AC_H$	0	00_H	$02AC_H$
Final values of channel_cfg, after the DMA cycle				
src_size = 10_B, dst_inc = 10_B, n_minus_1 = 0_B, cycle_ctrl = 0_B				

1) This value is the result of count being shifted left by the value of dst_inc.

Table 62 lists the destination addresses for a DMA transfer of 12 bytes using a halfword increment.

Table 62 DMA Cycle of 12 Bytes Using a Halfword Increment

Initial values of channel_cfg, prior to the DMA cycle				
src_size = 00_B, dst_inc = 01_B, n_minus_1 = 1011_B, cycle_ctrl = 1_B, R_power = 11_B				
DMA Transfers	**End Pointer**	**Count**	**Difference[1)]**	**Address**
	$05E7_H$	11	016_H	$05D1_H$
	$05E7_H$	10	014_H	$05D3_H$
	$05E7_H$	9	012_H	$05D5_H$
	$05E7_H$	8	010_H	$05D7_H$
	$05E7_H$	7	$0E_H$	$05D9_H$
	$05E7_H$	6	$0C_H$	$05DB_H$
	$05E7_H$	5	$0A_H$	$05DD_H$
	$05E7_H$	4	08_H	$05DF_H$
Values of channel_cfg after 2^R DMA transfers				
src_size = 00_B, dst_inc = 01_B, n_minus_1 = 011_B, cycle_ctrl = 1_B, R_power = 11_B				
DMA Transfers	**End Pointer**	**Count**	**Difference[2)]**	**Address**
	$05E7_H$	3	06_H	$05E1_H$
	$05E7_H$	2	04_H	$05E3_H$
	$05E7_H$	1	02_H	$05E5_H$
	$05E7_H$	0	00_H	$05E7_H$
Final values of channel_cfg, after the DMA cycle.				
src_size = 00_B, dst_inc = 01_B, n_minus_1 = 0_B, cycle_ctrl = 0_B, R_power = 11_B				

1) This value is the result of count being shifted left by the value of dst_inc.

2) After the controller completes the DMA cycle it invalidates the channel_cfg memory location by clearing the cycle_ctrl field.

9.4 Register Definition

Please refer to [1], Chapter 3, Programmer's Model and chapter 4, Register Description

Table 63 shows the DMA module base address (for the configuration register).

Table 63 Register Address Space

Module	Base Address	End Address	Note
DMA	50014000_H	$50017FFF_H$	

Table 64 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
STATUS	DMA Status	00_H	$000D\ 0000_H$
CFG	DMA Configuration	04_H	$0000\ 0000_H$
CTRL_BASE_PTR	Channel Control Data Base Pointer	08_H	$0000\ 0000_H$
ALT_CTRL_BASE_PTR	Channel Alternate Control Data Base Pointer	$0C_H$	$0000\ 0100_H$
WAITONREQ_STATUS	Channel Wait on Request Status	10_H	$0000\ 0000_H$
CHNL_SW_REQUEST	Channel Software Request	14_H	$0000\ 0000_H$
CHNL_USEBURST_SET	Channel Useburst Set	18_H	$0000\ 0000_H$
CHNL_USEBURST_CLR	Channel Useburst Clear	$1C_H$	$0000\ 0000_H$
CHNL_REQ_MASK_SET	Channel Request Mask Set	20_H	$0000\ 0000_H$
CHNL_REQ_MASK_CLR	Channel Request Mask Clear	24_H	$0000\ 0000_H$
CHNL_ENABLE_SET	Channel Enable Set	28_H	$0000\ 0000_H$
CHNL_ENABLE_CLR	Channel Enable Clear	$2C_H$	$0000\ 0000_H$
CHNL_PRI_ALT_SET	Channel Primary-Alternate Set	30_H	$0000\ 0000_H$
CHNL_PRI_ALT_CLR	Channel Primary-Alternate Clear	34_H	$0000\ 0000_H$
CHNL_PRIORITY_SET	Channel Priority Set	38_H	$0000\ 0000_H$
CHNL_PRIORITY_CLR	Channel Priority Clear	$3C_H$	$0000\ 0000_H$
ERR_CLR	Bus Error Clear	$4C_H$	$0000\ 0000_H$

The registers are addressed wordwise.

DMA Status Registers

The read-only DMA_STATUS Register returns the status of the controller. You cannot read this register when the controller is in the reset state.

STATUS
DMA Status **(00_H)** **Reset Value: $000D\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				RES							CHNLS_MINUS1				
r				r							r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								STATE				RES			MASTER_ENABLE
r								r				r			r

Field	Bits	Type	Description
RES	31:28	r	**Reserved** Read as 0.
RES	27:21	r	**Reserved** Read as 0.
CHNLS_MINUS1	20:16	r	**Available Channels minus 1** Number of available DMA channels minus one 1101_B **CHCNT**, controller configured to use 14 DMA channels
RES	15:8	r	**Reserved**
STATE	7:4	r	**Current State of the Control State Machine** State can be one of the following bit combinations (other bit combinations are undefined): 0000_B**value0**, idle 0001_B**value1**, reading channel controller date 0010_B**value2**, reading source data end pointer 0011_B**value3**, reading destination data end pointer 0100_B**value4**, reading source data 0101_B**value5**, writing destination data 0110_B**value6**, waiting for DMA request to clear 0111_B**value7**, writing channel controller data 1000_B**value8**, stalled 1001_B**value9**, done 1010_B**value10**, peripheral scatter-gather transition
RES	3:1	r	**Reserved**
MASTER_ENABLE	0	r	**Enable Status of the Controller** 0_B **DISABLED**, controller is disabled 1_B **ENABLED**, controller is enabled

DMA Configuration Register

The read-only dma_status Register returns the status of the controller. You cannot read this register when the controller is in the reset state.

CFG
DMA Configuration **(04_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								CHN1_PROT_CTRL			RES				MASTER_ENABLE
w								w			w				w

Field	Bits	Type	Description
RES	31:8	w	**Reserved** Write as zero.
CHN1_PROT_CTRL	7:5	w	**CHN1_PROT_CTRL** Sets the AHB-Lite protection by controlling the HPROT[3:1] signal levels as follows: bit 7 controls HPROT[3] to indicate if a cacheable access is occurring. bit 6 controls HPROT[2] to indicate if a bufferable access is occurring. bit 5 controls HPROT[1] to indicate if a privileged access is occurring. **Notes** 1. *When bit[n]=1, then the corresponding HPROT is HIGH. When bit[n]=0, then the corresponding HPROT is LOW.* 2. *The CHN1_PROT_CTRL bits must not be changed when the MASTER:ENABLE bit is set because this may cause a protocol error on the AHB master interface. As the DMA_CFG register is write-only the user must read the status of the master enable bit from the DMA_STATUS register.*
RES	4:1	w	**Reserved** Write as zero.
MASTER_ENABLE	0	w	**Enable for the Controller** 0_B **DISABLE**, disables the controller 1_B **ENABLE**, enables the controller

Channel Control Data Base Pointer Register

The CTRL_BASE_PTR register is a read/write register. You must configure this register to that base pointer points to a location in your system memory.

Note: The controller provides no internal memory for storing the channel control data.

The amount of system memory that you must assign to the controller depends on the number of DMA channels and whether you configure it to use the alternate data structure. Therefore, the base pointer address requires a variable number of bits that depend on the system implementation.

You cannot read this register when the controller is in the reset state. **Figure 50** shows the possible bit assignments for this register, depending on the number of DMA channels that you configure the controller to contain

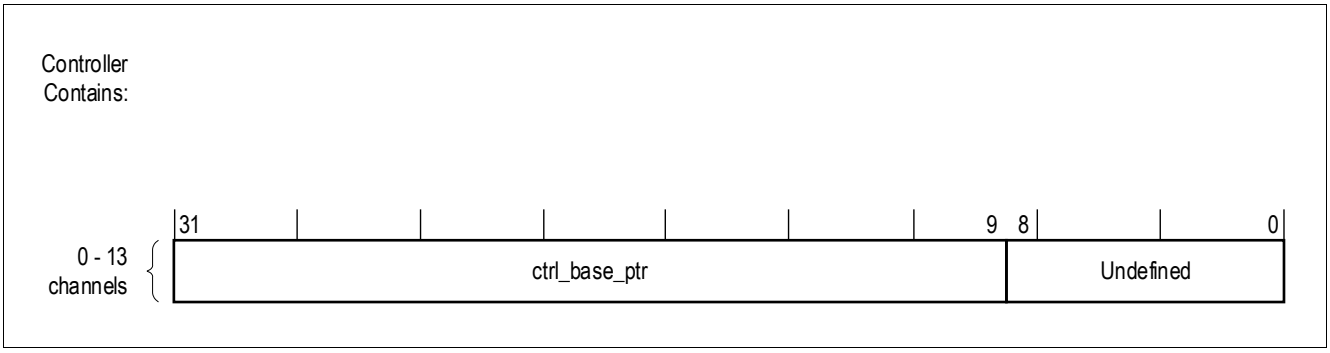


Figure 50 CTRL_BASE_PTR Register Bit Assignments

CTRL_BASE_PTR
Channel Control Data Base Pointer **(08_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTRL_BASE_PTR															
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL_BASE_PTR							**RES**								
rw							r								

Field	Bits	Type	Description
CTRL_BASE_PTR	31:9	rw	**CTRL_BASE_PTR** Pointer to the base address of the primary data structure. See **Chapter 9.2** for information about the data structure.
RES	8:0	r	**Reserved** read as zero.

Channel Alternate Control Data Base Pointer Registers

The read-only ALT_CTRL_BASE_PTR Register returns the base address of the alternate data structure. You cannot read this register when the controller is in the reset state.

ALT_CTRL_BASE_PTR
Channel Alternate Control Data Base Pointer (0C$_H$) Reset Value: 0000 0100$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALT_CTRL_BASE_PTR															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALT_CTRL_BASE_PTR															
r															

Field	Bits	Type	Description
ALT_CTRL_BASE_PTR	31:0	r	**Base Address of the Alternate Data Structure**

Channel Wait on Request Status Register

The read-only DMA_WAITONREQ_STATUS Register returns the status of dma_waitonreq[].You cannot read this register when the controller is in the reset state.

WAITONREQ_STATUS
Channel Wait on Request Status (10$_H$) Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		WAITONREQ_STATUS													
r		r													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
WAITONREQ_STATUS	13:0	r	**Channel Wait on Request Status** Read as for each bit C (C = 13 to 0): 0_B **LOW**, dma_waitonreq[C] is LOW. 1_B **HIGH**, dma_waitonreq[C] is HIGH.

Channel Software Request Register

The write-only CHNL_SW_REQUEST Register enables you to generate a software DMA request.

CHNL_SW_REQUEST
Channel Software Request **(14_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_SW_REQUEST													
r		w													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_SW_REQUEST	13:0	w	**CHNL_SW_REQUEST** Set the appropriate bit to generate a software DMA request on the corresponding DMA channel. Write as for each bit C (C = 13 to 0): *Note: Writing to this bit where a DMA channel is not implemented does not create a DMA request for that channel.* 0_B **NOREQ**, does not create a DMA request for channel C. 1_B **SWREQ**, creates a DMA request for channel C.

Channel Useburst Set Register

The read/write CHNL_USEBURST_SET Register disables the single request dma_sreq[] input from generating requests, and therefore only the request, dma_req[], generates requests. Reading the register returns the useburst status.

CHNL_USEBURST_SET
Channel Useburst Set **(18_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_USEBURST_SET													
r		rw													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_USEBURST_SET	13:0	rw	**CHNL_USEBURST_SET** Returns the useburst status, or disables dma_sreq[C] from generating DMA requests. For each bit C (C = 13 to 0) *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **value0**, on read: DMA channel n responds to requests that it receives on dma_req[C] or dma_sreq[C]. The controller performs 2^R, or single, bus transfers. 1_B **value1**, on read: DMA channel n does not respond to requests that it receives on dma_req[C] or dma_sreq[C]. The controller only responds to dma_req[C] requests and performs 2^R transfers. 0_B **value0**, on write: No effect. Use the CHNL_USEBURST_CLR Register to set bit [C] to 0. 1_B **value1**, on write: Disables dma_sreq[C] from generating DMA requests. The controller performs 2^R transfers.

After the penultimate 2^R transfer completes, if the number of remaining transfers, N, is less than 2^R then the controller resets the CHNL_USEBURST_SET bit to 0. This enables you to complete the remaining transfers using dma_req[ ] or DMA_SREQ[ ].

Note: If you program CHANNEL_CFG with a value of N less than 2^R then you must not set the corresponding CHNL_USEBURST_SET bit, if the peripheral does not assert dma_req[ ].

In peripheral scatter-gather mode, if the next_useburst bit is set in CHANNEL_CFG then the controller sets the CHNL_USEBURST_SET [n] bit to a 1, when it completes the DMA cycle that uses the alternate data structure.

Channel Useburst Clear Register

The write-only CHNL_USEBURST_CLR Register enables dma_sreq[] to generate requests.

CHNL_USEBURST_CLR
Channel Useburst Clear **($1C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_USEBURST_CLR													
r		w													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_USEBURST_CLR	13:0	w	**CHNL_USEBURST_CLR** Set the appropriate bit to enable dma_sreq[] to generate requests. Write as for each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **NOEFFECT**, No effect. Use the CHNL_USEBURST_SET Register to disable dma_sreq[] from generating requests. 1_B **ENABLE**, Enables dma_sreq[C] to generate DMA requests.

Channel Request Mask Set Register

The read/write CHNL_REQ_MASK_SET Register disables a HIGH on dma_req[], or dma_sreq[], from generating a request. Reading the register returns the request mask status for dma_req[] and dma_sreq[].

CHNL_REQ_MASK_SET
Channel Request Mask Set **(20_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_REQ_MASK_SET													
r		rw													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_REQ_MASK_SET	13:0	rw	**CHNL_REQ_MASK_SET** Returns the request mask status of dma_req[] and dma_sreq[], or disables the corresponding channel from generating DMA requests. For each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **value0**, on read: External requests are enabled for channel C. 1_B **value1**, on read: External requests are disabled for channel C. 0_B **value0**, on write: No effect. Use the CHNL_REQ_MASK_CLR Register to enable DMA requests. 1_B **value1**, on write: Disables dma_req[C] and dma_sreq[C] from generating DMA requests.

Channel Request Mask Clear Register

The write-only CHNL_REQ_MASK_CLR Register enables a HIGH on dma_req[], or dma_sreq[], to generate a request.

CHNL_REQ_MASK_CLR
Channel Request Mask Clear **(24_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_REQ_MASK_CLR													
r		w													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_REQ_MASK_CLR	13:0	w	**CHNL_REQ_MASK_CLR** Set the appropriate bit to enable DMA requests for the channel corresponding to dma_req[] and dma_sreq[]. Write as for each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **NOEFFECT**, No effect. Use the CHNL_REQ_MASK_SET Register to disable dma_req[] and dma_sreq[] from generating requests. 1_B **ENABLE**, Enables dma_req[C] or dma_sreq[C] to generate DMA request.

Channel Enable Set Register

The read/write CHNL_ENABLE_SET Register enables you to enable a DMA channel. Reading the register returns the enable status of the channels.

CHNL_ENABLE_SET
Channel Enable Set **(28_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_ENABLE_SET													
r		rw													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_ENABLE_SET	13:0	rw	**CHNL_ENABLE_SET** Returns the enable status of the channels, or enables the corresponding channels. For each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **DISABLED**, on read: Channel C is disabled. 1_B **ENABLED**, on read: Channel C is enabled. 0_B **NOEFFECT**, on write: No effect. Use the CHNL_ENABLE_CLR Register to disable a channel. 1_B **ENABLE**, on write: Enables channel C.

Channel Enable Clear Register

The write-only CHNL_ENABLE_CLR Register enables you to disable a DMA channel.

CHNL_ENABLE_CLR
Channel Enable Clear **($2C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_ENABLE_CLR													
r		w													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_ENABLE_CLR	13:0	w	**CHNL_ENABLE_CLR** Set the appropriate bit to disable the corresponding DMA channel. Write as for each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **NOEFFECT**, No effect. Use the CHNL_ENABLE_SET Register to enable DMA channels. 1_B **DISABLE**, Disables channel C.

Note: The controller disables a channel, by setting the appropriate bit, when either:

- it completes the DMA cycle
- it reads a channel_cfg memory location which has cycle_ctrl = 000_B
- an ERROR occurs on the AHB-Lite bus

Channel Primary-Alternate Set Register

The read/write CHNL_PRI_ALT_SET Register enables you to configure a DMA channel to use the alternate data structure. Reading the register returns the status of which data structure is in use for the corresponding DMA channel.

CHNL_PRI_ALT_SET
Channel Primary-Alternate Set **(30_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_PRI_ALT_SET													
r		rw													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_PRI_ALT_SET	13:0	rw	**CHNL_PRI_ALT_SET** Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channel. For each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **PRIMSTRUC**, on read: DMA channel C is using the primary data structure. 1_B **ALTSTRUC**, on read: DMA channel C is using the alternate data structure. 0_B **NOEFFECT**, on write: No effect. Use the CHNL_PRI_ALT_CLR Register to set bit [C] to 0. 1_B **ALTSTRUC**, on write: Selects the alternate data structure for channel C.

Note: The controller toggles the value of the CHNL_PRI_ALT_SET [C] bit after it completes:

- the four transfers that the primary data structure specifies for a memory scatter-gather, or peripheral scatter-gather, DMA cycle
- all the transfers that the primary data structure specifies for a ping-pong DMA cycle
- all the transfers that the alternate data structure specifies for the following DMA cycle types:
  - ping-pong
  - memory scatter-gather
  - peripheral scatter-gather

Channel Primary-Alternate Clear Register

The write-only CHNL_PRI_ALT_CLR Register enables you to configure a DMA channel to use the primary data structure.

CHNL_PRI_ALT_CLR
Channel Primary-Alternate Clear **(34_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_PRI_ALT_CLR													
r		w													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_PRI_ALT_CLR	13:0	w	**CHNL_PRI_ALT_CLR** Set the appropriate bit to select the primary data structure for the corresponding DMA channel. Write as for each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **NOEFFECT**, No effect. Use the CHNL_PRI_ALT_SET Register to select the alternate data structure. 1_B **PRIMSTRUC**, Selects the primary data structure for channel C.

Note: The controller toggles the value of the chnl_pri_alt_clr [C] bit after it completes:

- the four transfers that the primary data structure specifies for a memory scatter-gather, or peripheral scatter-gather, DMA cycle
- all the transfers that the primary data structure specifies for a ping-pong DMA cycle
- all the transfers that the alternate data structure specifies for the following DMA cycle types:
  - ping-pong
  - memory scatter-gather
  - peripheral scatter-gather

Channel Priority Set Register

The read/write CHNL_PRIORITY_SET Register enables you to configure a DMA channel to use the high priority level. Reading the register returns the status of the channel priority mask.

CHNL_PRIORITY_SET
Channel Priority Set **(38_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_PRIORITY_SET													
r		rw													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_PRIORITY_SET	13:0	rw	**CHNL_PRIORITY_SET** Returns the channel priority mask status, or sets the channel priority to high. For each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **DEFPRIO**, on read: DMA channel C is using the default priority level. 1_B **HIGHPRIO**, on read: DMA channel C is using a high priority level. 0_B **NOEFFECT**, on write: No effect. Use the CHNL_ENABLE_CLR Register to set channel C to the default priority level. 1_B **HIGHPRIO**, on write: Channel C uses the high priority level.

Channel Priority Clear Register

The write-only CHNL_ENABLE_CLR Register enables you to configure a DMA channel to use the default priority level.

CHNL_PRIORITY_CLR
Channel Priority Clear **($3C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		CHNL_PRIORITY_CLR													
r		w													

Field	Bits	Type	Description
RES	31:14	r	**Reserved** Always read as 0
CHNL_PRIORITY_CLR	13:0	w	**CHNL_PRIORITY_CLR** Set the appropriate bit to select the default priority level for the specified DMA channel. Write as for each bit C (C = 13 to 0): *Note: Writing to a bit where a DMA channel is not implemented has no effect.* 0_B **NOEFFECT**, No effect. Use the CHNL_ENABLE_SET Register to set channel C to the high priority level. 1_B **DEFPRIO**, Channel C uses the default priority level.

Bus Error Clear Register

The read/write ERR_CLR Register returns the status of dma_err, and enables you to set dma_err LOW.

ERR_CLR
Bus Error Clear **($4C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES															ERR_CLR
r															rw

Field	Bits	Type	Description
RES	31:1	r	**Reserved** Write as zero.
ERR_CLR	0	rw	**ERR_CLR** Returns the status of dma_err, or sets the signal LOW. *Note: For test purposes, use the err_set register to set dma_err HIGH.* 0_B **LOW**, on read: dma_err is LOW. 1_B **HIGH**, on read: dma_err is HIGH. 0_B **NOEFFECT**, on write: No effect, status of dma_err is unchanged. 1_B **LOW**, on write: Sets dma_err LOW.

Note: If you deassert dma_err at the same time as an ERROR occurs on the AHB-Lite bus, then the ERROR condition takes precedence and dma_err remains asserted.

10 Address space organization

The TLE987x manipulates operands in the following memory spaces:

- Up to 256 KByte (incl. 4 KByte emulated EEPROM) of Flash memory in code space
- 32 KByte Boot ROM memory in code space (used for boot code and IP storage)
- Up to 8 KByte RAM memory in code space and data space (RAM can be read/written as program memory or external data memory)
- Special function registers (SFRs) in peripheral space

The figure below shows the detailed address alignment of TLE987x:

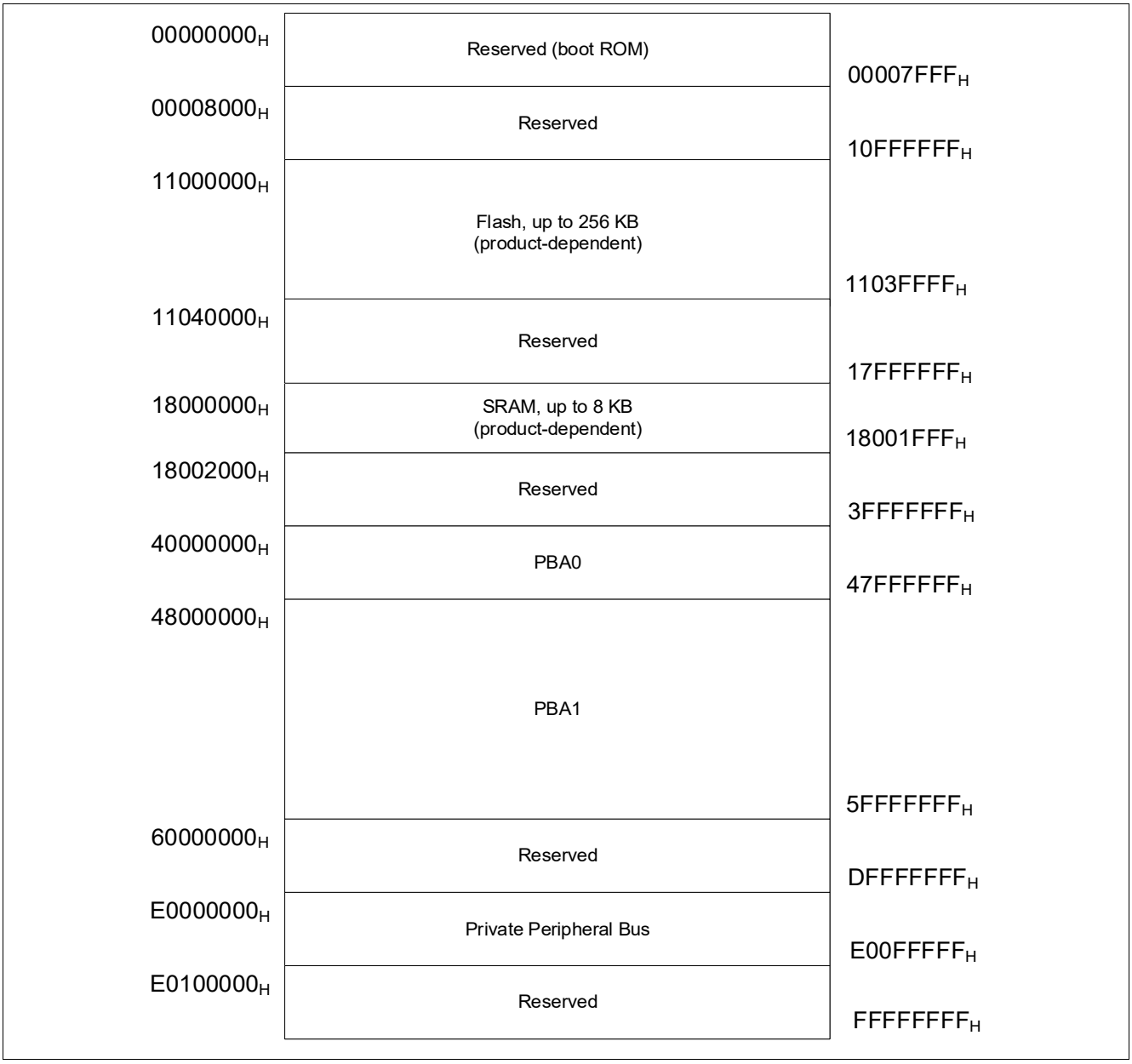


Figure 51 TLE987x Memory Map

Each module provides, beside the physical memory implementation, standard AHB-Lite interface and Error Correction Code (ECC) logic if needed.

Table 65 Memory Map

Start (hex)	End (hex)	Size (hex)	Space Name	Usage
0000_0000	0000_7FFF	8000	Code/Data	Boot ROM
0000_8000	10FF_FFFF		Reserved	Reserved
1100_0000	1103_FFFF	40000	Code/Data	NVM, up to 256 KB
1104_0000	17FF_FFFF		Reserved	Reserved
1800_0000	1800_1FFF	2000	Code/Data	SRAM, up to 8 KB
1800_2000	3FFF_FFFF			Reserved
4000_0000	47FF_FFFF	08000000	Peripheral 0	Peripheral 0 (PBA0)
4800_0000	5FFF_FFFF	08000000	Peripheral 1	Peripheral 1 (PBA1)
6000_0000	DFFF_FFFF		Reserved	reserved
E000_0000	E00F_FFFF	00100000	PPB, Private Peripheral Bus	CPU
E010_0000	EFFF_FFFF		Vendor specific 1	Reserved
F000_0000	FFFF_FFFF		Vendor specific 2	Reserved

Table 66 Peripheral Memory Map

Bus Structure	Modules	Start Address	End Address
Peripherals 0	Reserved	40000000_H	$40003FFF_H$
	ADC1	40004000_H	$40007FFF_H$
	ADC34[1)]	40008000_H	$4000BFFF_H$
	CCU6	$4000C000_H$	$4000FFFF_H$
	GPT12	40010000_H	$40013FFF_H$
	Reserved	40014000_H	$40017FFF_H$
	Reserved	40018000_H	$4001BFFF_H$
	Reserved	$4001C000_H$	$4001FFFF_H$
	Reserved	40020000_H	$40023FFF_H$
	Reserved	40024000_H	$40027FFF_H$
	Reserved	40028000_H	$4002BFFF_H$
	Reserved	$4002C000_H$	$4002FFFF_H$
	Reserved	40030000_H	$40033FFF_H$
	DRV	40034000_H	$40037FFF_H$
	Reserved	40038000_H	$4003BFFF_H$
	Reserved	$4003C000_H$	$4003FFFF_H$
	Reserved	40044000_H	$40047FFF_H$
	Reserved	40048000_H	$47FFFFFF_H$
Peripherals 1	Reserved	48000000_H	$48003FFF_H$
	TIMER2	48004000_H	$48004FFF_H$
	TIMER21	48005000_H	$48005FFF_H$
	TIMER3	48006000_H	$48007FFF_H$
	Reserved	48008000_H	$4800BFFF_H$
	Reserved	$4800C000_H$	$4800FFFF_H$
	Reserved	48010000_H	$48013FFF_H$
	Reserved	48014000_H	$48017FFF_H$
	MF	48018000_H	$4801BFFF_H$
	ADC2	$4801C000_H$	$4801DFFF_H$
	LIN	$4801E000_H$	$4801FFFF_H$
	UART1	48020000_H	$48021FFF_H$
	UART2	48022000_H	$48023FFF_H$
	SSC1	48024000_H	$48025FFF_H$
	SSC2	48026000_H	$48027FFF_H$
	PORT	48028000_H	$48029FFF_H$
	Reserved	$4802A000_H$	$4802BFFF_H$
	Reserved	$4802C000_H$	$4802DFFF_H$
	Reserved	$4802E000_H$	$4802FFFF_H$

Table 66 Peripheral Memory Map (cont'd)

Bus Structure	Modules	Start Address	End Address
	Reserved	48030000_H	$4FFFFFFF_H$
	Reserved	48004000_H	$4801FFFF_H$
	Reserved	50000000_H	$50003FFF_H$
	PMU	50004000_H	$50004FFF_H$
	SCU	50005000_H	$50005FFF_H$
	SCUPM	50006000_H	$50006FFF_H$
	Reserved	50007000_H	$50007FFF_H$
	Reserved	50008000_H	$50009FFF_H$
	Reserved	$5000A000_H$	$5000BFFF_H$
	Reserved	$5000C000_H$	$5000FFFF_H$
	Reserved	50010000_H	$50013FFF_H$
	DMA	50014000_H	$50017FFF_H$
	Reserved	50018000_H	$5001BFFF_H$
	Reserved	$5001C000_H$	$5001FFFF_H$
	Reserved	50020000_H	$50023FFF_H$
	Reserved	50024000_H	$50027FFF_H$
	Reserved	50028000_H	$5002BFFF_H$
	Reserved	$5002C000_H$	$57FFFFFF_H$
	Reserved	58000000_H	$58003FFF_H$
	Reserved	58004000_H	$58007FFF_H$
	Reserved	58008000_H	$5800BFFF_H$
	Reserved	$5800C000_H$	$5800FFFF_H$
	Reserved	58010000_H	$58013FFF_H$
	Reserved	58014000_H	$58017FFF_H$
	Reserved	58018000_H	$5FFFFFFF_H$

1) Not available in all products of the TLE987x family. See the datasheet for details.

11 Memory control unit

11.1 Features

- Handles all system memory types and their interaction with the CPU
- Memory protection for D-flash and P-flash system memory
- Address management with access violation detection including reporting
- Linear address range for all memory types (no paging)

11.2 Introduction

11.2.1 Block diagram

The memory control unit is divided into the following submodules:

- NVM memory module (embedded flash memory)
- RAM memory module
- BootROM memory module
- Memory protection unit (MPU) module
- Peripheral bridge PBA0
- LMB (Local Memory Bus) interface logic.

A block diagram view of the MCU, together with the main interface signals, is shown in the **Figure 52**.

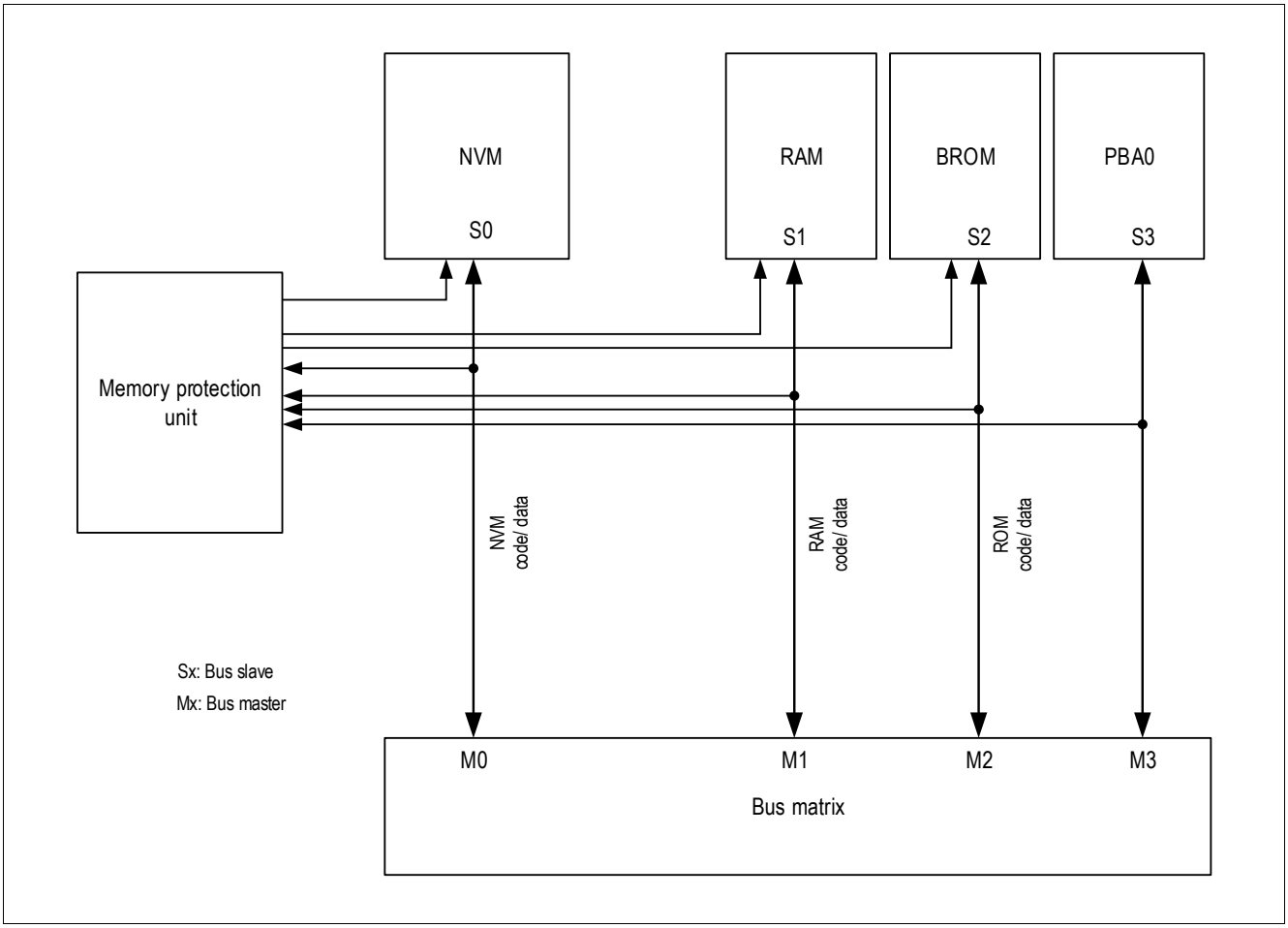


Figure 52 Block diagram of the memory control unit

As shown in the **Figure 52**, the MCU interface communicates with the external world, mainly the core, via 4 AHB-Lite interfaces, Data/Code access to the NVM, BootROM and RAM plus an access to the NVM internal registers. The AMBA bus matrix block decodes the access requests coming from the masters and forwards them to the target module interface together with the required sideband signals. The AMBA bus matrix block provides all the needed interface functions between the masters and the memory peripheral. It will generate proper HSEL signals, and multiplex the response coming from the modules. In addition, the AMBA bus matrix block takes care of forwarding the transfer according the a fixed priority policy described in the AMBA chapter.

Besides the AHB-Lite and sideband signals, the MCU has access to further Core specific signals, relevant for memory protection .

11.3 NVM module (flash memory)

The flash memory provides embedded user-programmable non-volatile memory, allowing fast and reliable storage of user code and data.

Features

- In-system programming via LIN (flash mode) and SWD.
- Error correction code (ECC) for detection of single-bit and double-bit errors and dynamic correction of single-bit errors.
- Interrupts and signals double-bit errors by the NMI.
- Program width of 128 byte (page).
- Minimum erase width of 128 bytes (page).
- Integrated hardware support for EEPROM emulation.
- 8-byte read access.
- Physical read access time: 75 ns.
- Code-read access acceleration integrated; read buffer and automatic pre-fetch.
- Page-program time: t_{PR} (see data sheet for value).
- Page-erase (128 bytes) and sector-erase (4 KB) time: t_{ER} (see data sheet for value).
- Erased bit (cell) is read as '1', for code flash and 100TP.
- Erased bit (cell) is read as '0' plus NMIMAP request, for data flash.

Note: The user has to ensure that no flash operations which change the content of the flash get interrupted at any time.

The clock for the NVM is supplied with the system frequency f_{sys}. Integrated firmware routines for erasing NVM, EEPROM emulation, and other operations are provided.

The TLE987x NVM module provides physical implementation of the memory module as well as needed complementary features and interface towards the core.

The TLE987x NVM module consists of the memory cell array and all the control circuits and registers needed to access the array itself. The up to 256 Kbyte data module is mapped in the Arm® Cortex®-M3 code address range 11000000H - 1103FFFFH.

Access of NVM module is granted through the AMBA matrix block that forwards to the memory modules AHB-Lite interfaces the requests generated by the masters according to the defined priority policy.

11.3.1 Definitions

This section defines the nomenclature and some abbreviations. The used flash memory is a non-volatile memory (“NVM”) based on a floating gate one-transistor cell. It is called “non-volatile” because the memory content is kept when the memory power supply is shut off.

11.3.1.1 General Definitions

Logical and Physical states

Erasing

The erased state of a cell is ´1´. Forcing an NVM cell to this state is called erasing. Erasing is possible with a granularity of a page (see below).

Writing

The written state of a cell is ´0´. Forcing an NVM cell to this state is called writing. Each bit can be individually written.

Programming

The combination of erasing and writing is called ‘programming’. Programming often means also writing a previously erased page.

The wording ‘write’ or ‘writing’ are also used for accessing special function registers and the assembly buffer. The meaning depends therefore on the context.

The above listed processes have certain limitations:

Retention: This is the time during which the data of a flash cell can be read reliably. The retention time is a statistical figure that depends on the operating conditions of the flash array (temperature profile) and the accesses to the flash array. With an increasing number of program/erase cycles (see endurance) the retention is lowered. Drain and gate disturbs decrease data retention as well.

Endurance: As described above, the data retention is reduced with an increasing number of program/erase cycles. A flash cell incurs one cycle whenever its page or sector is erased. This number is called “endurance”. As said for the retention, it is a statistical figure that depend on operating conditions and the use of the flash cells and on the required quality level.

Drain Disturb: Because of using a so called “one-transistor” flash cell each program access disturbs all pages of the same sector slightly. Over long these “drain disturbs” make 0 and 1 values indistinguishable and thus provoke read errors. This effect is again interrelated with the retention. A cell that incurred a high number of drain disturbs will have a lower retention. The physical sectors of the flash array are isolated from each other. So pages of a different sector do not incur a drain disturb. this effect must be therefore considered when the page erase feature is used or when re-programming an ready programmed page (implicitly causing an erase of the page before writing the new data).

Data Portions

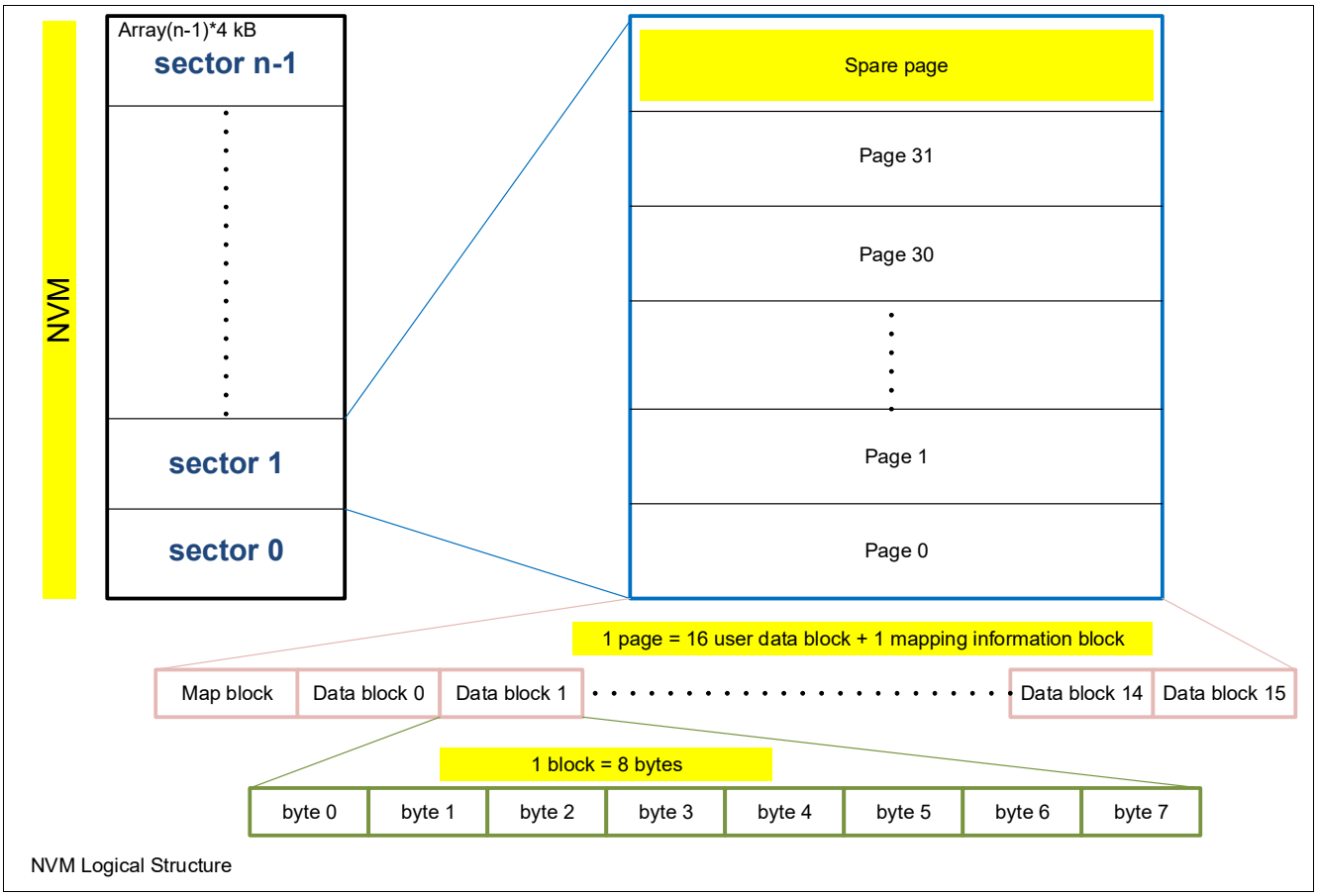


Figure 53 Logical structure of the NVM core

Doubleword

A doubleword consists of 64 bits. A doubleword represents the data size that is read from or written to the NVM core module within one access cycle.

Block

A block consists of one doubleword and its associated ECC data (64 bit data and 8 bit ECC). A block represents the smallest data portion that can be changed in the assembly buffer. Since the ECC protects 64 bits, when a byte is written to the assembly buffer automatically an NVM internal read of the complete block is triggered, the byte and the ECC are updated and the complete block is written back to the assembly buffer.

Mapblock

A map block consists of a module specific number of ECC -protected bits that hold the necessary information to map a physical page to a logical page.

Page

A page consists of 16 blocks and one map block.

Spare page

A spare page is an additional page in a sector used in each programming routine to allow tearing-safe programming.

Sector

A sector consists of 32 logical and 33 physical page.

11.3.2 Functional Description

The main tasks of the NVM module are reading form the memory array, writing to the assembly buffer, enabling (tearing safe) programming of a single page, provide basic in-module functionality for code protection and prefetch feature for optimized system performance. The main features are listed following:

- Up to 256 KB memory size
- 4 KB configuration sector
- write time t_{PR} per page
- erase time t_{ER} per page
- Error correction and Error Detection code (ECC and EDC)
- In module memory protection logic
- Prefetch logic for optimized module performance

11.3.2.1 Basic Block Functions

Figure 54 shows a schematic block diagram of the NVM module

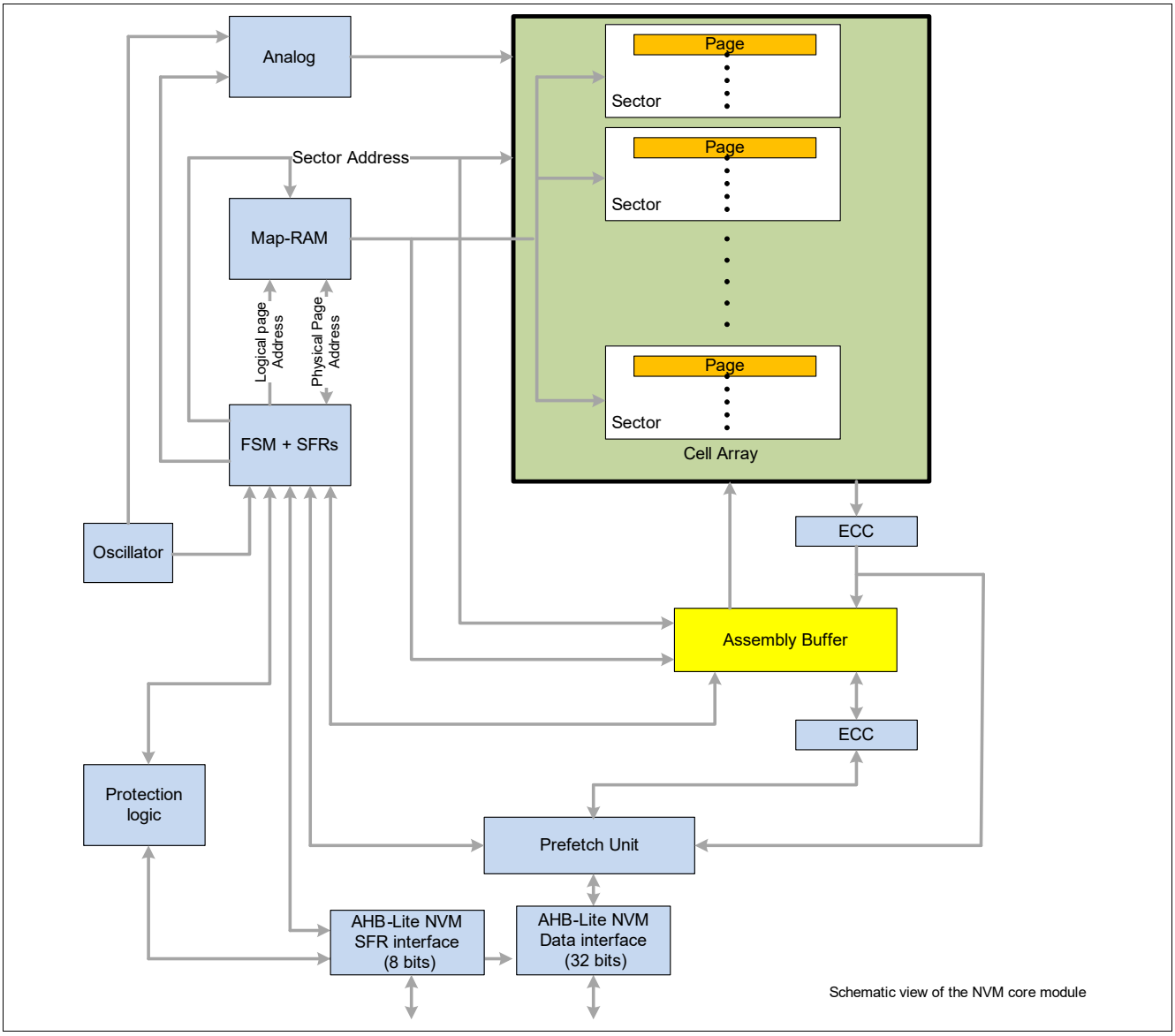


Figure 54 Schematic view of the NVM core module

11.3.2.2 Memory Cell Array

The non-volatile memory cells are organized in sectors, which consists of pages, which are structured in blocks and map block.

Page

Each page consists of 16 data blocks of 64 bits each and one map block. The map block stores the mapping information of the page in the sector. All blocks of a page are ECC-protected.

A page is the smallest granularity of data that can be changed (erased or written) within the cell array. One data block is the minimum granularity of data that can be read from the NVM module within memory read access.

Employing the integrated EEPROM emulation using the map RAM, the minimum granularity of data that can be changed in the NVM is one byte, while all other bytes in the page do not change.

Assembly Buffer

The assembly buffer is a RAM that can hold the content of one page including the map block.

Sector

A sector consists of 33 physical pages. 32 pages can be logically addressed during a memory access. One page is internally used as a spare page.

Map RAM

The map RAM is a static RAM that holds the mapping of a logical page addresses to physical page addresses for each sector. Is is completely handled by the NVM programming related BootROM routines. The map-RAM currently is sized to support a max of 16 map-RAM mapped sectors.

FSM and SFR Block

This block contains the special function registers (SFRs) of the NVM module. Beside memory reads and writes to the assembly all interactions of the BootROM software with the module take places through register accesses. The finite state machine (FSM) controls the actions (e.g. read, erase and write) of the NVM module.

Analog Components

The module contains analog components to provide all the voltages necessary for erasing, writing and reading the non-volatile memory cells.

11.3.2.3 SFR Accesses

All SFRs can only be accessed through the NVM related BootROM routines, that is, the customer software cannot access the SFRs directly but has to use BootROM routines.

11.3.2.4 Memory Read

The NVM memory internally can be read with a minimum granularity of one block (64 data bits).

If the block is not within the memory address range of the NVM module, the module does not react at all and a different memory module may handle the access.

Memory read accesses are only possible while no FSM procedures (program, init, sleep or copy) is in progress. A memory read access while the FSM is busy is stalled as long as the FSM is busy and the access is carried out when the FSM is in idle mode again.

Since a read to the memory field takes a fixed time mostly independent of the system frequency, an optimized number of waitstates (3, 1 or, 0) is generated for different system frequencies.

Furthermore, a module internal read buffer holds the block read last. An access to an address within this block does not trigger a new reading from the memory field but is directly served from the read buffer.

To reduce system performance penalties due to the waitstates, a special local in-module prefetch unit is implemented.

11.3.2.5 Memory Write

Data is not written to the memory array directly, but to the assembly buffer and then copied into the cell array by the write sequence.

Memory writes are handled through the BootROM software, which at first copies the existing content of a page to the assembly buffer, allows the user to modify the content of the assembly buffer and afterwards executes the programming of the data to the memory field followed by a verification step.

11.3.2.6 Timing

The target timing of the hardware sequences excluding the software overhead is shown below:

Erase: t_{ER} per page

Write: t_{PR} per page

Program (= Erase+ Write): $t_{ER} + t_{PR}$ per page

The disturb handling routine when enabled with a probability of a approximately 0.1% adds additional t_{ER} + t_{PR} to a page write or program operation.

11.3.2.7 Verify

The data programmed by the BootROM function is verified by the BootROM routine itself. The programmed data in the cell array is compared with the data still available in the assembly buffer. This is done using suitable hard-read levels. These hard-read levels provide a margin compared to the normal read level to ensure that the data is actually programmed with suitably distinct levels for written and erased bits.

11.3.2.8 Tearing-Safe programming

The mapping mechanism of the NVM module is used like a log-structured file system: When a page is programmed in the sector the old values are not physically overwritten, but a different physical page (spare page) is programmed in the same sector in fact. If the programming fails (e.g. because of power loss during the erase or write procedure), the old values are still present in the sector. The BootROM routines therefore can program a single page in a tearing-safe way.

When an erase or write procedure to the memory field was interrupted by a power-down, this is identified during the reconstruction of the map-RAM content after the next reset. In this case, a special routine in the BootROM (called Service Algorithm) is automatically started, identifies this tearing case of respective logical page and repairs the NVM state, ensuring that either the old or the new data (or both) are fully valid.

11.3.2.9 Disturb handling

Due to the implementation of the cell array, while writing a page into the cell array all other pages within the same sector are slightly written (disturbed) too. If some pages of a sector are changed often and other pages of the same sector only rarely, these rarely programmed pages may be disturbed too often and loose their data.

If the disturbs for a page exceed a specific value (this happens only when a different page in the same sector is programmed), the page has to be reprogrammed (refreshed). A dedicate option of the programming routines provided with the BootROM make sure that the pages are refreshed in time.

As mentioned, the refreshing of a page - when actually triggered - will double the overall programming time.

11.3.2.10 ECC and EDC

The NVM module provides all needed logic for proper error correction and detection logic. Since the block is the smallest data portion used for accessing the array in read and write, the ECC and EDC are performed at block level. Requirement is to provide a single bit ECC and 2 bits EDC per block, that is 1 bit correction over 64 data bits.

Since the ECC protects 64bits, when a byte is written to the assembly buffer automatically an NVM internal read of the complete block is triggered, the byte and the ECC are updated and the complete block is written back to the assembly buffer.

11.3.2.11 Resume from disturbed Program/Erase operation

If a NVM operation like Program or Erase was interrupted by any means, then a data integrity check of the data flash is required. The data integrity check can be done by performing a cold reset, power-up reset, pin reset, WDT1 reset or exit from SleepMode. All these resets are running through the MapRAM Initialization of the BootROM, which executes the Service Algorithm in case a data integrity issue inside the data flash was detected. The Service Algorithm tries to resolve a data integrity issue by erasing erroneous data flash pages in order to maintain an proper data flash mapping. The return value of the Service Algorithm is provided inside the register MEMSTAT to the user application. The user application has to evaluate the **MEMSTAT** register in order to perform appropriate corrective actions if needed. Furthermore the register **SYS_STRTUP_STS** provides status information about the MapRAM Initialization function executed during start-up. It allows the user directly to judge the data integrity of the data flash. In case the **SYS_STRTUP_STS** register reports a MapRAM Initialization fail it is not recommended to perform any further write operation to the data flash, as this might result in unrecoverable loss of data integrity inside the data flash. A reinitialization of the data flash by performing a SECTOR_ERASE will then be the only solution. Instead a reset of the device might be triggered in order to execute the Service Algorithm. If even the Service Algorithm fails to resolve the data integrity issue then the data flash sector has to be reinitialized. In order to provide full reliability of the data flash module and to avoid any loss of data integrity inside the data flash the user has to ensure that no NVM operation which changes the content of the data flash module, program or erase, get interrupted at any time. Appropriate actions to support this could be:

- the capacitor at the VS input has to be dimensioned large enough to provide enough charge to the device to keep the VS supply in the specified range until the NVM operation ended normally
- check the supply voltage to be high enough and stable before a NVM operation gets started in order to end the NVM operation normally without interruption
- disable interrupts in the system before a NVM operation gets started, re-enable the interrupts upon return
- avoid nested NVM operations
- trigger the WDT1 in short-open-window mode for any NVM operation
- evaluate the return values of the NVM operations and perform corrective actions accordingly
- check the data integrity of the data flash by executing the USER_MAPRAM_INIT function and perform corrective actions accordingly

Note: The above mentioned recommendation do also apply to NVM write/erase operations to the code flash and as well as to the 100TP pages.

11.3.2.12 Code and Data Access through the AHB-Lite Interface

The system provides access to the data stored in the NVM cell array through an AHB Lite interface. Whenever the core needs to fetch instructions or read data form or write data into the NVM module, a proper AHB Lite compliant access request is forwarded by the bus matrix block into the module.

11.3.2.13 Prefetch Unit

At typical system working frequency, each NVM read access into the array takes 3 system clock cycles to move into the read buffer a complete block (containing 64 data bits). Since the on-chip micro (Arm® Cortex®-M3) is 32-bit microprocessor with an instruction set made of 16-bit and 32 bit opcodes, the amount of data read in 1 array access contains from a minimum of 2 to a max of 4 instructions. For this reason, in case of 32-bit instructions sequential code execution, 3 waitstates every 2 executed instructions have to be added.

In order to reduce the performance penalties due to the access time of the NVM, an internal prefetch logic is implemented. The idea is to use a prefetch unit that, assuming a sequential code execution, starts reading in a second read buffer the next 64 data bits while the system is still moving out the previously accessed data.

The idea is to have two read buffers that work interleaved: while the first read buffer is used as source of data for the AHB-Lite buffer interface, the second one is starting in parallel a new cell memory array access, assuming that data will be required in sequential order. When all the data from the first read buffer have been moved out of the memory module and the second one has completed the array access, then this last one will start providing data on the external bus while the first one will start prefetching next data from the array.

In case, a new non-sequential access request comes while both read buffers are filled with data not yet required by the core, the read buffer which was latest filled will be made available, flushing its content, to serve as soon as possible the non-sequential access request.

11.4 BootROM Module

The TLE987x BootROM module provides physical implementation of the memory module as well as needed complementary features and interface towards the core.

The BootROM module in TLE987x has a capacity of 24 Kbyte, organized with words of 32 bits.

The BootROM contents consists basically of three parts, used for:

startup and boot SW

Boot Strap Loader routines

User routines

11.4.1 BootROM addressing

The BootROM, as visible from the memory map, is mapped starting at the address range 00000000_H - $00007FFF_H$. After any reset, the device hardware-controlled start address is 00000000_H. At this location, the first instruction of the startup procedure is stored.

11.4.2 BootROM firmware program structure

The BootROM firmware provides basic functionality required to be executed after reset and routines for specific operation, such as:

Startup routines, which is the main control firmware in the BootROM executed after every reset. This routine checks which kind of reset was issued and accordingly preforms different kinds of operation to proper configure the device.

Bootstrap loader, which provides basic functionality for code and data upload via LIN or UART into the RAM or NVM module.

User routines, which provide functions for proper NVM operation handling and other useful ready-to -use routines designed for the customer.

For a complete description of the BootROM code, please refer to the BootROM Manual.

11.5 RAM Module

The TLE987x RAM module provides physical implementation of the memory module as well as needed complementary features and interface towards the core.

The RAM module in TLE987x has a capacity of up to 8 Kbyte, organized with words of 32 bits.

The module support 1 bit Error correction and 2 bits error detection per 32-bit word (actually requiring 7 bits parity per word). When an ECC error occurs, the corresponding status flag in the register EDCSTAT will be set. A double bit error can be configured via the interrupt enable bit in register EDCCON to trigger an exception.

11.5.1 RAM Addressing

The RAM, as visible from the memory map, is mapped at the address range 18000000H - 18001FFFH.The module is mapped in the code area of the M3 map regions and can be used as program memory for code fetching as well as data storing.

11.6 Memory Protection Unit (MPU)

The target of the memory protection scheme is to prevent unintentional writing to and reading from the NVM, as well as unauthorized reading of critical data and user IP from the BootROM and NVM.

The BootROM protection cannot be disabled. The NVM protection can be enabled separately for linear and non-linear mapped ranges by calling the corresponding routines in the BootROM.

When protection of the BootROM or NVM is enabled, instructions from unsafe memory sections (such as RAM) that try to access these areas are blocked.

Further information is available in the BootROM user manual.

12 Interrupt system

12.1 Features

- Up to 16 interrupt nodes for on-chip peripherals
- Up to 8 NMI nodes for critical system events
- Maximum flexibility for all 16 interrupt nodes

12.2 Introduction

Before enabling an interrupt, all corresponding interrupt status flags must be cleared.

12.2.1 Overview

The TLE987x supports 16 interrupt vectors with 16 priority levels. Fifteen of these interrupt vectors are assigned to the on-chip peripherals: GPT12, SSC, CCU6, DMA, bridge driver and A/D converter are each assigned to one dedicated interrupt vector; while UART1 and Timer2, as well as UART2, external interrupt 2 and Timer21 share interrupt vectors. Two vectors are dedicated for external interrupt 0 and 1.

A non-maskable interrupt (NMI) with the highest priority is shared by the following:

- Watchdog Timer, warning before overflow
- MI_CLK Watchdog Timer overflow event
- PLL, loss of lock
- Flash, on operation complete e.g. erase.
- Oscillator watchdog detection for too low oscillation of f_{OSC}
- Flash map error
- Uncorrectable ECC error on Flash and RAM
- VSUP supply prewarning when any supply voltage drops below or exceeds any threshold.
- Overtemperature prewarning when system temperature exceeds a certain limit.

Figure 55, **Figure 58**, **Figure 59**, **Figure 60** and **Figure 60** give a general overview of the interrupt sources and nodes, and their corresponding control and status flags. **Figure 66** gives the corresponding overview for the NMI sources. The table below shows the available interrupt vectors.

Table 67 Interrupt vector table

Service request	Node ID	Description
GPT12	0/1	GPT interrupt (T2-T6, CAPIN)
MU-ADC2/T3	2	Measurement unit, VBG, SDADC[1)], Timer3, BEMF
ADC1	3	ADC1 interrupt / VREF5V overload / VREF5V OV/UV
CCU0	4	CCU6 node 0 interrupt
CCU1	5	CCU6 node 1 interrupt
CCU2	6	CCU6 node 2 interrupt
CCU3	7	CCU6 node 3 interrupt
SSC1	8	SSC1 interrupt (receive, transmit, error)
SSC2	9	SSC2 interrupt (receive, transmit, error)

Table 67 Interrupt vector table (cont'd)

Service request	Node ID	Description
UART1	10	UART1 (ASC-LIN) interrupt (receive, transmit), Timer2, linsync1, LIN
UART2	11	UART2 interrupt (receive, transmit), Timer21, external interrupt (EINT2)
EXINT0	12	External interrupt (EINT0), MON
EXINT1	13	External interrupt (EINT1)
BDRV/CP	14	Bridge driver / charge pump
DMA	15	DMA controller

1) Available only for products with SDADC. See the datasheet for details.

Table 68 NMI interrupt table

Service request	Node	Description
Watchdog timer NMI	NMI	Watchdog timer overflow
PLL NMI	NMI	PLL loss-of-lock
NVM operation complete NMI	NMI	NVM operation complete
Overtemperature NMI	NMI	System overtemperature
Oscillator watchdog NMI	NMI	Oscillator watchdog / MI_CLK watchdog timer overflow
NVM map error NMI	NMI	NVM map error
ECC error NMI	NMI	RAM/NVM uncorrectable ECC error
Supply prewarning NMI	NMI	Supply prewarning

12.3 Functional Description

12.3.1 Interrupt Node Assignment

12.3.1.1 Interrupt Node 0 and 1 - GPT12 Timer Module

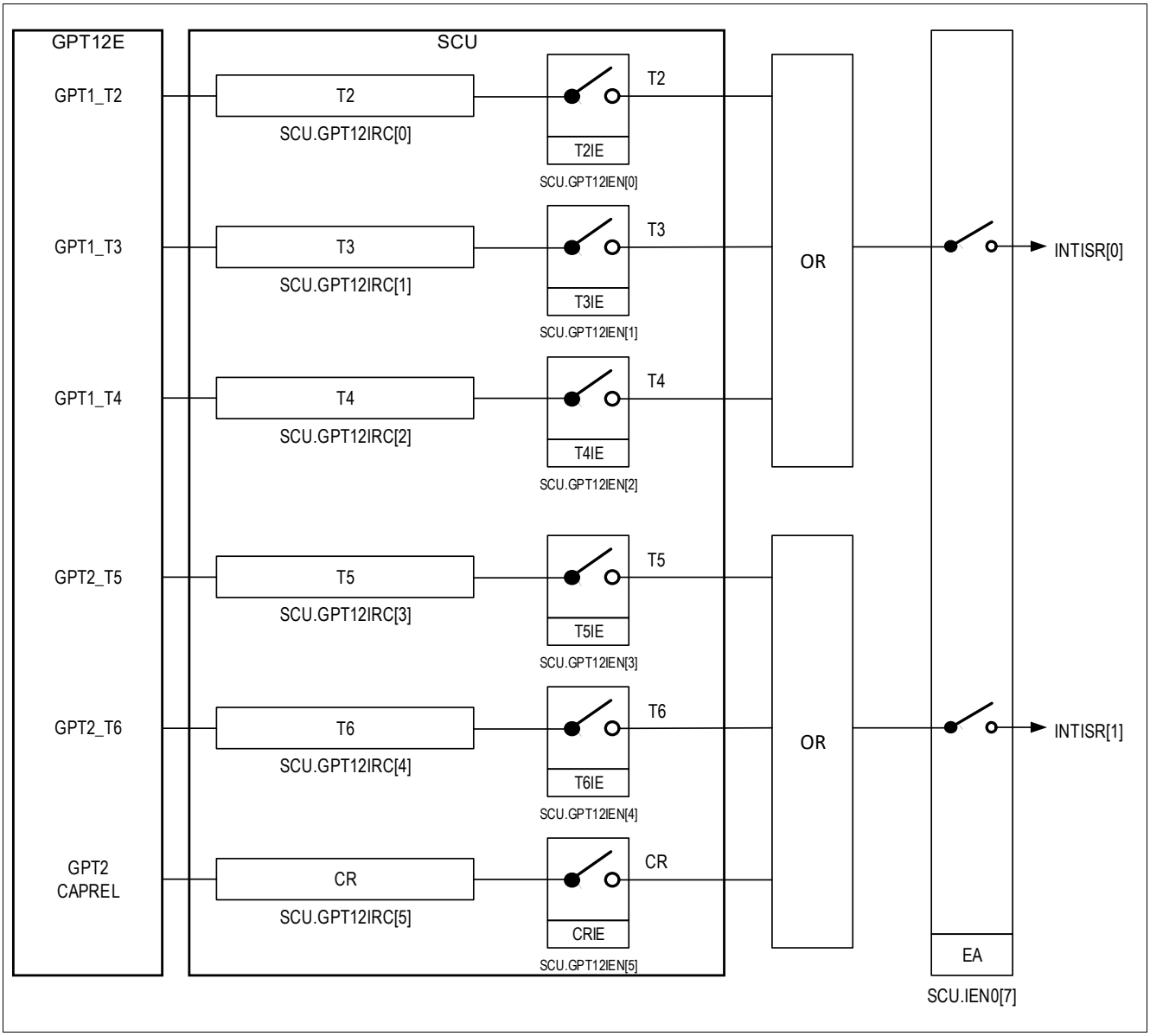


Figure 55 Interrupt Request Sources 0 and 1 (GPT12)

12.3.1.2 Interrupt Node 2 - Measurement Unit

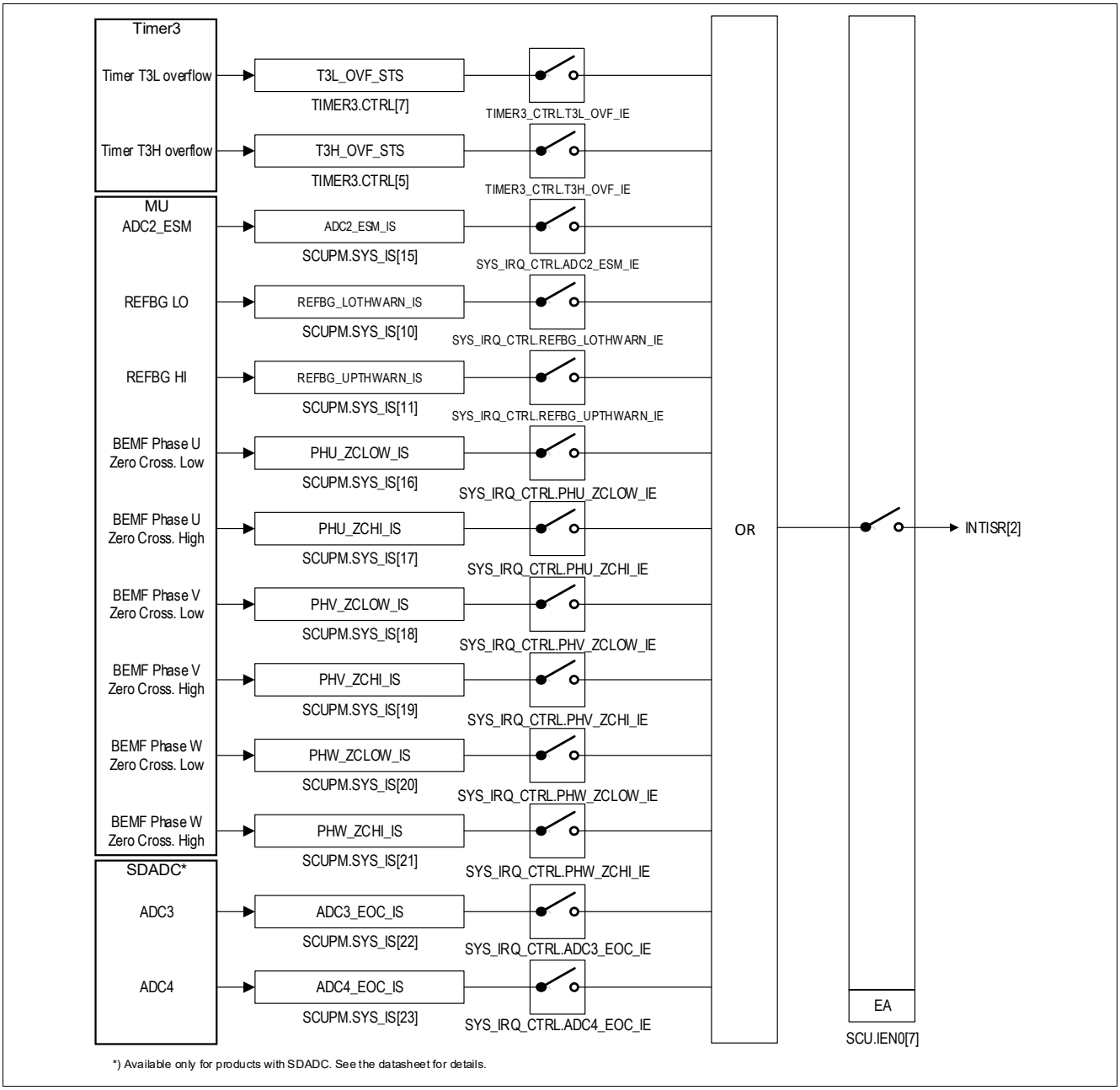


Figure 56 Interrupt Request Sources 2 (MU)

12.3.1.3 Interrupt Node 3 - 10 Bit ADC (ADC1)

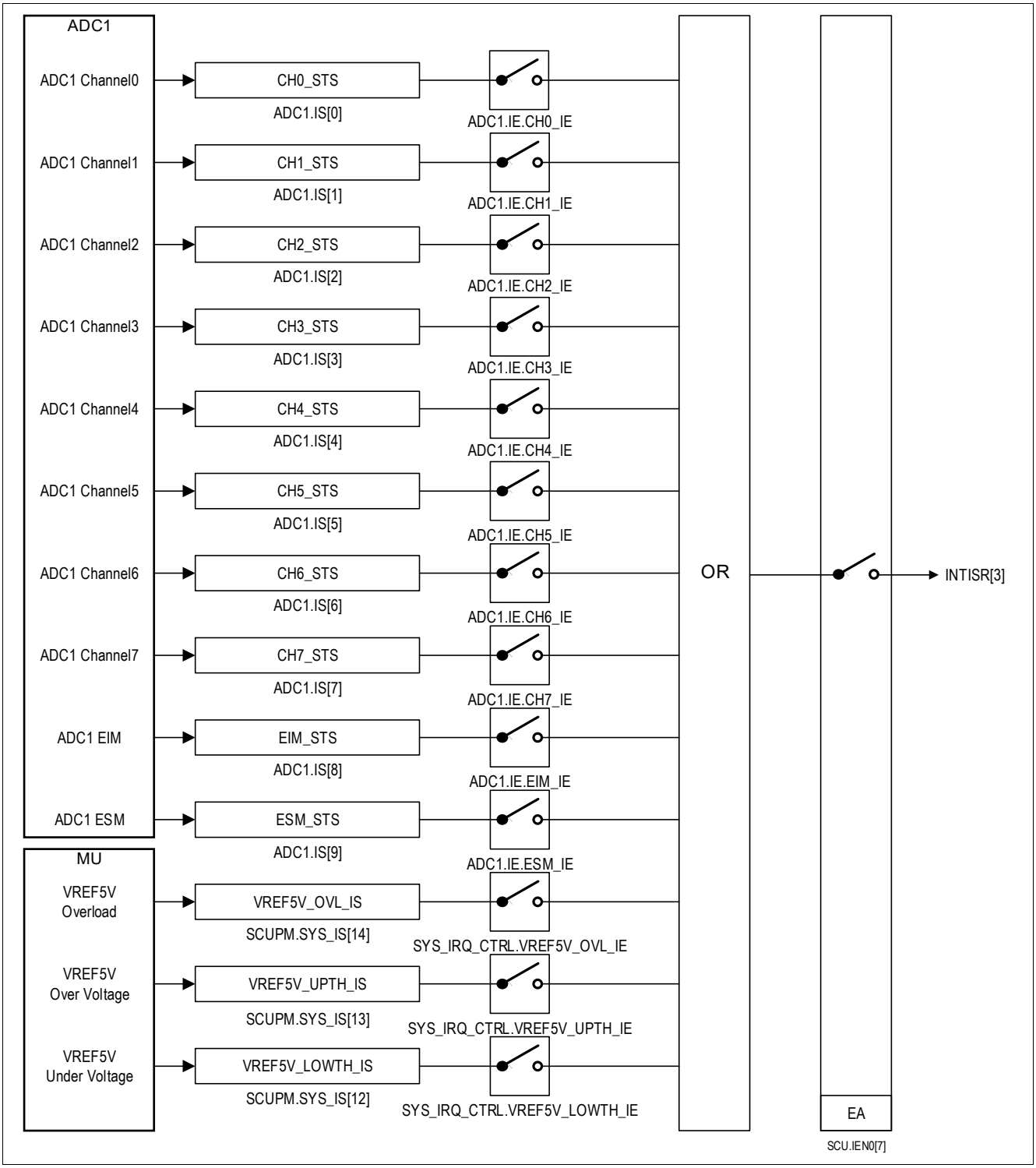


Figure 57 Interrupt Request Sources 3 (ADC1)

12.3.1.4 Interrupt Nodes 4/5/6/7 - Capture Compare Unit (CCU6)

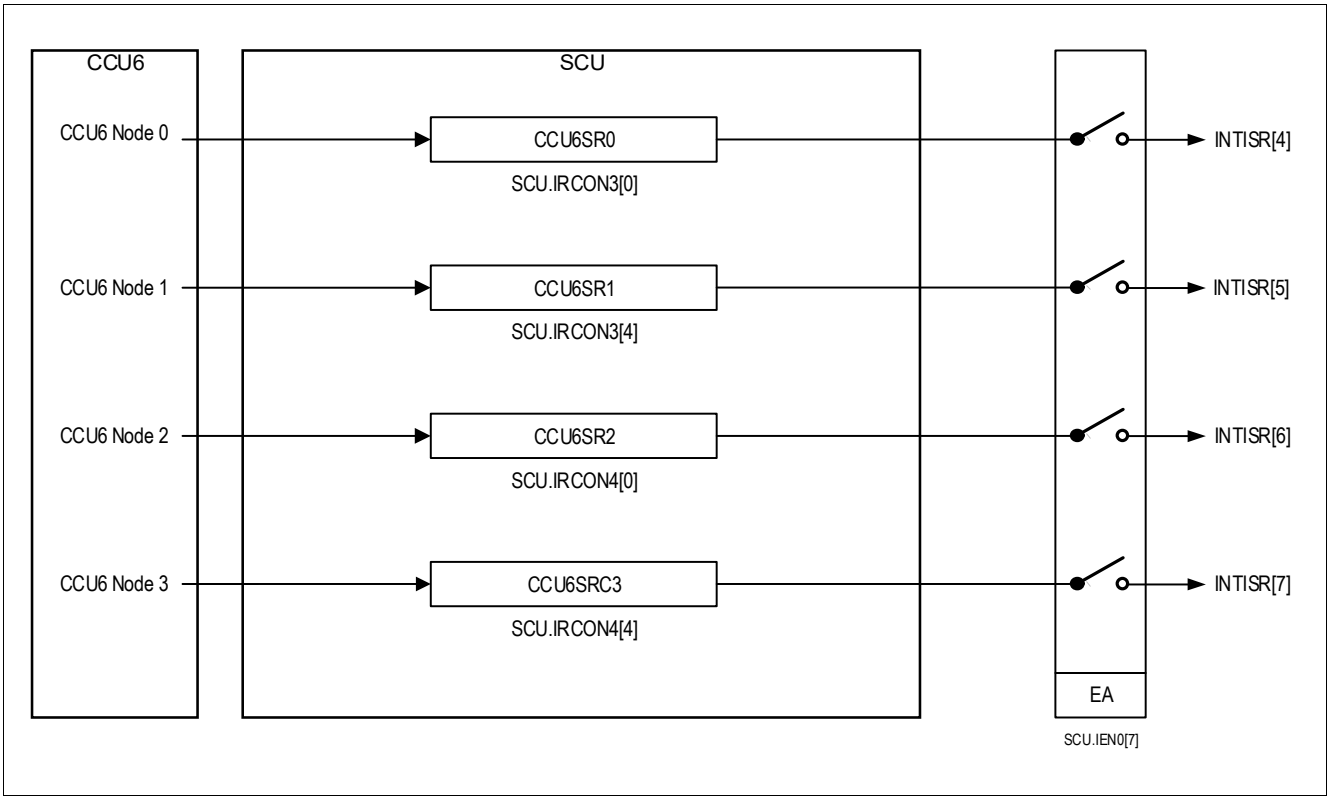


Figure 58 Interrupt Request Sources 4, 5, 6, 7 (CCU6)

12.3.1.5 Interrupt Node 8 and 9 - SSC1/SSC2

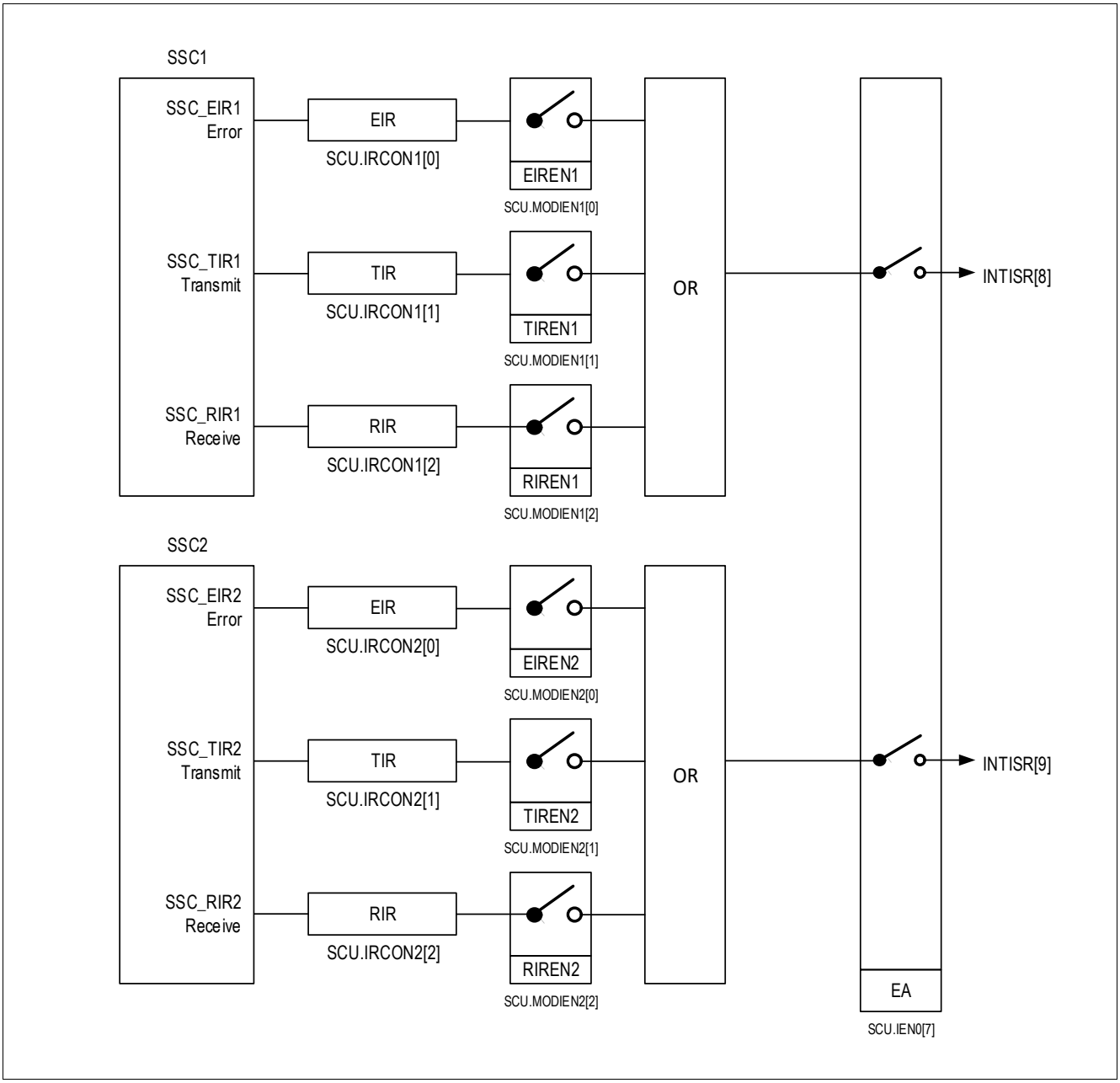


Figure 59 Interrupt Request Sources 8 and 9 (SSC)

12.3.1.6 Interrupt Node 10 - UART1/LIN Transceiver

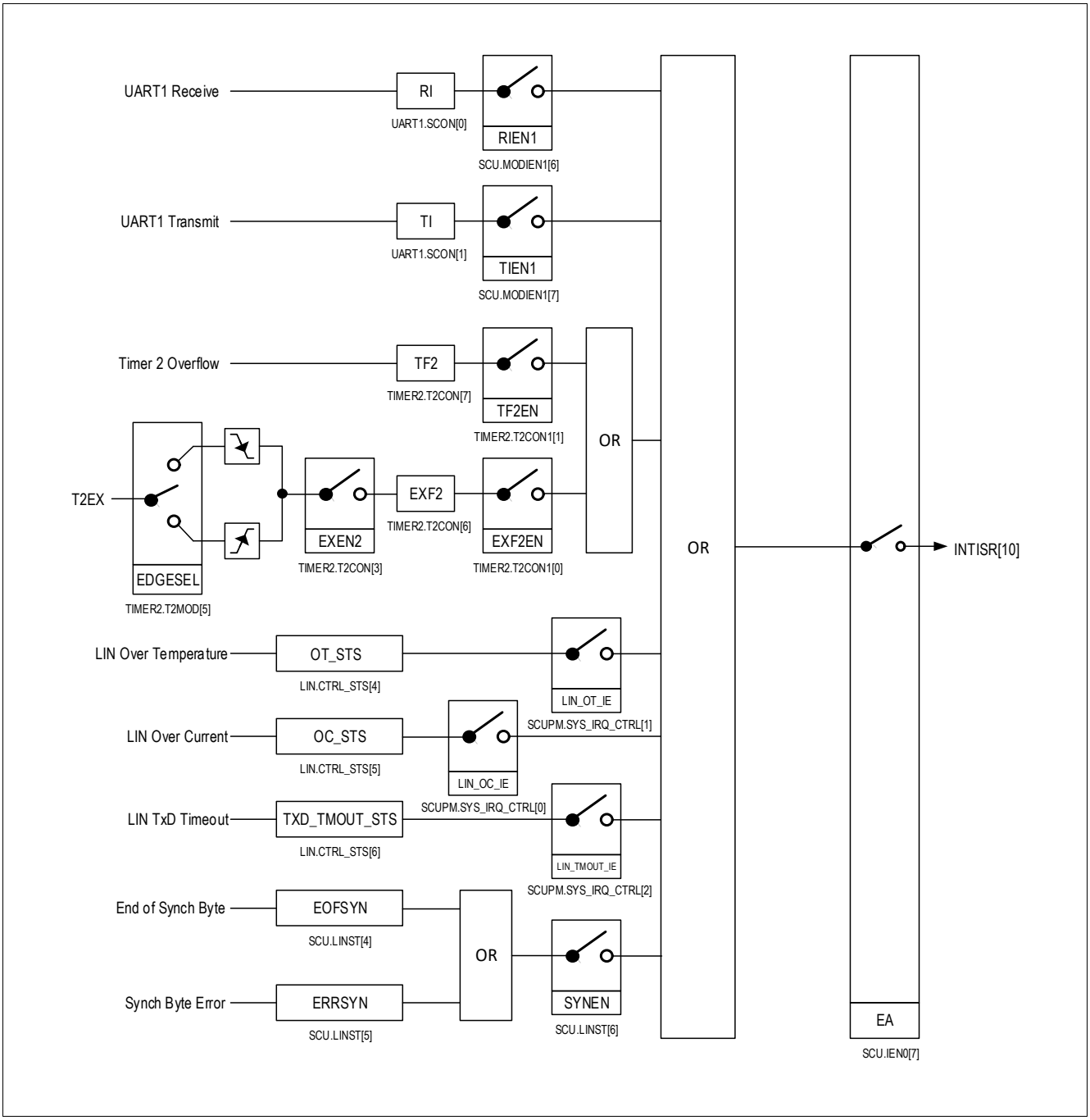


Figure 60 Interrupt Request Source 10 (UART1)

12.3.1.7 Interrupt Node 11 - UART2

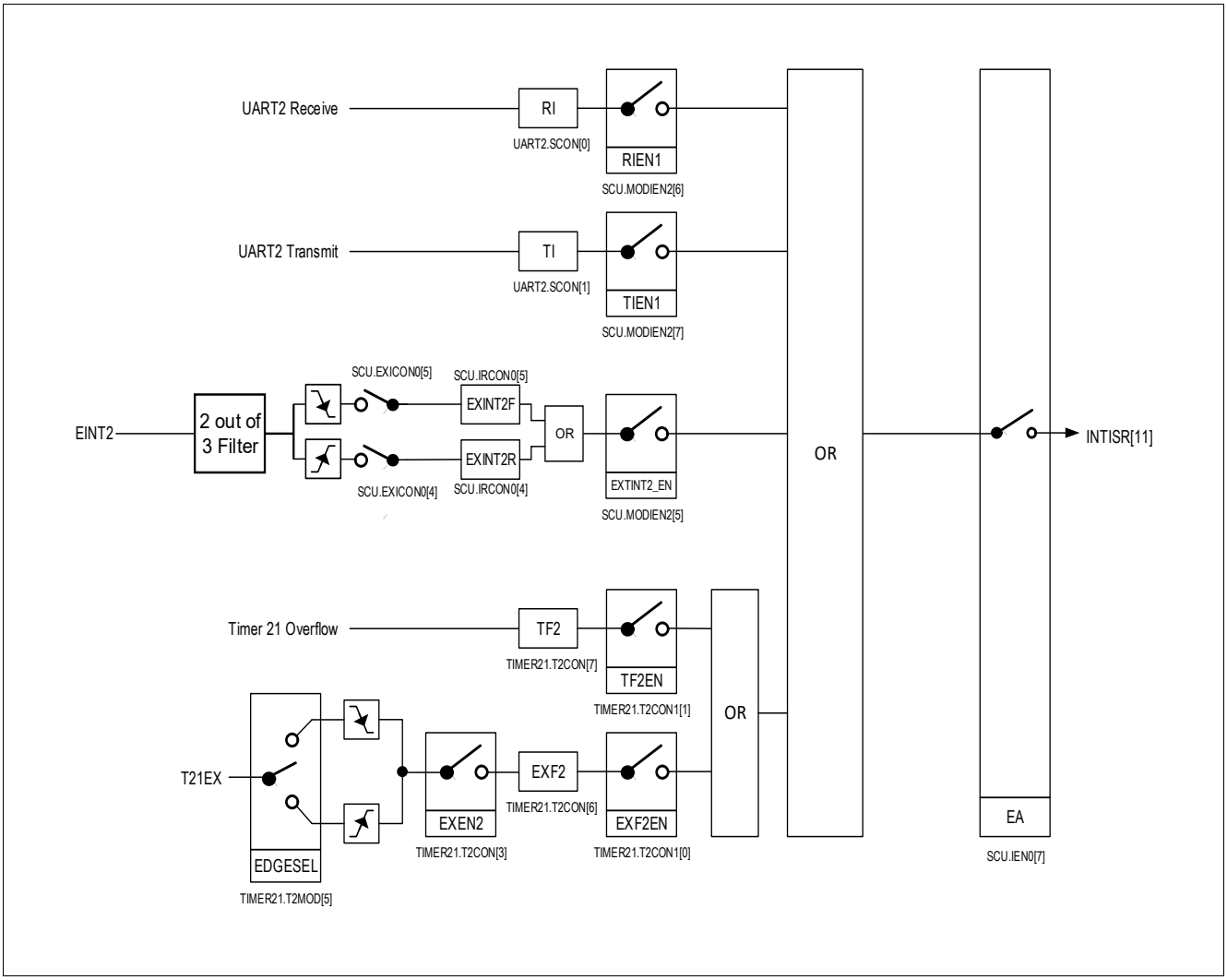


Figure 61 Interrupt Request Source 11 (UART2)

12.3.1.8 Interrupt Node 12 - External Interrupt 0/MON

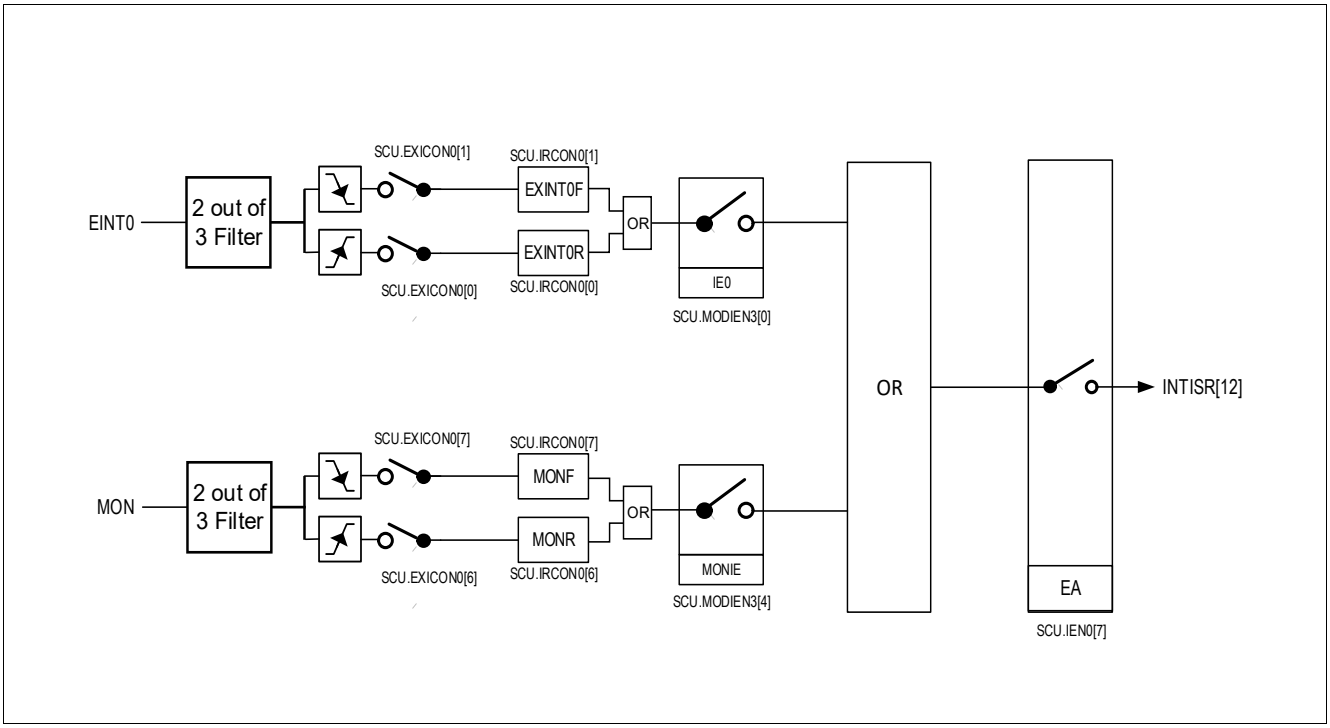


Figure 62 Interrupt Request Sources 12 (Interrupt)

12.3.1.9 Interrupt Node 13 - External Interrupt 1

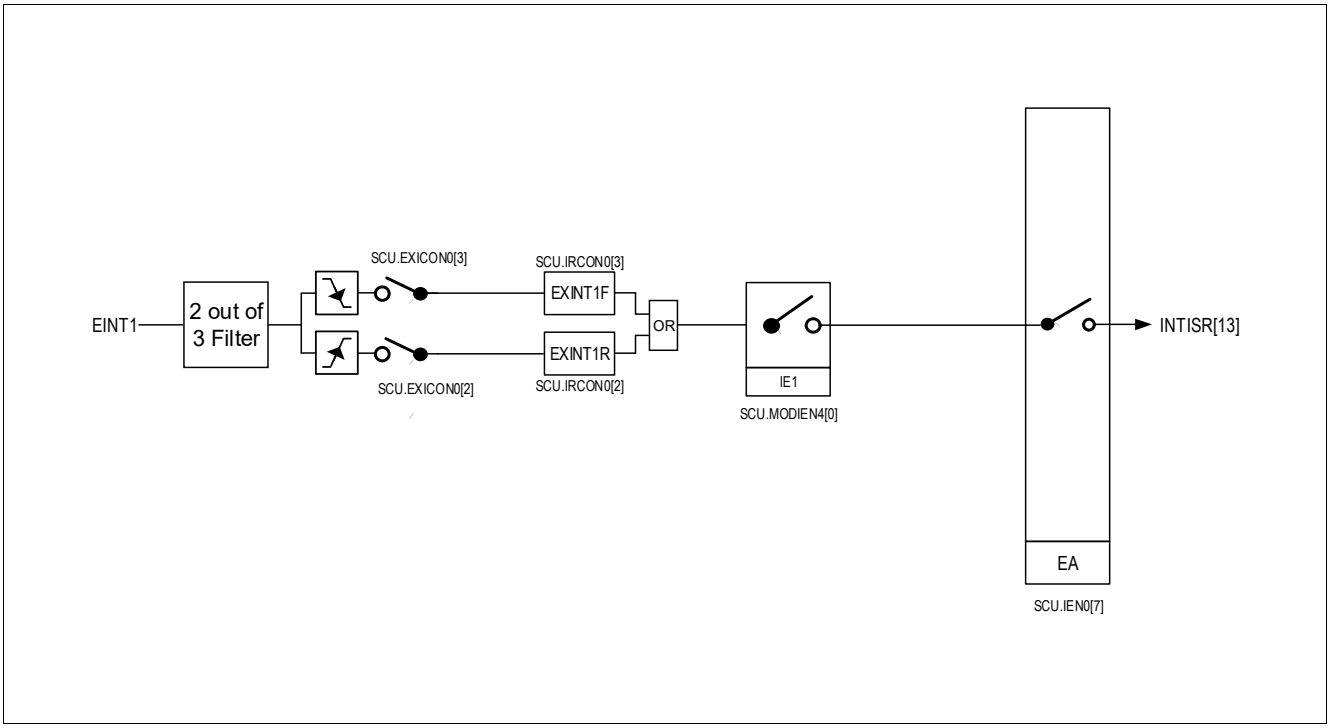


Figure 63 Interrupt Request Sources 13 (Interrupt)

12.3.1.10 Interrupt Node 14 - Bridge Driver

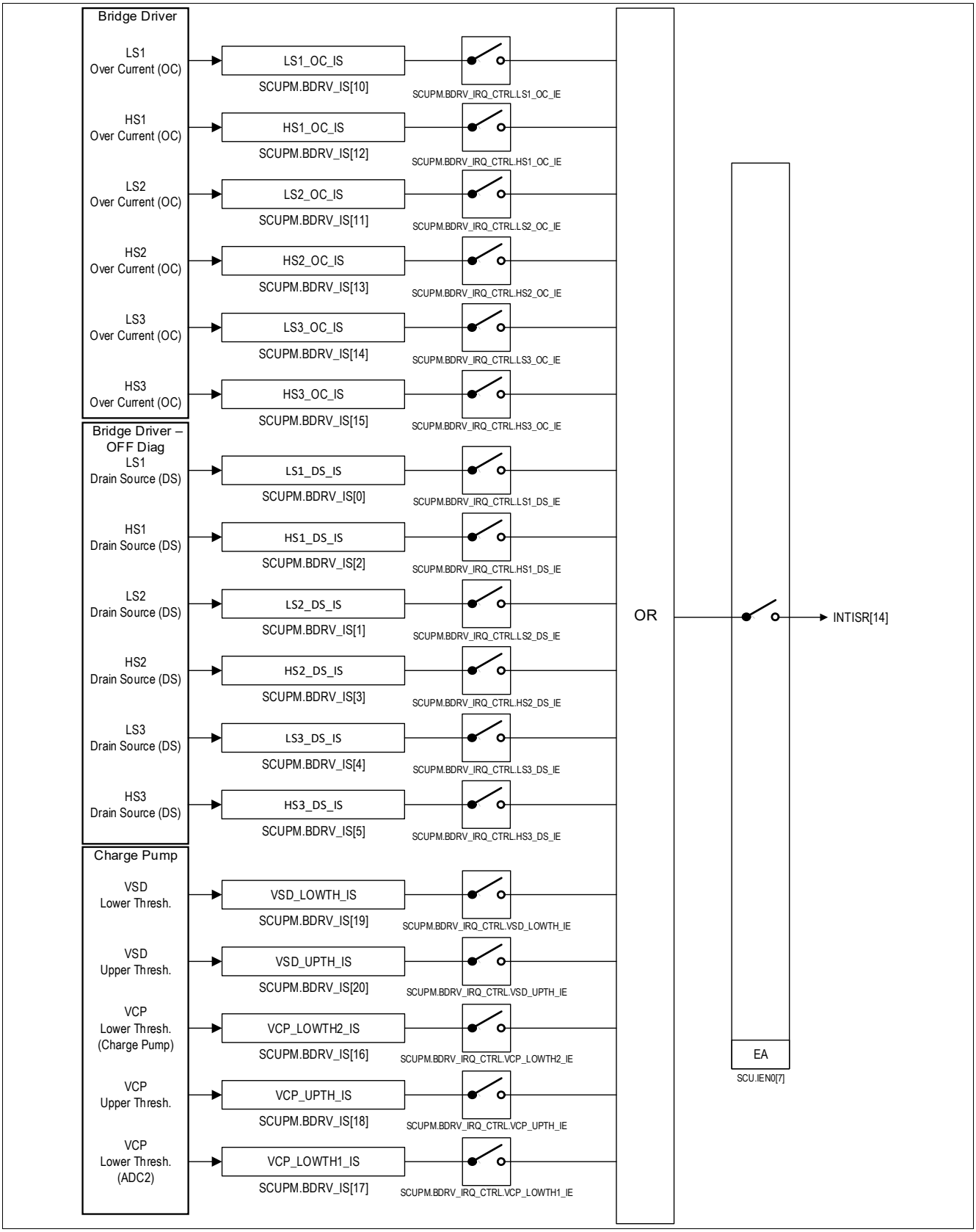


Figure 64 Interrupt Request Sources 14 (Bridge Driver)

12.3.1.11 Interrupt Node 15 - DMA Controller

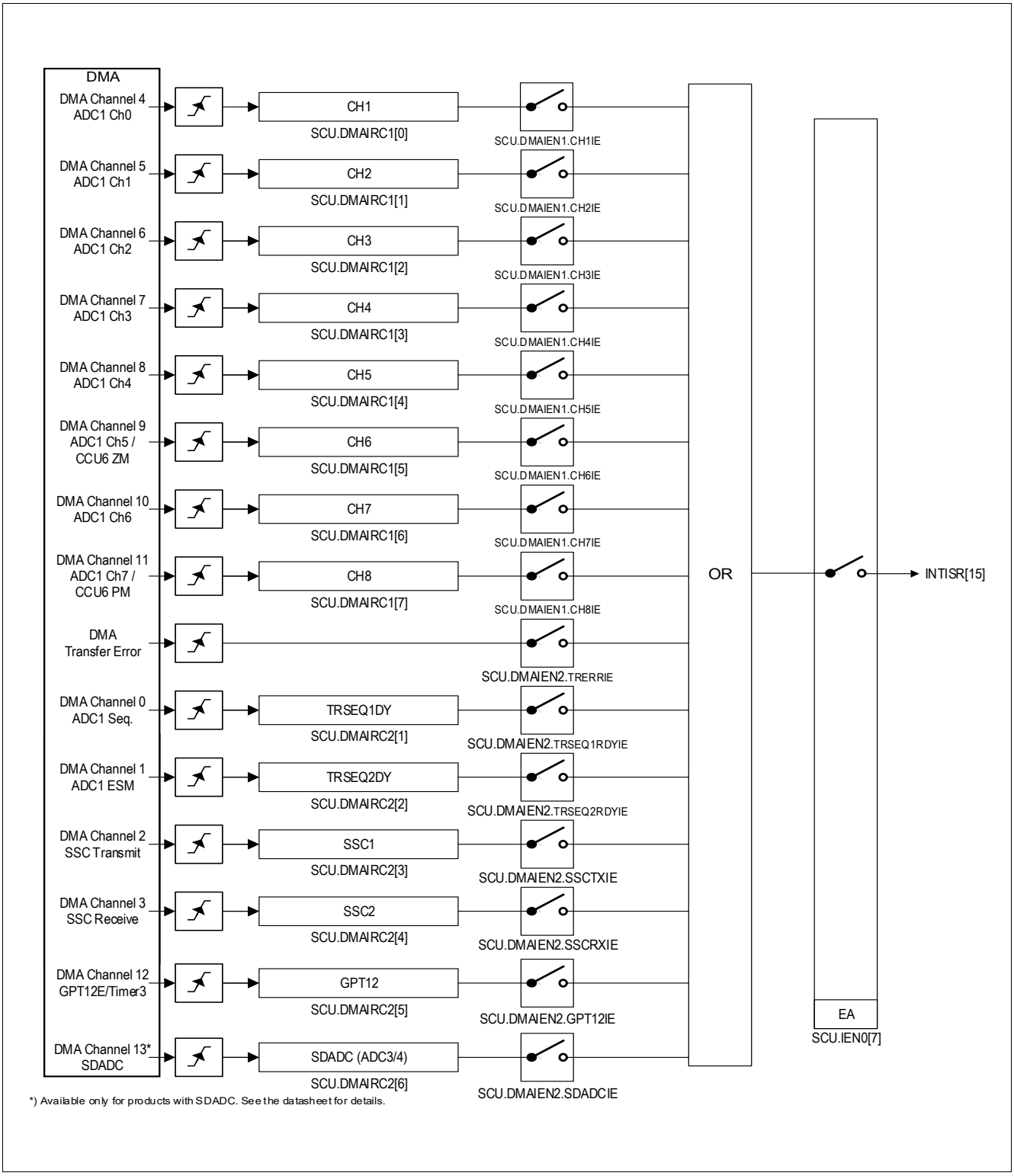


Figure 65 Interrupt Request Sources 15 (DMA)

12.3.1.12 Non-Maskable Interrupt (NMI)

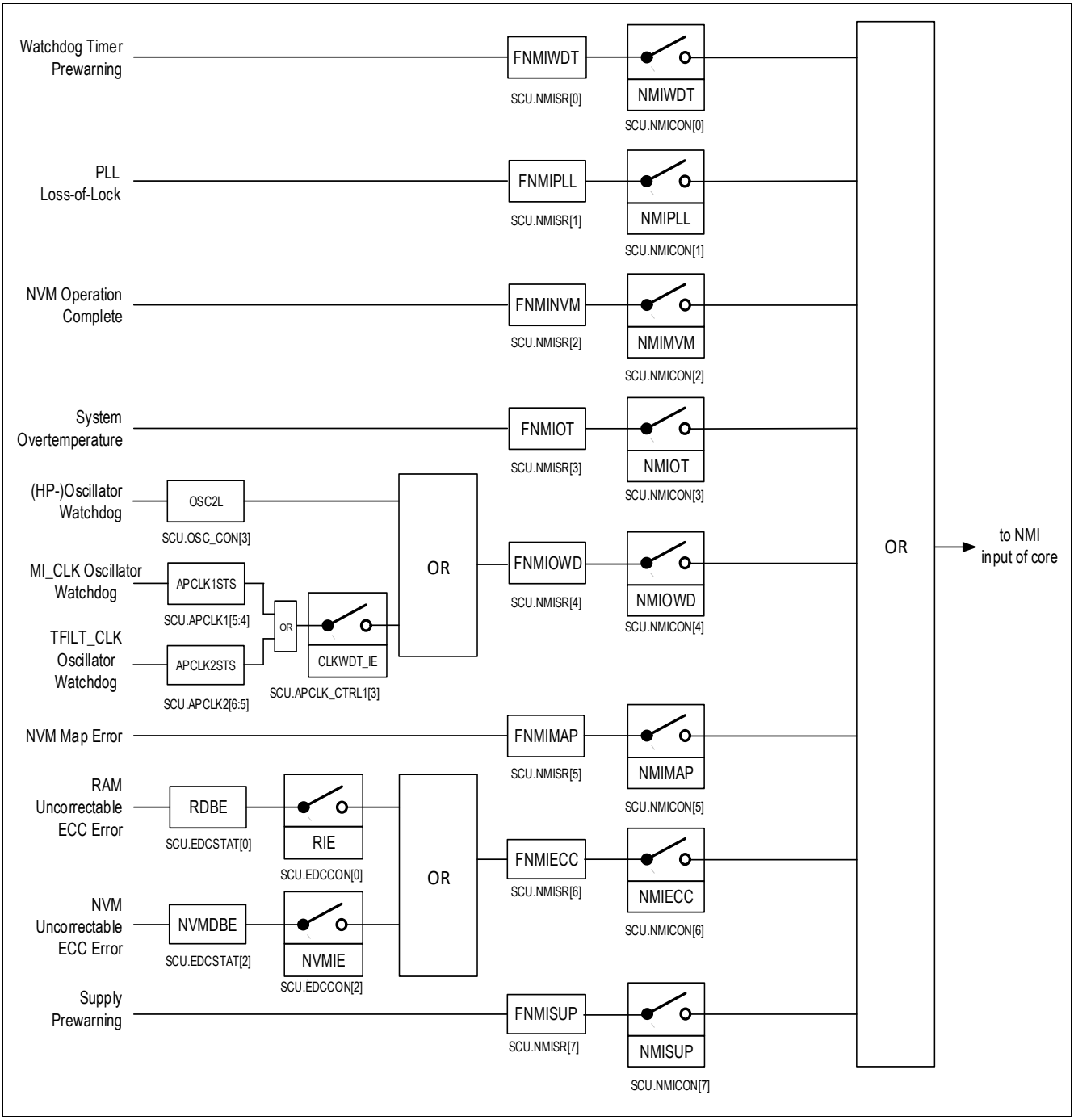


Figure 66 Non-Maskable Interrupt Request Source

12.3.1.13 Interrupt Flags Overview

Table 69 All Interrupt Flags and Enable

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
Non-Maskable Interrupts					
OT NMI					
SYS_OTWARN	NMI	edge	set until cleared by software	SCUPM.**SYS_IS**.SYS_OTWARN_IS	SCUPM.**SYS_IRQ_CTRL**.SYS_OTWARN_IE
SYS_OT	NMI	edge	set until cleared by software	SCUPM.**SYS_IS**.SYS_OT_IS	SCUPM.**SYS_IRQ_CTRL**.SYS_OT_IE
PMU_OTWARN	NMI	edge	set until cleared by software	SCUPM.**SYS_IS**.PMU_OTWARN_IS	SCUPM.**SYS_IRQ_CTRL**.PMU_OTWARN_IE
PMU_OT	NMI	edge	set until cleared by software	SCUPM.**SYS_IS**.PMU_OT_IS	SCUPM.**SYS_IRQ_CTRL**.PMU_OT_IE
Supply NMI					
PREWARN_SUPMON_UV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.MON_UV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.MON_UV_IE
PREWARN_SUPVS_UV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.VS_UV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.VS_UV_IE
PREWARN_SUPVDD5V_UV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.VDD5V_UV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.VDD5V_UV_IE
PREWARN_SUPVDD1V5_UV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.VDD1V5_UV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.VDD1V5_UV_IE
PREWARN_SUPMON_OV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.MON_OV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.MON_OV_IE
PREWARN_SUPVS_OV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.VS_OV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.VS_OV_IE
PREWARN_SUPVDD5V_OV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.VDD5V_OV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.VDD5V_OV_IE
PREWARN_SUPVDD1V5_OV	NMI	edge	set until cleared by software	SCUPM.**SYS_SUPPLY_IRQ_STS**.VDD1V5_OV_IS	SCUPM.**SYS_SUPPLY_IRQ_CTRL**.VDD1V5_OV_IE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
VDDP_OVERVOLT	NMI	level	set until cleared by software	PMU.**PMU_SUPPLY_STS**.PMU_5V_OVERVOLT	PMU.**PMU_SUPPLY_STS**.PMU_5V_FAIL_EN
VDDP_OVERLOAD	NMI	level	set until cleared by software	PMU.**PMU_SUPPLY_STS**.PMU_5V_OVERLOAD	PMU.**PMU_SUPPLY_STS**.PMU_5V_FAIL_EN
VDDC_OVERVOLT	NMI	level	set until cleared by software	PMU.**PMU_SUPPLY_STS**.PMU_1V5_OVERVOLT	PMU.**PMU_SUPPLY_STS**.PMU_1V5_FAIL_EN
VDDC_OVERLOAD	NMI	level	set until cleared by software	PMU.**PMU_SUPPLY_STS**.PMU_1V5_OVERLOAD	PMU.**PMU_SUPPLY_STS**.PMU_1V5_FAIL_EN
VDDEXT_OVERVOLT	NMI	level	set until cleared by software	PMU.**VDDEXT_CTRL**.OVERVOLT	PMU.**VDDEXT_CTRL**.FAIL_EN
VDDEXT_OVERLOAD	NMI	level	set until cleared by software	PMU.**VDDEXT_CTRL**.OVERLOAD	PMU.**VDDEXT_CTRL**.FAIL_EN
VDDEXT_SHORT	NMI	level	set until cleared by software	PMU.**VDDEXT_CTRL**.SHORT	PMU.**VDDEXT_CTRL**.FAIL_EN
MI_CLK CLKWDT	NMI	level		SCU.**NMISR**.FNMIOWD SCU.**APCLK1**.APCLK1STS	SCU.**NMICON**.NMIOWD SCU.**APCLK_CTRL1**.CLKWDT_IE
TFILT_CLK CLKWDT	NMI	level		SCU.**NMISR**.FNMIOWD SCU.**APCLK2**.APCLK2STS	SCU.**NMICON**.NMIOWD SCU.**APCLK_CTRL1**.CLKWDT_IE
(HP)Oscillator CLKWDT	NMI	level		SCU.**OSC_CON**.OSC2L	SCU.**NMICON**.NMIOWD
ECC Error NMI					
RAM Uncorrectable	NMI	level	set until cleared by software	SCU.**EDCSTAT**.RDBE	SCU.**EDCCON**.RIE
NVM Uncorrectable	NMI	level	set until cleared by software	SCU.**EDCSTAT**.NVMDBE	SCU.**EDCCON**.NVMIE
Supply Prewarning	NMI	level	set until cleared by software	SCU.**NMISR**.FNMISUP	SCU.**NMICON**.NMISUP
INTISR<0/1> → GPT12					
GPT12-T2	0	level	2 per_clk cycles	SCU.**GPT12IRC**.T2	SCU.**GPT12IEN**.T2IE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
GPT12E T3	0	level	2 per_clk cycles	SCU.**GPT12IRC**.T3	SCU.**GPT12IEN**.T3IE
GPT12E T4	0	level	2 per_clk cycles	SCU.**GPT12IRC**.T4	SCU.**GPT12IEN**.T4IE
GPT12E T5	1	level	2 per_clk cycles	SCU.**GPT12IRC**.T5	SCU.**GPT12IEN**.T5IE
GPT12E T6	1	level	2 per_clk cycles	SCU.**GPT12IRC**.T6	SCU.**GPT12IEN**.T6IE
GPT12E CAPREL	0/1	level	2 per_clk cycles	SCU.**GPT12IRC**.CR	SCU.**GPT12IEN**.CRIE
INTISR<2> → MU					
ADC2 ESM	2	level	set until cleared by software	SCUPM.**SYS_IS**.ADC2_ESM_IS	SCUPM.**SYS_IRQ_CTRL**.ADC2_ESM_IE
T3L overflow	2	level	set until cleared by software	TIMER3.**CTRL**.T3L_OVF_STS	TIMER3.**CTRL**.T3L_OVF_IE
T3H overflow	2	level	set until cleared by software	TIMER3.**CTRL**.T3L_OVF_STS	TIMER3.**CTRL**.T3H_OVF_IE
REF_BG_LO	2	level	set until cleared by software	SCUPM.**SYS_IS**.REFBG_LOTHWARN_IS MF.**REF1_STS**.REFBG_LOTHWARN_STS	SCUPM.**SYS_IRQ_CTRL**.REFBG_LOTHWARN_IE
REF_BG_HI	2	level	set until cleared by software	SCUPM.**SYS_IS**.REFBG_UPTHWARN_IS MF.**REF1_STS**.REFBG_UPTHWARN_STS	SCUPM.**SYS_IRQ_CTRL**.REFBG_UPTHWARN_IE
PHU_ZCLOW	2	level	set until cleared by software	SCUPM.**SYS_IS**.PHU_ZCLOW_IS	SCUPM.**SYS_IRQ_CTRL**.PHU_ZCLOW_IE
PHU_ZCHI	2	level	set until cleared by software	SCUPM.**SYS_IS**.PHU_ZCHI_IS	SCUPM.**SYS_IRQ_CTRL**.PHU_ZCHI_IE
PHV_ZCLOW	2	level	set until cleared by software	SCUPM.**SYS_IS**.PHV_ZCLOW_IS	SCUPM.**SYS_IRQ_CTRL**.PHV_ZCLOW_IE
PHV_ZCHI	2	level	set until cleared by software	SCUPM.**SYS_IS**.PHV_ZCHI_IS	SCUPM.**SYS_IRQ_CTRL**.PHV_ZCHI_IE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
PHW_ZCLOW	2	level	set until cleared by software	SCUPM.**SYS_IS**.PHW_ZCLOW_IS	SCUPM.**SYS_IRQ_CTRL**.PHW_ZCLOW_IE
PHW_ZCHI	2	level	set until cleared by software	SCUPM.**SYS_IS**.PHW_ZCHI_IS	SCUPM.**SYS_IRQ_CTRL**.PHW_ZCHI_IE
ADC4 (SDADC)[1)]	2	level	set until cleared by software	SCUPM.**SYS_IS**.ADC4_EOC_IS	SCUPM.**SYS_IRQ_CTRL**.ADC4_EOC_IE
ADC3 (SDADC)[1)]	2	level	set until cleared by software	SCUPM.**SYS_IS**.ADC3_EOC_IS	SCUPM.**SYS_IRQ_CTRL**.ADC3_EOC_IE
INTISR<3> → ADC 10 Bit					
ADC1 CH0	3	level	set until cleared by software	ADC1.**IS**.CH0_STS	ADC1.**IE**.CH0_IE
ADC1 CH1	3	level	set until cleared by software	ADC1.**IS**.CH1_STS	ADC1.**IE**.CH1_IE
ADC1 CH2	3	level	set until cleared by software	ADC1.**IS**.CH2_STS	ADC1.**IE**.CH2_IE
ADC1 CH3	3	level	set until cleared by software	ADC1.**IS**.CH3_STS	ADC1.**IE**.CH3_IE
ADC1 CH4	3	level	set until cleared by software	ADC1.**IS**.CH4_STS	ADC1.**IE**.CH4_IE
ADC1 CH5	3	level	set until cleared by software	ADC1.**IS**.CH5_STS	ADC1.**IE**.CH5_IE
ADC1 CH6	3	level	set until cleared by software	ADC1.**IS**.CH6_STS	ADC1.**IE**.CH6_IE
ADC1 CH7	3	level	set until cleared by software	ADC1.**IS**.CH7_STS	ADC1.**IE**.CH7_IE
ADC1 ESM	3	level	set until cleared by software	ADC1.**IS**.ESM_STS	ADC1.**IE**.ESM_IE
ADC1 EIM	3	level	set until cleared by software	ADC1.**IS**.EIM_STS	ADC1.**IE**.EIM_IE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
VREF5V Overflow	3	level	set until cleared by software	SCUPM.**SYS_IS**.VREF5V_OVL_IS	SCUPM.**SYS_IRQ_CTRL**.VREF5V_OVL_IE
VREF5V_UPTH (VAREF)	3	level	set until cleared by software	SCUPM.**SYS_IS**.VREF5V_UPTH_IS	SCUPM.**SYS_IRQ_CTRL**.VREF5V_UPTH_IE
VREF5V_LOTH (VAREF)	3	level	set until cleared by software	SCUPM.**SYS_IS**.VREF5V_LOWTH_IS	SCUPM.**SYS_IRQ_CTRL**.VREF5V_LOWTH_IE
INTISR<4,5,6,7> → CCU6					
CCU0[2)] Node 0 (SR0)	4	level	2 per_clk cycles	SCU.**IRCON3**.CCU6SR0	CPU.**NVIC_ISER0**.Int_CCU6SR0
CCU1[2)] Node 1 (SR1)	5	level	2 per_clk cycles	SCU.**IRCON3**.CCU6SR1	CPU.**NVIC_ISER0**.Int_CCU6SR1
CCU2[2)] Node 2 (SR2)	6	level	2 per_clk cycles	SCU.**IRCON4**.CCU6SR2	CPU.**NVIC_ISER0**.Int_CCU6SR2
CCU3[2)] Node 3 (SR3)	7	level	2 per_clk cycles	SCU.**IRCON4**.CCU6SR3	CPU.**NVIC_ISER0**.Int_CCU6SR3
INTISR<8,9> → SSC1/SSC2					
SSC1 Error	8	level	2 per_clk cycles	SCU.**IRCON1**.EIR	SCU.**MODIEN1**.EIREN
SSC1 Transmit	8	level	2 per_clk cycles	SCU.**IRCON1**.TIR	SCU.**MODIEN1**.TIREN
SSC1 Receive	8	level	2 per_clk cycles	SCU.**IRCON1**.RIR	SCU.**MODIEN1**.RIREN
SSC2 Error	9	level	2 per_clk cycles	SCU.**IRCON2**.EIR	SCU.**MODIEN2**.EIREN
SSC2 Transmit	9	level	2 per_clk cycles	SCU.**IRCON2**.TIR	SCU.**MODIEN2**.TIREN
SSC2 Receive	9	level	2 per_clk cycles	SCU.**IRCON2**.RIR	SCU.**MODIEN2**.RIREN
INTISR<10,11> → UART1/UART2					
UART1 Receive	10	level	copy of RI bit, set until cleared by software	UART1.**SCON**.RI	SCU.**MODIEN1**.RIEN
UART1 Transmit	10	level	copy of TI bit, set until cleared by software	UART1.**SCON**.TI	SCU.**MODIEN1**.TIEN

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
UART2 Receive	11	level	copy of RI bit, set until cleared by software	UART2.**SCON**.RI	SCU.**MODIEN2**.RIEN
UART2 Transmit	11	level	copy of TI bit, set until cleared by software	UART2.**SCON**.TI	SCU.**MODIEN2**.TIEN
LIN sync byte error	10	level		SCU.**LINST**.ERRSYN	SCU.**LINST**.SYNEN
LIN end of sync byte	10	level		SCU.**LINST**.EOFSYN	SCU.**LINST**.SYNEN
Timer2 Overflow	10	level		TIMER2.**T2CON**.TF2	TIMER2.**T2CON1**.TF2EN
Timer2 T2EX	10			TIMER2.**T2CON**.EXF2	TIMER2.**T2CON1**.EXF2EN
LIN OT	10	level		LIN.**CTRL_STS**.OT_STS	SCUPM.**SYS_IRQ_CTRL**.LIN_OT_IE
LIN OC	10	level		LIN.**CTRL_STS**.OC_STS	SCUPM.**SYS_IRQ_CTRL**.LIN_OC_IE
TxD_TMOUT	10			LIN.**CTRL_STS**.TXD_TMOUT_STS	SCUPM.**SYS_IRQ_CTRL**.LIN_TMOUT_IE
Timer21 Overflow	11	level		TIMER21.**T2CON**.TF2	TIMER21.**T2CON1**.TF2EN
T21EX	11			TIMER21.**T2CON**.EXF2	TIMER21.**T2CON1**.EXF2EN
EINT2	11			SCU.**IRCON0**.EXINT2R/F	SCU.**MODIEN2**.EXINT2_EN
INTISR<12,13> → EXTINT1/EXTINT2					
exint0	12	level		SCU.**IRCON0**.EXINT0R/F	SCU.**MODIEN3**.IE0
MON	12	level		SCU.**IRCON0**.MONR/F	SCU.**MODIEN3**.MONIE
exint1	13	level		SCU.**IRCON0**.EXINT1R/F	SCU.**MODIEN4**.IE1
INTISR<14> → Bridge Driver					
BDRV/CP VSD Lower Threshold	14	level	set until cleared by software	SCUPM.**BDRV_IS**.VSD_LOWTH_IS	SCUPM.**BDRV_IRQ_CTRL**.VSD_LOWTH_IE
BDRV/CP VSD Upper Threshold	14	level	set until cleared by software	SCUPM.**BDRV_IS**.VSD_UPTH_IS	SCUPM.**BDRV_IRQ_CTRL**.VSD_UPTH_IE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
BDRV/CP VCP Lower Threshold1 (ADC2.Ch3)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.VCP_LOWTH1_IS	SCUPM.**BDRV_IRQ_CTRL**.VCP_LOWTH1_IE
BDRV/CP VCP Lower Threshold2	14	level	set until cleared by software	SCUPM.**BDRV_IS**.VCP_LOWTH2_IS	SCUPM.**BDRV_IRQ_CTRL**.VCP_LOWTH2_IE
BDRV/CP VCP Upper Threshold	14	level	set until cleared by software	SCUPM.**BDRV_IS**.VCP_UPTH_IS	SCUPM.**BDRV_IRQ_CTRL**.VCP_UPTH_IE
BDRV/CP LS1 Over Current (OC)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.LS1_OC_IS	SCUPM.**BDRV_IRQ_CTRL**.LS1_OC_IE
BDRV/CP HS1 Over Current (OC)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.HS1_OC_IS	SCUPM.**BDRV_IRQ_CTRL**.HS1_OC_IE
BDRV/CP LS2 Over Current (OC)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.LS2_OC_IS	SCUPM.**BDRV_IRQ_CTRL**.LS2_OC_IE
BDRV/CP HS2 Over Current (OC)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.HS2_OC_IS	SCUPM.**BDRV_IRQ_CTRL**.HS2_OC_IE
BDRV/CP LS3 Over Current (OC)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.LS3_OC_IS	SCUPM.**BDRV_IRQ_CTRL**.LS3_OC_IE
BDRV/CP HS3 Over Current (OC)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.HS3_OC_IS	SCUPM.**BDRV_IRQ_CTRL**.HS3_OC_IE
BDRV/CP LS1 Drain-Source Monitoring (DS)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.LS1_DS_IS BDRV.**CTRL1**.LS1_DS_STS	SCUPM.**BDRV_IRQ_CTRL**.LS1_DS_IE
BDRV/CP HS1 Drain-Source Monitoring (DS)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.HS1_DS_IS BDRV.**CTRL1**.HS1_DS_STS	SCUPM.**BDRV_IRQ_CTRL**.HS1_DS_IE
BDRV/CP LS2 Drain-Source Monitoring (DS)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.LS2_DS_IS BDRV.**CTRL1**.LS2_DS_STS	SCUPM.**BDRV_IRQ_CTRL**.LS2_DS_IE
BDRV/CP HS2 Drain-Source Monitoring (DS)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.HS2_DS_IS BDRV.**CTRL1**.HS2_DS_STS	SCUPM.**BDRV_IRQ_CTRL**.HS2_DS_IE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
BDRV/CP LS3 Drain-Source Monitoring (DS)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.LS3_DS_IS BDRV.**CTRL2**.LS3_DS_STS	SCUPM.**BDRV_IRQ_CTRL**.LS3_DS_IE
BDRV/CP HS3 Drain-Source Monitoring (DS)	14	level	set until cleared by software	SCUPM.**BDRV_IS**.HS3_DS_IS BDRV.**CTRL2**.HS3_DS_STS	SCUPM.**BDRV_IRQ_CTRL**.HS3_DS_IE
INTISR<15> → DMA Controller					
DMA Channel 4 Src: ADC1 Ch0	15	level	set until cleared by software	SCU.**DMAIRC1**.CH1	SCU.**DMAIEN1**.CH1IE
DMA Channel 5 Src: ADC1 Ch1	15	level	set until cleared by software	SCU.**DMAIRC1**.CH2	SCU.**DMAIEN1**.CH2IE
DMA Channel 6 Src: ADC1 Ch2	15	level	set until cleared by software	SCU.**DMAIRC1**.CH3	SCU.**DMAIEN1**.CH3IE
DMA Channel 7 Src: ADC1 Ch3	15	level	set until cleared by software	SCU.**DMAIRC1**.CH4	SCU.**DMAIEN1**.CH4IE
DMA Channel 8 Src: ADC1 Ch4	15	level	set until cleared by software	SCU.**DMAIRC1**.CH5	SCU.**DMAIEN1**.CH5IE
DMA Channel 9 Src: ADC1 Ch5 or CCU6 ZM	15	level	set until cleared by software	SCU.**DMAIRC1**.CH6	SCU.**DMAIEN1**.CH6IE
DMA Channel 10 Src: ADC1 Ch6	15	level	set until cleared by software	SCU.**DMAIRC1**.CH7	SCU.**DMAIEN1**.CH7IE
DMA Channel 11 Src: ADC1 Ch7 or CCU6 PM	15	level	set until cleared by software	SCU.**DMAIRC1**.CH8	SCU.**DMAIEN1**.CH8IE
DMA Channel 12 Src: GPT12E, or TIMER3	15	level	set until cleared by software	SCU.**DMAIRC2**.GPT12	SCU.**DMAIEN2**.GPT12IE
DMA Channel 13 Src: ADC3 or ADC4 (SDADC)[1)]	15	level	set until cleared by software	SCU.**DMAIRC2**.SDADC	SCU.**DMAIEN2**.SDADCIE
DMA Channel 0 Src: ADC1 Sequence	15	level	set until cleared by software	SCU.**DMAIRC2**.TRSEQ1DY	SCU.**DMAIEN2**.TRSEQ1RDYIE

Table 69 All Interrupt Flags and Enable (cont'd)

Service Request	Node ID	Level/Edge Sensitive	Duration	SFR Flag	Interrupt Enable
DMA Channel 1 Src: ADC1 ESM	15	level	set until cleared by software	SCU.**DMAIRC2**.TRSEQ2DY	SCU.**DMAIEN2**.TRSEQ2RDYIE
DMA Channel 2 Src: SSC Transmit	15	level	set until cleared by software	SCU.**DMAIRC2**.SSC1RDY	SCU.**DMAIEN2**.SSCTXIE
DMA Channel 3 Src: SSC Receive	15	level	set until cleared by software	SCU.**DMAIRC2**.SSC2RDY	SCU.**DMAIEN2**.SSCRXIE

1) Available only for products with SDADC. See the datasheet for details.

2) Each CCU6 interrupt can be assigned to any of the CCU6 interrupt nodes [3:0] via CCU6 registers CCU6.**INP**.

12.3.2 Interrupt Structure

An interrupt event source may be generated from the on-chip peripherals or from external. Detection of interrupt events is controlled by the respective on-chip peripherals. Interrupt status flags are available for determining which interrupt event has occurred, especially useful for an interrupt node which is shared by several event sources. Each interrupt node (except NMI) has a global enable/disable bit. In most cases, additional enable bits are provided for enabling/disabling particular interrupt events (provided for NMI events). No interrupt will be requested for any occurred event that has its interrupt enable bit disabled.

There is an interrupt masking bit EA available, which is used to globally enable or disable all interrupt requests (except NMI) to the core. Resetting bit EA to 0 only masks the pending interrupt requests from the core, but does not block the capture of incoming interrupt requests.

As displayed in **Figure 67**, the interrupt event will set the interrupt status flag which doubles as a pending interrupt request to the core. An active pending interrupt request will interrupt the core only if it is corresponding interrupt node is enabled. Once an interrupt node is serviced (interrupt acknowledged), its pending interrupt request (represented by the interrupt status flag) may be automatically cleared by hardware (the core).

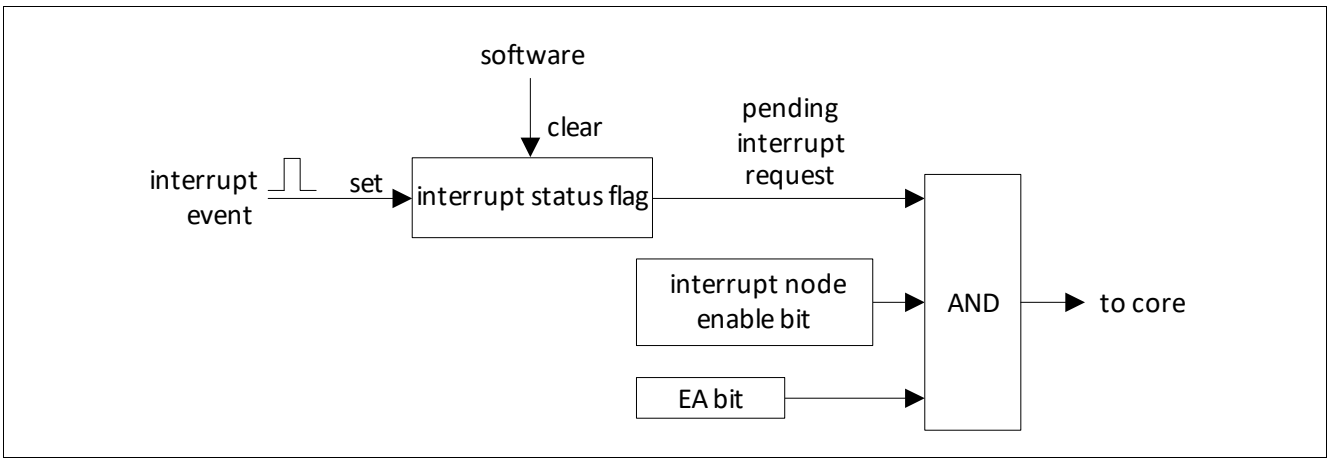


Figure 67 Interrupt Structure

For the TLE987x, interrupt sources like ADC1, MU and Bridge Driver (each have a dedicated interrupt node) will have their respective interrupt status flags in the dedicated registers. This flags are not cleared by the core once their corresponding pending interrupt request is serviced. They have to be cleared by software. For the UART which has its dedicated interrupt node, interrupt status flags RI and TI in register SCON will not be cleared by the core even when its pending interrupt request is serviced. The UART interrupt status flags (and hence the pending interrupt request) can only be cleared by software.

12.3.3 Interrupt Source and Vector

Each interrupt event source has an associated interrupt vector address for the interrupt node it belongs to. This vector is accessed to service the corresponding interrupt node request. The interrupt service of each interrupt node can be individually enabled or disabled via an enable bit. The assignment of the TLE987x interrupt sources to the interrupt vector address and the corresponding interrupt node enable bits are summarized in **Table 67**.

Table 70 Interrupt Vector Address

Interrupt Node	Assignment for TLE987x	Enable Bit	SFR
NMI	Watchdog Timer NMI	NMIWDT	NMICON
	PLL NMI	NMIPLL	
	NVM Operation Complete NMI	NMINVM	
	Overtemperature NMI	NMIOT	
	Oscillator Watchdog NMI	NMIOWD	
	NVM Map Error NMI	NMIMAP	
	ECC Error NMI	NMIECC	
	Supply Prewarning NMI	NMISUP	
INTISR[0]	GPT1_T2, GPT1_T3, GPT1_T4	GPT12	GPT12IEN
INTISR[1]	GPT2_T5, GPT2_T6, GPT2_CR	GPT12	GPT12IEN
INTISR[2]	MU/Timer3/SDADC[1)]	see SFR	Timer3_CTRL SYS_IRQ_CTRL
INTISR[3]	ADC1		ADC1
	VREF5V	SYS_IRQ_CTRL	
INTISR[4]	CCU6 Node 0	CCU6SR0	IRCON3.0
INTISR[5]	CCU6 Node 1	CCU6SR1	IRCON3.4
INTISR[6]	CCU6 Node 2	CCU6SR2	IRCON4.0
INTISR[7]	CCU6 Node 3	CCU6SRC3	IRCON4.4
INTISR[8]	SSC1	EIREN	MODIEN1.0
		TIREN	MODIEN1.1
		RIREN	MODIEN1.2
INTISR[9]	SSC2	EIREN	MODIEN2.0
		TIREN	MODIEN2.1
		RIREN	MODIEN2.2
INTISR[10]	UART1 Receive	RIEN1	MODIEN1.6
	UART1 Transmit	TIEN1	MODIEN1.7
	Timer 2 Overflow	TF2EN	T2_T2CON1.1
	T2EX	EXF2EN	T2_T2CON1.0
	LIN_OT_STS	LIN_CTRL_STS	LIN_CTRL_STS
	LIN_OC_STS		LIN_CTRL_STS
	TXD_TMOUT		LIN_CTRL_STS/
	EOFSYN	SYNEN	SYNCST
	ERRSYN		

Table 70 Interrupt Vector Address (cont'd)

Interrupt Node	Assignment for TLE987x	Enable Bit	SFR
INTISR[11]	UART2 Receive	RIEN2	MODIEN2.6
	UART2 Transmit	TIEN2	MODIEN2.7
	EINT2	EXINT2	IRCON0.2
	Timer 21 Overflow	TF2EN/	T21T2CON1.1
	T21EX	EXF2EN	T21T2CON1.0
INTISR[12]	EINT0	IE0	MODIEN3.0
INTISR[13]	EINT1	IE1	MODIEN4.0
INTISR[14]	BDRV-VDS(1-6)	BRDRV_IS	BRDRV_IRQ_CTRL
	CP-UV, CP-OV		
INTISR[15]	DMA	ECCIP3	DMA TRSEQ SSC1 SSC2

1) Available only for products with SDADC. See the datasheet for details.

12.3.4 Interrupt Priority

An interrupt that is currently being serviced can only be interrupted by a higher-priority interrupt, but not by another interrupt of the same or lower priority. Hence, an interrupt of the highest priority cannot be interrupted by any other interrupt request.

If two or more requests of different priority levels are received simultaneously, the request with the highest priority is serviced first. If requests of the same priority are received simultaneously, an internal polling sequence determines which request is serviced first. Thus, within each priority level, there is a second priority structure determined by the polling sequence as shown in **Table 69**.

Table 71 Interrupt Node Table

Service Request	Node ID	Description
GPT1	0	GPT1 interrupt (T2-T4)
GPT2	1	GPT2 interrupt (T5-T6, CR)
ADC2/Timer3	2	ADC2, Measurement Unit, VBG, SDADC[1)], Timer3, BEMF
ADC1	3	ADC1 interrupt / VREF5V Overload / VREF5V OV/UV
CCU0	4	CCU6 node 0 interrupt
CCU1	5	CCU6 node 1 interrupt
CCU2	6	CCU6 node 2 interrupt
CCU3	7	CCU6 node 3 interrupt
SSC1	8	SSC1 interrupt (receive, transmit, error)
SSC2	9	SSC2 interrupt (receive, transmit, error)
UART1	10	UART1 (ASC-LIN) interrupt (receive, transmit), Timer2, linsync1, LIN
UART2	11	UART2 interrupt (receive, transmit), Timer21, linsync2, External interrupt (EINT2)
EXINT0	12	External interrupt (EINT0), wakeup
EXINT1	13	External interrupt (EINT1)

Table 71 Interrupt Node Table (cont'd)

Service Request	Node ID	Description
BDRV/CP	14	Bridge Driver / Charge Pump
DMA	15	DMA Controller

1) Available only for products with SDADC. See the datasheet for details.

The interrupt priority is configured in the corresponding NVIC control register:

Table 72

Register Short name	Register Long Name	Offset Address	Reset Value
NVIC_IPR0	Interrupt Priority	400_H	$0000\ 0000_H$
NVIC_IPR1	Interrupt Priority	404_H	$0000\ 0000_H$
NVIC_IPR2	Interrupt Priority	408_H	$0000\ 0000_H$
NVIC_IPR3	Interrupt Priority	$40C_H$	$0000\ 0000_H$

For further description see Arm_Architecture_v7n_Reference_Manual.

12.4 Interrupt Handling

See also Arm_Architecture_v7n_Reference_Manual. The most important Interrupt Registers are listed below. This registers are dedicated to the 16 available interrupt nodes. For all nodes which are a combination of several interrupt requests, the corresponding control and status registers are located in the System Control Unit (SCU) or the System Control Unit for the Power Modules (SCU_PM).

Table 73

Register Short name	Register Long Name	Offset Address	Reset Value
NVIC_ISER0	Interrupt Set-Enable	100_H	$0000\ 0000_H$
NVIC_ICER0	Interrupt Clear-Enable	180_H	$0000\ 0000_H$
NVIC_ISPR0	Interrupt Set-Pending	200_H	$0000\ 0000_H$
NVIC_ICPR0	Interrupt Clear-Pending	280_H	$0000\ 0000_H$
NVIC_IABR0	Active Bit Register	300_H	$0000\ 0000_H$

12.5 Register Definition

Interrupt registers are used for interrupt node enable, external interrupt control, interrupt flags and interrupt priority setting.

Table 74 Register Address Space Interrupt Registers

Module	Base Address	End Address	Note
SCU	50005000_H	$50005FFF_H$	

12.5.1 Interrupt Node Enable Registers

Register IEN0 contains the global interrupt masking bit (EA), which can be cleared to block all pending interrupt requests at once.

The NMI interrupt vector is shared by a number of sources, each of which can be enabled or disabled individually via register NMICON.

This register IEN0 is reset by RESET_TYPE_4.

Register IEN0 is being initialized by the BootROM during the boot-up process to a value of $7F_H$.

IEN0
Interrupt Enable Register 0 **($01C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
EA	**RES**						
rw	r						

Field	Bits	Type	Description
EA	7	rw	**Global Interrupt Mask** 0_B All pending interrupt requests (except NMI) are blocked from the core. 1_B Pending interrupt requests are not blocked from the core.
RES	6:0	r	**Reserved** Returns 0 if read; should be written with 0.

NMICON
NMI Control Register **(024_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
NMISUP	**NMIECC**	**NMIMAP**	**NMIOWD**	**NMIOT**	**NMINVM**	**NMIPLL**	**NMIWDT**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
NMISUP	7	rw	**Supply Prewarning NMI Enable** 0_B Supply NMI is disabled. 1_B Supply NMI is enabled.
NMIECC	6	rw	**ECC Error NMI Enable** 0_B ECC Error NMI is disabled. 1_B ECC Error NMI is enabled.
NMIMAP	5	rw	**NVM Map Error NMI Enable** 0_B NVM Map Error NMI is disabled. 1_B NVM Map Error NMI is enabled.
NMIOWD	4	rw	**Oscillator Watchdog NMI Enable** 0_B Oscillator watchdog NMI is disabled. 1_B Oscillator watchdog NMI is enabled.
NMIOT	3	rw	**NMI OT Enable** 0_B NMI OT is disabled. 1_B NMI OT is enabled.
NMINVM	2	rw	**NVM Operation Complete NMI Enable** 0_B NVM operation complete NMI is disabled. 1_B NVM operation complete NMI is enabled.
NMIPLL	1	rw	**PLL Loss of Lock NMI Enable** 0_B PLL Loss of Lock NMI is disabled. 1_B PLL Loss of Lock NMI is enabled.
NMIWDT	0	rw	**Watchdog Timer NMI Enable** 0_B WDT NMI is disabled. 1_B WDT NMI is enabled.

12.5.2 External Interrupt Control Registers

The external interrupts, EXT_INT[2:0], are driven into the TLE987x from the ports. External interrupts can be positive, negative or double edge triggered. Register EXICON0 specifies the active edge for the external interrupt.

If the external interrupt is positive (negative) edge triggered, the external source must hold the request pin low (high) for at least one CCLK cycle, and then hold it high (low) for at least one CCLK cycle to ensure that the transition is recognized.

External interrupts 2 share their interrupt node with other interrupt sources. Therefore in addition to the corresponding interrupt node enable, external interrupt 2 may be disabled individually, and are disabled by default after reset.

Note: Several external interrupts support alternative input pin, selected via MODPISEL register in the SCU. When switching inputs, the active edge/level trigger select and the level on the associated pins should be considered to prevent unintentional interrupt generation.

Register EXICON0 is being initialized by the BootROM during the boot-up process to a value of 00_H.

EXICON0
External Interrupt Control Register 0 **(028_H)** **Reset Value: 30_H**

7	6	5	4	3	2	1	0
MON_Trig_Sel		**EXINT2**		**EXINT1**		**EXINT0**	
rw		rw		rw		rw	

Field	Bits	Type	Description
MON_Trig_Sel	7:6	rw	**MON Input Trigger Select** 00_B external interrupt MON is disabled. 01_B Interrupt on rising edge. 10_B Interrupt on falling edge. 11_B Interrupt on both rising and falling edge.
EXINT2	5:4	rw	**External Interrupt 2 Trigger Select** 00_B external interrupt 2 is disabled. 01_B Interrupt on rising edge. 10_B Interrupt on falling edge. 11_B Interrupt on both rising and falling edge.
EXINT1	3:2	rw	**External Interrupt 1 Trigger Select** 00_B external interrupt 1 is disabled. 01_B Interrupt on rising edge. 10_B Interrupt on falling edge. 11_B Interrupt on both rising and falling edge.
EXINT0	1:0	rw	**External Interrupt 0 Trigger Select** 00_B external interrupt 0 is disabled. 01_B Interrupt on rising edge. 10_B Interrupt on falling edge. 11_B Interrupt on both rising and falling edge.

12.5.3 Interrupt Flag Registers

The interrupt flags for the different interrupt sources are located in several special function registers. This section describes the interrupt flags located in system registers or external interrupts belonging to system. Other interrupt flags located in respective module registers are described in the specific module chapter.

In case of software and hardware access to a flag bit at the same time, hardware will have higher priority.

IRCON0
Interrupt Request Register 0 **(004_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
MONF	**MONR**	**EXINT2F**	**EXINT2R**	**EXINT1F**	**EXINT1R**	**EXINT0F**	**EXINT0R**
r	r	r	r	r	r	r	r

Field	Bits	Type	Description
MONF	7	r	**Interrupt Flag for External Interrupt MON on falling edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on falling edge event has not occurred. 1_B Interrupt on falling edge event has occurred.
MONR	6	r	**Interrupt Flag for External Interrupt MON on rising edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on rising edge event has not occurred. 1_B Interrupt on rising edge event has occurred.
EXINT2F	5	r	**Interrupt Flag for External Interrupt 2x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on falling edge event has not occurred. 1_B Interrupt on falling edge event has occurred.
EXINT2R	4	r	**Interrupt Flag for External Interrupt 2x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on rising edge event has not occurred. 1_B Interrupt on rising edge event has occurred.
EXINT1F	3	r	**Interrupt Flag for External Interrupt 1x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on falling edge event has not occurred. 1_B Interrupt on falling edge event has occurred.
EXINT1R	2	r	**Interrupt Flag for External Interrupt 1x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on rising edge event has not occurred. 1_B Interrupt on rising edge event has occurred.
EXINT0F	1	r	**Interrupt Flag for External Interrupt 0x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on falling edge event has not occurred. 1_B Interrupt on falling edge event has occurred.

Field	Bits	Type	Description
EXINT0R	0	r	**Interrupt Flag for External Interrupt 0x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B Interrupt on rising edge event has not occurred. 1_B Interrupt on rising edge event has occurred.

IRCON0CLR
Interrupt Request 0 Clear Register **($02C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
MONFC	**MONRC**	**EXINT2FC**	**EXINT2RC**	**EXINT1FC**	**EXINT1RC**	**EXINT0FC**	**EXINT0RC**
w	w	w	w	w	w	w	w

Field	Bits	Type	Description
MONFC	7	w	**Interrupt Flag for External Interrupt MON on falling edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on falling edge event is not cleared. 1_B Interrupt on falling edge event is cleared.
MONRC	6	w	**Interrupt Flag for External Interrupt MON on rising edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on rising edge event is not cleared. 1_B Interrupt on rising edge event is cleared.
EXINT2FC	5	w	**Interrupt Flag for External Interrupt 2x on falling edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on falling edge event is not cleared. 1_B Interrupt on falling edge event is cleared.
EXINT2RC	4	w	**Interrupt Flag for External Interrupt 2x on rising edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on rising edge event is not cleared. 1_B Interrupt on rising edge event is cleared.
EXINT1FC	3	w	**Interrupt Flag for External Interrupt 1x on falling edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on falling edge event is not cleared. 1_B Interrupt on falling edge event is cleared.
EXINT1RC	2	w	**Interrupt Flag for External Interrupt 1x on rising edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on rising edge event is not cleared. 1_B Interrupt on rising edge event is cleared.
EXINT0FC	1	w	**Interrupt Flag for External Interrupt 0x on falling edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on falling edge event is not cleared. 1_B Interrupt on falling edge event is cleared.

Field	Bits	Type	Description
EXINT0RC	0	w	**Interrupt Flag for External Interrupt 0x on rising edge clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt on rising edge event is not cleared. 1_B Interrupt on rising edge event is cleared.

IRCON1

Interrupt Request Register 1 **(008_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES					**RIR**	**TIR**	**EIR**
r					r	r	r

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.
RIR	2	r	**Receive Interrupt Flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.
TIR	1	r	**Transmit Interrupt Flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.
EIR	0	r	**Error Interrupt Flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.

IRCON1CLR
Interrupt Request 1 Clear Register **(178_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES					**RIRC**	**TIRC**	**EIRC**
		r			w	w	w

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.
RIRC	2	w	**Receive Interrupt Flag for SSC1 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.
TIRC	1	w	**Transmit Interrupt Flag for SSC1 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.
EIRC	0	w	**Error Interrupt Flag for SSC1 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.

IRCON2
Interrupt Request Register 2 **($00C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES					**RIR**	**TIR**	**EIR**
		r			r	r	r

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.
RIR	2	r	**Receive Interrupt Flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.
TIR	1	r	**Transmit Interrupt Flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.
EIR	0	r	**Error Interrupt Flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.

IRCON2CLR
Interrupt Request 2 Clear Register **($17C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES					**RIRC**	**TIRC**	**EIRC**
		r			w	w	w

Field	Bits	Type	Description
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.
RIRC	2	w	**Receive Interrupt Flag for SSC2 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.
TIRC	1	w	**Transmit Interrupt Flag for SSC2 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.
EIRC	0	w	**Error Interrupt Flag for SSC2 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.

IRCON3
Interrupt Request Register 3 **(010_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**CCU6SR1**	**RES**			**CCU6SR0**
r			r	r			r

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR1	4	r	**Interrupt Flag 1 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.
RES	3:1	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR0	0	r	**Interrupt Flag 0 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.

IRCON3CLR
Interrupt Request 3 Clear Register **(190_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**CCU6SR1C**	**RES**			**CCU6SR0C**
r			w	r			w

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR1C	4	w	**Interrupt Flag 1 for CCU6 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.
RES	3:1	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR0C	0	w	**Interrupt Flag 0 for CCU6 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.

IRCON4
Interrupt Request Register 4 **(014_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**CCU6SR3**	**RES**			**CCU6SR2**
r			r	r			r

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR3	4	r	**Interrupt Flag 3 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.
RES	3:1	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR2	0	r	**Interrupt Flag 2 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event has not occurred. 1_B Interrupt event has occurred.

IRCON4CLR
Interrupt Request 4 Clear Register **(194_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**CCU6SR3C**	**RES**			**CCU6SR2C**
r			w	r			w

Field	Bits	Type	Description
RES	7:5	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR3C	4	w	**Interrupt Flag 3 for CCU6 Clear** This bit is set by hardware and can only be cleared by software. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.
RES	3:1	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR2C	0	w	**Interrupt Flag 2 for CCU6 Clear** This bit is set by software and can only be cleared by hardware. 0_B Interrupt event is not cleared. 1_B Interrupt event is cleared.

GPT12IRC
Timer and Counter Control/Status Register **(160_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		**CR**	**T6**	**T5**	**T4**	**T3**	**T2**
r		r	r	r	r	r	r

Field	Bits	Type	Description
RES	7:6	r	**Reserved** This Flag is always read as zero.
CR	5	r	**GPT Module 2 Capture Reload Interrupt Status** Capture Reload Event of GPT1 Module Interrupt Status 0_B No Capture Reload Interrupt has occurred. 1_B Capture Reload Interrupt has occurred.
T6	4	r	**GPT Module 2Timer6 Interrupt Status** Timer 6 of GPT Module Interrupt Status 0_B No Timer 6 Interrupt has occurred. 1_B Timer 6 Interrupt has occurred.
T5	3	r	**GPT Module 2 Timer5 Interrupt Status** Timer 5 of GPT2 Module Interrupt Status 0_B No Timer 5 Interrupt has occurred. 1_B Timer 5 Interrupt has occurred.
T4	2	r	**GPT Module 1 Timer4 Interrupt Status** Timer 4 of GPT1 Module Interrupt Status 0_B No Timer 4 Interrupt has occurred. 1_B Timer 4 Interrupt has occurred.
T3	1	r	**GPT Module 1 Timer3 Interrupt Status** Timer 3 of GPT1 Module Interrupt Status 0_B No Timer 3 Interrupt has occurred. 1_B Timer 3 Interrupt has occurred.
T2	0	r	**GPT Module 1 Timer 2 Interrupt Status** Timer 2 of GPT1 Module Interrupt Status 0_B No Timer 2 Interrupt has occurred. 1_B Timer 2 Interrupt has occurred.

GPT12ICLR
Timer and Counter Control/Status Clear Register (164_H) **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		CRC	T6C	T5C	T4C	T3C	T2C
r		w	w	w	w	w	w

Field	Bits	Type	Description
RES	7:6	r	**Reserved** This Flag is always read as zero.
CRC	5	w	**GPT Module 2 Capture Reload Interrupt Status Clear** Capture Reload Event of GPT1 Module Interrupt Status Clear 0_B No Capture Reload Interrupt is cleared. 1_B Capture Reload Interrupt is cleared.
T6C	4	w	**GPT Module 2Timer6 Interrupt Status Clear** Timer 6 of GPT Module Interrupt Status Clear 0_B No Timer 6 Interrupt is cleared. 1_B Timer 6 Interrupt is cleared.
T5C	3	w	**GPT Module 2 Timer5 Interrupt Status Clear** Timer 5 of GPT2 Module Interrupt Status Clear 0_B No Timer 5 Interrupt is cleared. 1_B Timer 5 Interrupt is cleared.
T4C	2	w	**GPT Module 1 Timer4 Interrupt Status Clear** Timer 4 of GPT1 Module Interrupt Status Clear 0_B No Timer 4 Interrupt is cleared. 1_B Timer 4 Interrupt is cleared.
T3C	1	w	**GPT Module 1 Timer3 Interrupt Status Clear** Timer 3 of GPT1 Module Interrupt Status Clear 0_B No Timer 3 Interrupt is cleared. 1_B Timer 3 Interrupt is cleared.
T2C	0	w	**GPT Module 1 Timer 2 Interrupt Status Clear** Timer 2 of GPT1 Module Interrupt Status Clear 0_B No Timer 2 Interrupt is cleared. 1_B Timer 2 Interrupt is cleared.

DMAIRC1
DMA Interrupt Control Register 1 **(154_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
CH8	**CH7**	**CH6**	**CH5**	**CH4**	**CH3**	**CH2**	**CH1**
r	r	r	r	r	r	r	r

Field	Bits	Type	Description
CH8	7	r	**DMA Channel 11 Interrupt Status (ADC1 Channel 7)** DMA Transfer of Channel has finished 0_B No DMA Channel 11 Interrupt has occurred. 1_B DMA Channel 11 Interrupt has occurred.
CH7	6	r	**DMA Channel 10 Interrupt Status (ADC1 Channel 6)** DMA Transfer of Channel has finished 0_B No DMA Channel 10 Interrupt has occurred. 1_B DMA Channel 10 Interrupt has occurred.
CH6	5	r	**DMA Channel 9 Interrupt Status (ADC1 Channel 5)** DMA Transfer of Channel has finished 0_B No DMA Channel 9 Interrupt has occurred. 1_B DMA Channel 9 Interrupt has occurred.
CH5	4	r	**DMA Channel 8 Interrupt Status (ADC1 Channel 4)** DMA Transfer of Channel has finished 0_B No DMA Channel 8 Interrupt has occurred. 1_B DMA Channel 8 Interrupt has occurred.
CH4	3	r	**DMA Channel 7 Interrupt Status (ADC1 Channel 3)** DMA Transfer of Channel has finished 0_B No DMA Channel 7 Interrupt has occurred. 1_B DMA Channel 7 Interrupt has occurred.
CH3	2	r	**DMA Channel 6 Interrupt Status (ADC1 Channel 2)** DMA Transfer of Channel has finished 0_B No DMA Channel 6 Interrupt has occurred. 1_B DMA Channel 6 Interrupt has occurred.
CH2	1	r	**DMA Channel 5 Interrupt Status (ADC1 Channel 1)** DMA Transfer of Channel has finished 0_B No DMA Channel 5 Interrupt has occurred. 1_B DMA Channel 5 Interrupt has occurred.
CH1	0	r	**DMA Channel 4 Interrupt Status (ADC1 Channel 0)** DMA Transfer of Channel has finished 0_B No DMA Channel 4 Interrupt has occurred. 1_B DMA Channel 4 Interrupt has occurred.

Note: If a conversion is finished for the dedicated channel, the corresponding flag above will be set.

DMAIRC1CLR
DMA Interrupt Control 1 Clear Register **(184_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
CH8C	**CH7C**	**CH6C**	**CH5C**	**CH4C**	**CH3C**	**CH2C**	**CH1C**
w	w	w	w	w	w	w	w

Field	Bits	Type	Description
CH8C	7	w	**DMA Channel 11 Interrupt Status Clear (ADC1 Channel 7)** Transfer of data, triggered by ADC1 Channel 7, has finished 0_B No DMA Channel 11 Interrupt is cleared. 1_B DMA Channel 11 Interrupt is cleared.
CH7C	6	w	**DMA Channel 10 Interrupt Status Clear (ADC1 Channel 6)** Transfer of data, triggered by ADC1 Channel 6, has finished 0_B No DMA Channel 10 Interrupt is cleared. 1_B DMA Channel 10 Interrupt is cleared.
CH6C	5	w	**DMA Channel 9 Interrupt Status Clear (ADC1 Channel 5)** Transfer of data, triggered by ADC1 Channel 5, has finished 0_B No DMA Channel 9 Interrupt is cleared. 1_B DMA Channel 9 Interrupt is cleared.
CH5C	4	w	**DMA Channel 8 Interrupt Status Clear (ADC1 Channel 4)** Transfer of data, triggered by ADC1 Channel 4, has finished 0_B No DMA Channel 8 Interrupt is cleared. 1_B DMA Channel 8 Interrupt is cleared.
CH4C	3	w	**DMA Channel 7 Interrupt Status Clear (ADC1 Channel 3)** Transfer of data, triggered by ADC1 Channel 3, has finished 0_B No DMA Channel 7 Interrupt is cleared. 1_B DMA Channel 7 Interrupt is cleared.
CH3C	2	w	**DMA Channel 6 Interrupt Status Clear (ADC1 Channel 2)** Transfer of data, triggered by ADC1 Channel 2, has finished 0_B No DMA Channel 6 Interrupt is cleared. 1_B DMA Channel 6 Interrupt is cleared.
CH2C	1	w	**DMA Channel 5 Interrupt Status Clear (ADC1 Channel 1)** Transfer of data, triggered by ADC1 Channel 1, has finished 0_B No DMA Channel 5 Interrupt is cleared. 1_B DMA Channel 5 Interrupt is cleared.
CH1C	0	w	**DMA Channel 4 Interrupt Status Clear (ADC1 Channel 0)** Transfer of data, triggered by ADC1 Channel 0, has finished 0_B No DMA Channel 4 Interrupt is cleared. 1_B DMA Channel 4 Interrupt is cleared.

Note: If a conversion is finished for the dedicated channel, the corresponding flag above will be set.

DMAIRC2
ADC1 Interrupt Control Register 2 **(158_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**SDADC**	**GPT12**	**SSC2RDY**	**SSC1RDY**	**TRSEQ2DY**	**TRSEQ1DY**	**STRDY**
r	r	r	r	r	r	r	r

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; Should be written with 0.
SDADC	6	r	**DMA SDADC Transfer Ready (DMA channel 13)**[1)] Transfer of SDADC data has finished 0_B No DMA SDADC Transfer Ready Interrupt has occurred. 1_B DMA SDADC Transfer Ready Interrupt has occurred.
GPT12	5	r	**DMA Channel 12 Interrupt Status (GPT12/Timer3)** Transfer of data, triggered by GPT12/Timer3, has finished 0_B No DMA GPT12 Transfer Ready Interrupt has occurred. 1_B DMA GPT12 Transfer Ready Interrupt has occurred.
SSC2RDY	4	r	**DMA Channel 3 Interrupt Status (SSCx Receive)** Transfer of data, triggered by SSCx Receive, has finished 0_B No DMA SSCx Transfer Ready Interrupt has occurred. 1_B DMA SSCx Transfer Ready Interrupt has occurred.
SSC1RDY	3	r	**DMA Channel 2 Interrupt Status (SSCx Transmit)** Transfer of data, triggered by SSCx Transmit, has finished 0_B No DMA SSCx Transfer Ready Interrupt has occurred. 1_B DMA SSCx Transfer Ready Interrupt has occurred.
TRSEQ2DY	2	r	**DMA Channel 1 Interrupt Status (ADC1 ESM)** Transfer of data, triggered by ADC1 ESM, has finished 0_B No Transfer Sequence Ready Interrupt has occurred. 1_B Transfer Sequence Ready Interrupt has occurred.
TRSEQ1DY	1	r	**DMA Channel 0 Interrupt Status (ADC1 Sequence)** Transfer of data, triggered by ADC1 Sequence, has finished 0_B No Transfer Sequence Ready Interrupt has occurred. 1_B Transfer Sequence Ready Interrupt has occurred.
STRDY	0	r	**DMA Single Transfer Ready** This Flag is an OR of the DMA_IRC1 Register Bits 0_B No Single Transfer Interrupt has occurred. 1_B Single Transfer Ready Interrupt has occurred.

1) Available only for products with SDADC. See the datasheet for details.

DMAIRC2CLR
ADC1 Interrupt Control 2 Clear Register **(188_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**SDADCC**	**GPT12C**	**SSC2C**	**SSC1C**	**TRSEQ2DYC**	**TRSEQ1DYC**	**RES**
r	w	w	w	w	w	w	r

Field	Bits	Type	Description
RES	7	r	**Reserved** Returns 0 if read; Should be written with 0.
SDADCC	6	w	**DMA SDADC Transfer Ready (DMA channel 13) Clear**[1)] Transfer of SDADC data has finished 0_B No DMA SDADC Transfer Ready Interrupt is cleared. 1_B DMA SDADC Transfer Ready Interrupt is cleared.
GPT12C	5	w	**DMA Channel 12 Interrupt Status Clear (GPT12/Timer3)** Transfer of data has finished cleared 0_B No DMA GPT12 Transfer Ready Interrupt is cleared. 1_B DMA GPT12 Transfer Ready Interrupt is cleared.
SSC2C	4	w	**DMA Channel 3 Interrupt Status Clear (SSCx Receive)** Transfer of data has finished cleared 0_B No DMA SSC2 Transfer Ready Interrupt is cleared. 1_B DMA SSC2 Transfer Ready Interrupt is cleared.
SSC1C	3	w	**DMA Channel 2 Interrupt Status Clear (SSCx Transmit)** Transfer of data has finished cleared 0_B No DMA SSC1 Transfer Ready Interrupt is cleared. 1_B DMA SSC1 Transfer Ready Interrupt is cleared.
TRSEQ2DYC	2	w	**DMA Channel 1 Interrupt Status Clear (ADC1 ESM)** Transfer of data has finished clear 0_B No Transfer Sequence Ready Interrupt is cleared. 1_B Transfer Sequence Ready Interrupt is cleared.
TRSEQ1DYC	1	w	**DMA Channel 0 Interrupt Status Clear (ADC1 Sequence)** Transfer of data has finished clear 0_B No Transfer Sequence Ready Interrupt is cleared. 1_B Transfer Sequence Ready Interrupt is cleared.
RES	0	r	**Reserved** Returns 0 if read; Should be written with 0.

1) Available only for products with SDADC. See the datasheet for details.

Each NMI event and status flag is retained across these resets: 1) WDT reset, 2) soft reset. These include all the flags of NMISR register: FNMIWDT, FNMIPLL, FNMINVM, FNMIOT, FNMIOWD, FNMIMAP, and indirectly, FNMIECC and FNMISUP. In the case of NMIs with shared source i.e. watchdog, ECC or supply prewarning NMI, the respective indicator or event flags not located in NMISR are also retained. Refer to **Chapter 6.6.4** for identifying the NMI event.

NMISR
NMI Status Register **(018_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
FNMISUP	**FNMIECC**	**FNMIMAP**	**FNMIOWD**	**FNMIOT**	**FNMINVM**	**FNMIPLL**	**FNMIWDT**
r	r	r	r	r	r	r	r

Field	Bits	Type	Description
FNMISUP	7	r	**Supply Prewarning NMI Flag** This flag is cleared automatically by hardware when the corresponding event flags are cleared. 0_B No supply prewarning NMI has occurred. 1_B Supply prewarning has occurred.
FNMIECC	6	r	**ECC Error NMI Flag** This flag is cleared automatically by hardware when the corresponding enabled event flags are cleared. 0_B No uncorrectable ECC error has occurred on NVM, XRAM. 1_B Uncorrectable ECC error has occurred on NVM, RAM.
FNMIMAP	5	r	**NVM Map Error NMI Flag** This bit is set by hardware and can only be cleared by software. 0_B No NVM Map Error NMI has occurred. 1_B NVM Map Error has occurred.
FNMIOWD	4	r	**Oscillator Watchdog or MI_CLK Watchdog NMI Flag** This bit is set by hardware and can only be cleared by software. 0_B No oscillator / MI_CLK watchdog NMI has occurred. 1_B Oscillator / MI_CLK watchdog event has occurred.
FNMIOT	3	r	**Over-temperature NMI Flag** This bit is set by hardware and can only be cleared by software. As this is a shared NMI source, this flag should be cleared after checking and clearing the corresponding event flags. 0_B No OT NMI has occurred. 1_B OT NMI event has occurred.
FNMINVM	2	r	**NVM Operation Complete NMI Flag** This bit is set by hardware and can only be cleared by software. 0_B No NVM NMI has occurred. 1_B NVM operation complete event has occurred.
FNMIPLL	1	r	**PLL NMI Flag** This bit is set by hardware and can only be cleared by software. 0_B No PLL NMI has occurred. 1_B PLL loss-of-lock has occurred.

Field	Bits	Type	Description
FNMIWDT	0	r	**Watchdog Timer NMI Flag** This bit is set by hardware and can only be cleared by software. As this is a shared NMI source, this flag should be cleared after checking and clearing the corresponding event flags. 0_B No watchdog NMI has occurred. 1_B WDT prewarning has occurred.

This register NMISR is reset by RESET_TYPE_4.

NMICLR
NMI Clear Register **(000_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
NMISUPC	**NMIECCC**	**NMIMAPC**	**NMIOWDC**	**NMIOTC**	**NMINVMC**	**NMIPLLC**	**NMIWDTC**
w	w	w	w	w	w	w	w

Field	Bits	Type	Description
NMISUPC	7	w	**Supply Prewarning NMI Clear** 0_B Supply NMI is not cleared. 1_B Supply NMI is cleared.
NMIECCC	6	w	**ECC Error NMI Clear** 0_B ECC Error NMI is not cleared. 1_B ECC Error NMI is cleared.
NMIMAPC	5	w	**NVM Map Error NMI Clear** 0_B NVM Map Error NMI is not cleared. 1_B NVM Map Error NMI is cleared.
NMIOWDC	4	w	**Oscillator Watchdog NMI Clear** 0_B Oscillator watchdog NMI is not cleared. 1_B Oscillator watchdog NMI is cleared.
NMIOTC	3	w	**NMI OT Clear** 0_B NMI OT is not cleared. 1_B NMI OT is cleared.
NMINVMC	2	w	**NVM Operation Complete NMI Clear** 0_B NVM operation complete NMI is not cleared. 1_B NVM operation complete NMI is cleared.
NMIPLLC	1	w	**PLL Loss of Lock NMI Clear** 0_B PLL Loss of Lock NMI is not cleared. 1_B PLL Loss of Lock NMI is cleared.
NMIWDTC	0	w	**Watchdog Timer NMI Clear** 0_B WDT NMI is not cleared. 1_B WDT NMI is cleared.

12.6 Interrupt Priority Registers

Each interrupt node can be individually programmed to one of the 16 priority levels available. The user can set them in the corresponding **NVIC_IPRx** Register (see Core Chapter).

13 Watchdog timer (WDT1)

13.1 Features

There are two watchdog timers in the system. The watchdog timer (WDT) within the system control unit – digital modules (see SCU_DM) and the Watchdog Timer (WDT1) located within the system control unit – power modules (see SCU_PM). The watchdog timer WDT1 is described in this section.

In Active mode, the WDT1 acts as a windowed watchdog timer, which provides a highly reliable and safe way to recover from software or hardware failures.

The WDT1 is always enabled in Active mode. In Sleep mode, Stop Mode and SWD mode (Debug mode), the WDT1 is automatically disabled.

Functional Features

- Windowed watchdog timer with programmable timing in Active mode.
- Long-open window (typ. 80 ms) after power-up, reset, wake-up.
- Short-open window (typ. 30 ms) to facilitate flash programming.
- Disabled during debugging.
- Safety shutdown to Sleep mode after 5 missed WDT1 services.

13.2 Introduction

The behavior of the watchdog timer in Active mode is illustrated in **Figure 68**.

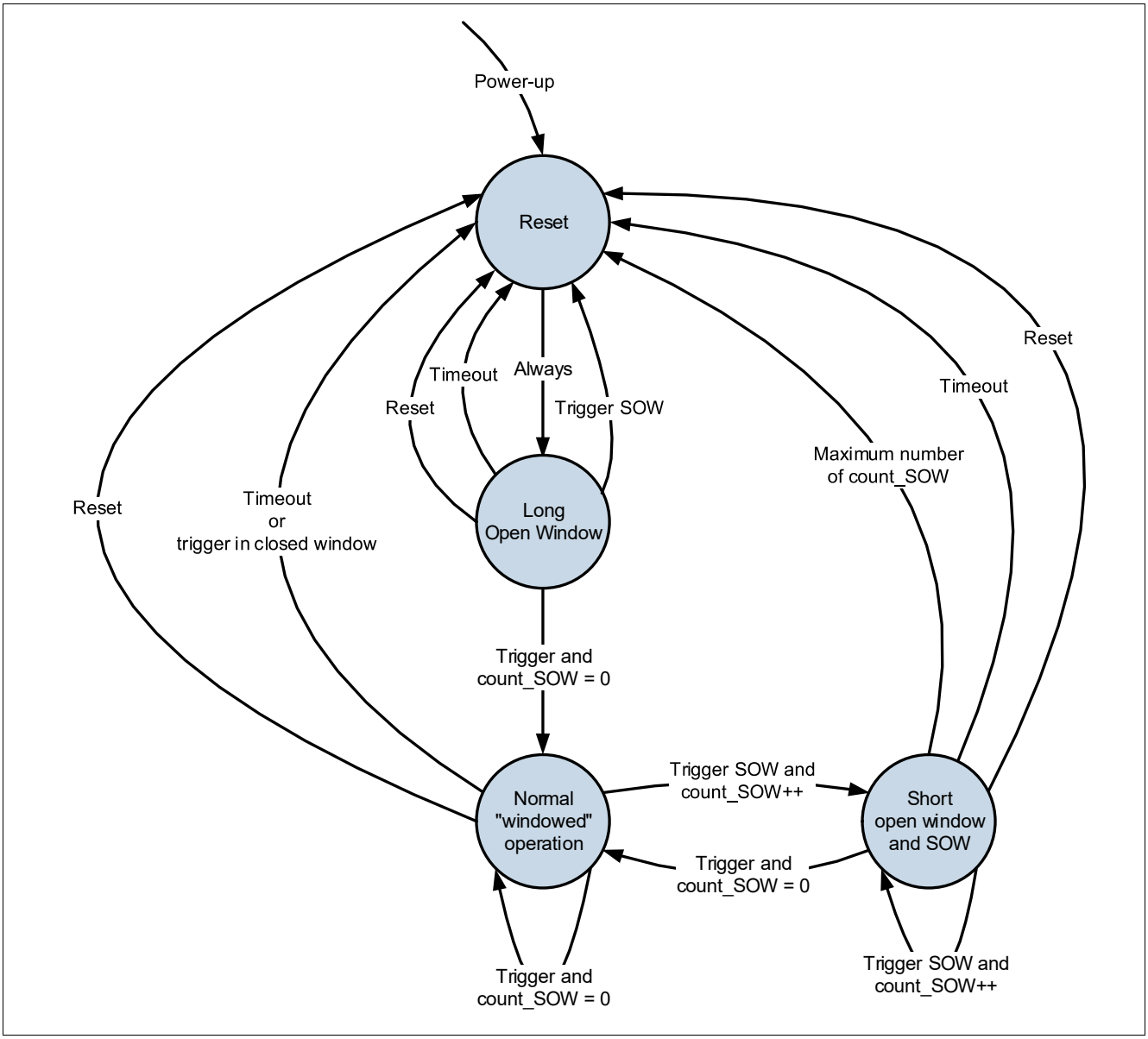


Figure 68 Watchdog timer behavior

13.3 Functional Description

13.3.1 Modes of Operation

The mode transition from the low power modes (WDT1 off) to active (WDT1 on) automatically initializes WDT1 to start in long open window mode.

13.3.2 Normal Operation

Software has to trigger the watchdog by writing to the **WDT1_TRIG** register. By triggering the watchdog also the length of the next watchdog period is selected inherently. The next period starts immediately with the trigger.

After Reset the WDT1 is starting with a long open window. The WDT1 has to be triggered within this long open window, otherwise a reset will be generated at the end of the long open window. After this first trigger the WDT1 operates in a window watchdog mode. Triggering of a short open window during the long open window is not allowed and will also cause a WDT1 reset.

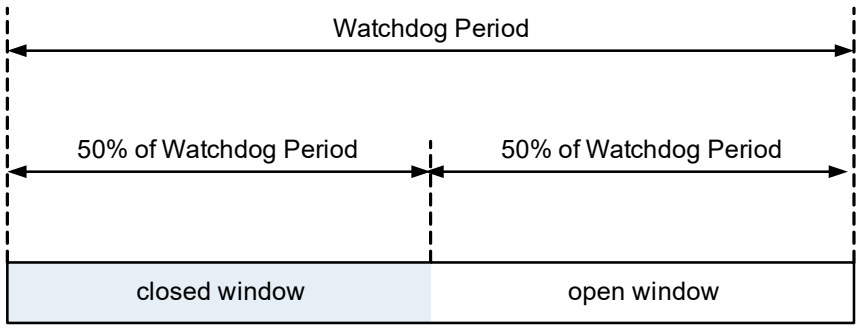


Figure 69 Windowed Watchdog

The first half of the watchdog period is the closed window and the second half is the open window. A trigger of the watchdog has to be done in the open window only. Any trigger in the closed window or failing to trigger the watchdog within the watchdog period will cause a reset. The reset will be indicated by the bit **PMU_ExtWDT** inside **PMU_RESET_STS1** register located inside PMU.

Effective open window (safe trigger point)

Due to the variations in the clock source of the WDT1 the effective usable open window, and therefore a safe trigger point, is shorter than 50% of the watchdog period as shown in **Figure 70**.

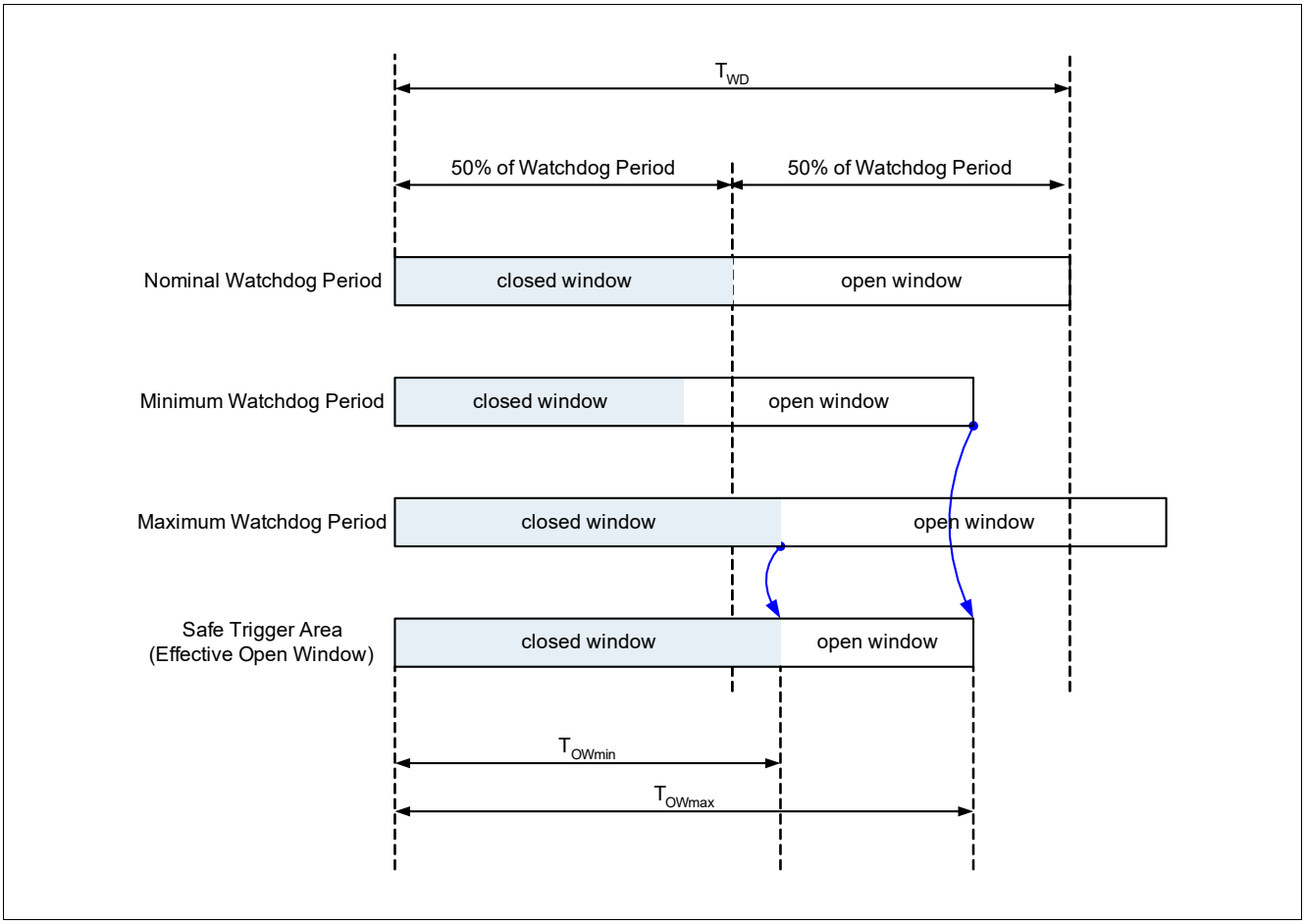


Figure 70 Effective open window

E.g. for a variation of 20% and a nominal watchdog period of T_{WD} the start of the effective open window T_{OWmin} is shifted back by 10%, and the end of the effective open window T_{OWmax} is shifted forward by 20%.

Short open window

Under certain programming conditions, e.g. NVM programming, it might be desired to interrupt the normal windowed watchdog operation. For this purpose a special trigger of a short open window (see **Figure 71**) allows to discard the current window period (also within the closed window) and immediately starts a short open window. The short open window has a fixed length of TSOW (typ. 30 ms) independent of the settings of the WDP_SEL bits.

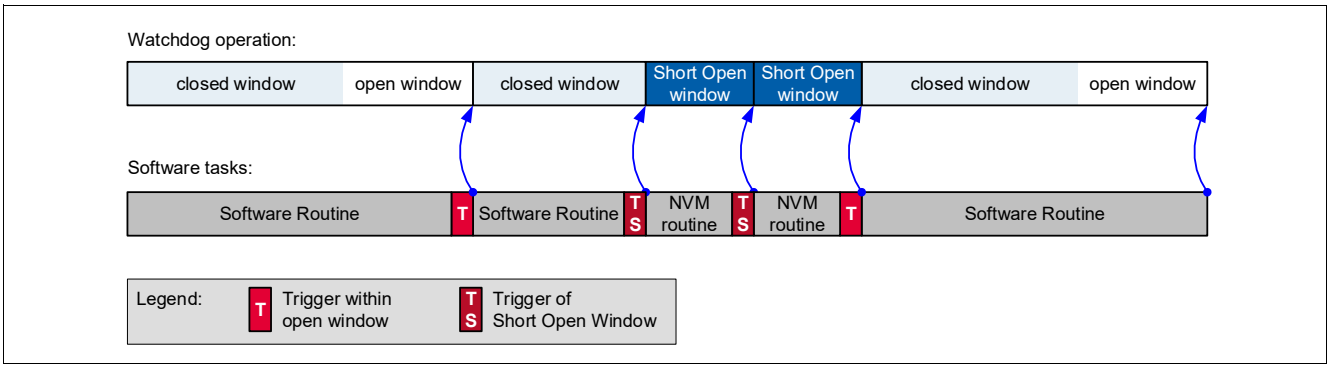


Figure 71 Short Open Window

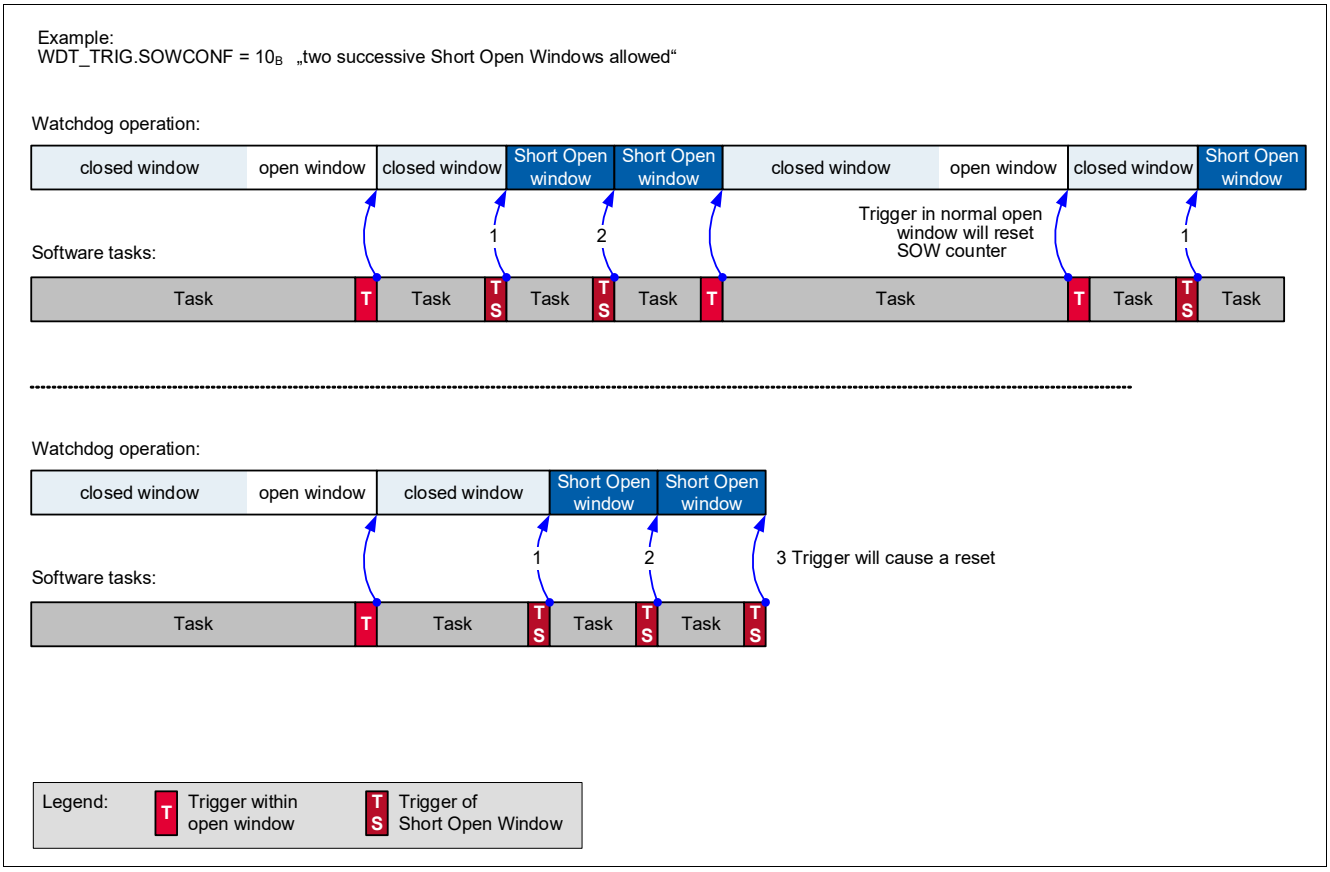


Figure 72 SOW Counter

The mechanism of inserting Short Open Windows has to be enabled/configured with the bits SOWCONF. The configuration allows to insert a maximum of three consecutive Short Open Windows. Each trigger of the Short Open Window will increase a SOW counter, if the SOW counter exceeds the maximum configured value a reset will be generated. The SOW counter value is reset to 0 by a normal trigger.

13.4 Register Definition

Table 75 shows the module base addresses.

Table 75 Register Address Space: SCU_PM Module Base Address List

Module	Base Address	End Address	Note
SCUPM	50006000_H	$50006FFF_H$	SCU Power Modules

Table 76 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
WDT1_TRIG	WDT1 Watchdog Control	34_H	$0000\ 0108_H$

The registers are addressed wordwise.

WDT1 Watchdog Control

The register is reset by RESET_TYPE_3.

WDT1_TRIG
WDT1 Watchdog Control **(34_H)** **Reset Value: $0000\ 0108_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES							RES	SOWCONF		WDP_SEL					
r							r	rw		rw					

Field	Bits	Type	Description
RES	31:9	r	**Reserved** Always read as 0
RES	8	r	**Reserved** Always read as 1
SOWCONF	7:6	rw	**Short Open Window Configuration** 00_B **DIS**, Short Open Windows disabled[1)] 01_B **SOW1**, one successive Short Open Window allowed 10_B **SOW2**, two successive Short Open Windows allowed 11_B **SOW3**, three successive Short Open Windows allowed

Field	Bits	Type	Description
WDP_SEL	5:0	rw	**Watchdog Period Selection and trigger** Selects the time for the next Watchdog period and allows to trigger the Short Open Window. 00_H **SOW_TRIG**, trigger short open window 01_H **WP_1**, Watchdog period 16 ms 02_H **WP_2**, Watchdog period 32 ms 03_H **WP_3**, Watchdog period 48 ms 04_H **WP_4**, Watchdog period 64 ms 05_H **WP_5**, Watchdog period 80 ms 06_H **WP_6**, Watchdog period 96 ms 07_H **WP_7**, Watchdog period 112 ms 08_H **WP_8**, Watchdog period 128 ms 09_H **WP_9**, Watchdog period 144 ms $0A_H$ **WP_10**, Watchdog period 160 ms $0B_H$ **WP_11**, Watchdog period 176 ms $0C_H$ **WP_12**, Watchdog period 192 ms $0D_H$ **WP_13**, Watchdog period 208 ms $0E_H$ **WP_14**, Watchdog period 224 ms $0F_H$ **WP_15**, Watchdog period 240 ms 10_H **WP_16**, Watchdog period 256 ms 11_H **WP_17**, Watchdog period 272 ms 12_H **WP_18**, Watchdog period 288 ms 13_H **WP_19**, Watchdog period 304 ms 14_H **WP_20**, Watchdog period 320 ms 15_H **WP_21**, Watchdog period 336 ms 16_H **WP_22**, Watchdog period 352 ms 17_H **WP_23**, Watchdog period 368 ms 18_H **WP_24**, Watchdog period 384 ms 19_H **WP_25**, Watchdog period 400 ms $1A_H$ **WP_26**, Watchdog period 416 ms $1B_H$ **WP_27**, Watchdog period 432 ms $1C_H$ **WP_28**, Watchdog period 448 ms $1D_H$ **WP_29**, Watchdog period 464 ms $1E_H$ **WP_30**, Watchdog period 480 ms $1F_H$ **WP_31**, Watchdog period 496 ms 20_H **WP_32**, Watchdog period 512 ms 21_H **WP_33**, Watchdog period 528 ms 22_H **WP_34**, Watchdog period 544 ms 23_H **WP_35**, Watchdog period 560 ms 24_H **WP_36**, Watchdog period 576 ms 25_H **WP_37**, Watchdog period 592 ms 26_H **WP_38**, Watchdog period 608 ms 27_H **WP_39**, Watchdog period 624 ms 28_H **WP_40**, Watchdog period 640 ms 29_H **WP_41**, Watchdog period 656 ms $2A_H$ **WP_42**, Watchdog period 672 ms $2B_H$ **WP_43**, Watchdog period 688 ms

Field	Bits	Type	Description
			Watchdog Period Selection and trigger (cont'd) $2C_H$ **WP_44**, Watchdog period 704 ms $2D_H$ **WP_45**, Watchdog period 720 ms $2E_H$ **WP_46**, Watchdog period 736 ms $2F_H$ **WP_47**, Watchdog period 752 ms 30_H **WP_48**, Watchdog period 768 ms 31_H **WP_49**, Watchdog period 784 ms 32_H **WP_50**, Watchdog period 800 ms 33_H **WP_51**, Watchdog period 816 ms 34_H **WP_52**, Watchdog period 832 ms 35_H **WP_53**, Watchdog period 848 ms 36_H **WP_54**, Watchdog period 864 ms 37_H **WP_55**, Watchdog period 880 ms 38_H **WP_56**, Watchdog period 896 ms 39_H **WP_57**, Watchdog period 912 ms $3A_H$ **WP_58**, Watchdog period 928 ms $3B_H$ **WP_59**, Watchdog period 944 ms $3C_H$ **WP_60**, Watchdog period 960 ms $3D_H$ **WP_61**, Watchdog period 976 ms $3E_H$ **WP_62**, Watchdog period 992 ms $3F_H$ **WP_63**, Watchdog period 1008 ms

1) Writing $0000\ 0000_H$ to the WDT1_TRIG register will cause a reset.

14 GPIO ports and peripheral I/O

This chapter describes the GPIO Ports of the TLE987x. It contains the following sections:

- Functional description of the GPIO Ports (see **Section 14.2**)
- GPIO Port register descriptions (see **Section 14.3**)
- TLE987x implementation specific details and registers of the GPIO module (see **Section 14.4**)

The TLE987x has 15 port pins organized into three parallel ports: Port 0 (P0), port 1 (P1) and port 2 (P2). Each port pin has a pair of internal pull-up and pull-down devices that can be individually enabled or disabled. P0 and P1 are bidirectional and can be used as general-purpose input/output (GPIO) or to perform alternate input/output functions for the on-chip peripherals. For ports configured as an output, the open drain mode can be selected. On port 2 (P2), analog inputs are shared with general-purpose inputs.

14.1 Features

Features of bidirectional ports (P0, P1)

- Configurable pin direction
- Configurable pull-up/pull-down devices
- Configurable open-drain mode
- Configurable drive strength
- Transfer of data through digital inputs and outputs (general-purpose I/O)
- Alternate input/output for on-chip peripherals

Features of the analog port (P2)

- Configurable pull-up/pull-down devices
- Transfer of data through digital inputs
- Alternate inputs for on-chip peripherals

14.2 Introduction

14.2.1 Port 0 and port 1

Figure 73 shows the block diagram of a TLE987x bidirectional port pin. Each port pin is equipped with a number of control and data bits, thus enabling very flexible usage of the pin. By defining the contents of the control register, each individual pin can be configured as an input or an output. The user can also configure each pin as an open-drain pin with or without an internal pull-up/pull-down.

Each bidirectional port pin can be configured for input or output operation. Switching between input and output mode is accomplished through the register Px_DIR (x = 0 or 1), which enables or disables the output and input drivers. A port pin can only be configured as either input or output at any one time.

In input mode (default after reset), the output driver is switched off (high-impedance). The voltage level present at the port pin is translated into a logical 0 or 1 via a Schmitt trigger device and can be read via the register Px_DATA.

In output mode, the output driver is activated and drives the value supplied through the multiplexer to the port pin. In the output driver, each port line can be switched to open-drain mode or normal mode (push-pull mode) via the register Px_OD.

The output multiplexer in front of the output driver enables the port output function to be used for different purposes. If the pin is used for general-purpose output, the multiplexer is switched by software to the data register Px_DATA. Software can set or clear the bit in Px_DATA and therefore directly influence the state of the port pin. If an on-chip peripheral uses the pin for output signals, alternate output lines (AltDataOut) can be switched via the multiplexer to the output driver circuitry. Selection of the alternate output function is defined in registers Px_ALTSEL0 and Px_ALTSEL1. When a port pin is used in an alternate function, its direction must be set accordingly in the register Px_DIR.

Each pin can also be programmed to activate an internal weak pull-up or pull-down device. Register Px_PUDSEL selects whether a pull-up or pull-down device is activated, while register Px_PUDEN enables or disables the pull device.

The port structure used in this device offers the possibility to select the output driver strength and the slew rate. These selections are independent from the output port functionality, such as open-drain, push/pull or input only. The driver strength for each pin can be adapted to the application requirements by registers Px_POCONy (y = 0, 1 or 2) in SCU.

The temperature compensation signals TC[1:0] of all output drivers are connected to all outputs and are controlled by register TCCR in SCU.

*Note: For the definition of Px_POCONy and TCCR registers, refer to **Chapter 6.7** of SCU chapter.*

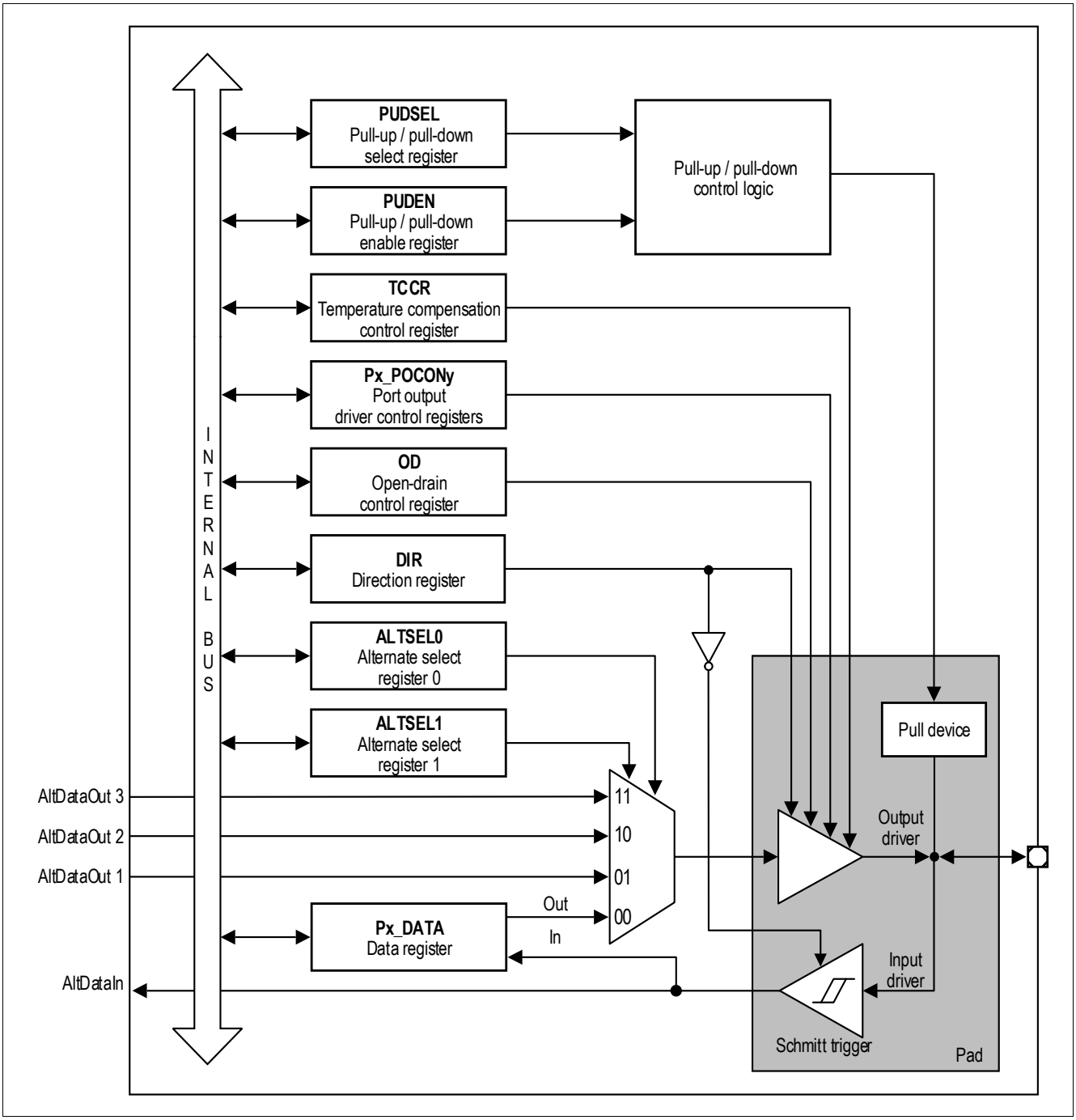


Figure 73 General structure of a bidirectional port (P0, P1)

14.2.2 Port 2

Figure 74 shows the structure of an input-only port pin. Each P2 pin can only function in input mode. Register P2_DIR is provided to enable or disable the input driver. When the input driver is enabled, the actual voltage level present at the port pin is translated into a logic 0 or 1 via a Schmitt trigger device and can be read via register P2_DATA. Each pin can also be programmed to activate an internal weak pull-up or pull-down device. Register P2_PUDSEL selects whether a pull-up or the pull-down device is activated, while register P2_PUDEN enables or disables the pull device. The analog input (AnalogIn) bypasses the digital circuitry and Schmitt trigger device for direct feed-through to the ADC input channels.

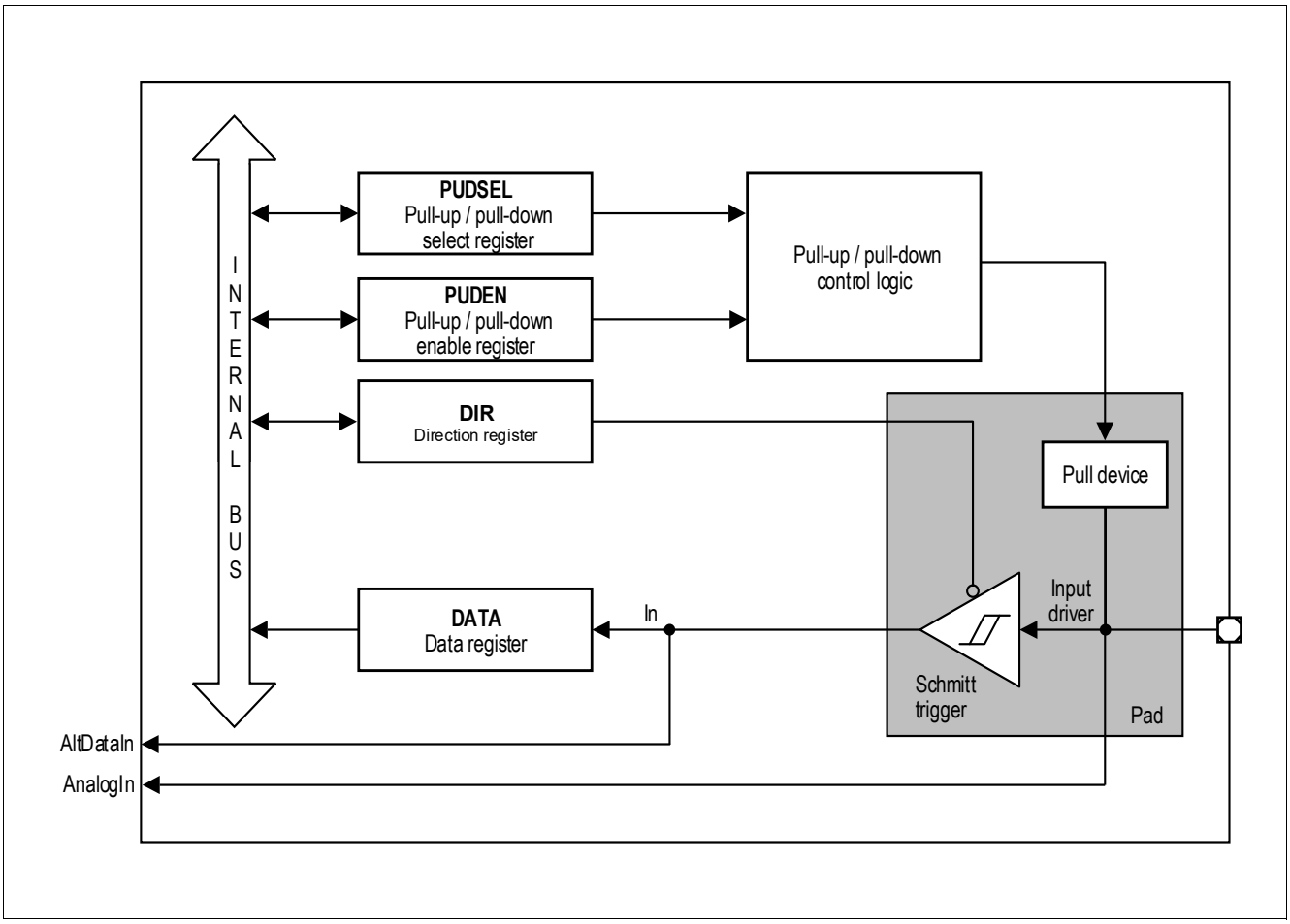


Figure 74 General structure of input port (P2)

14.3 General Port Register Description

Each port consists of 8-bit control and data registers. The registers are defined in **Figure 75**.

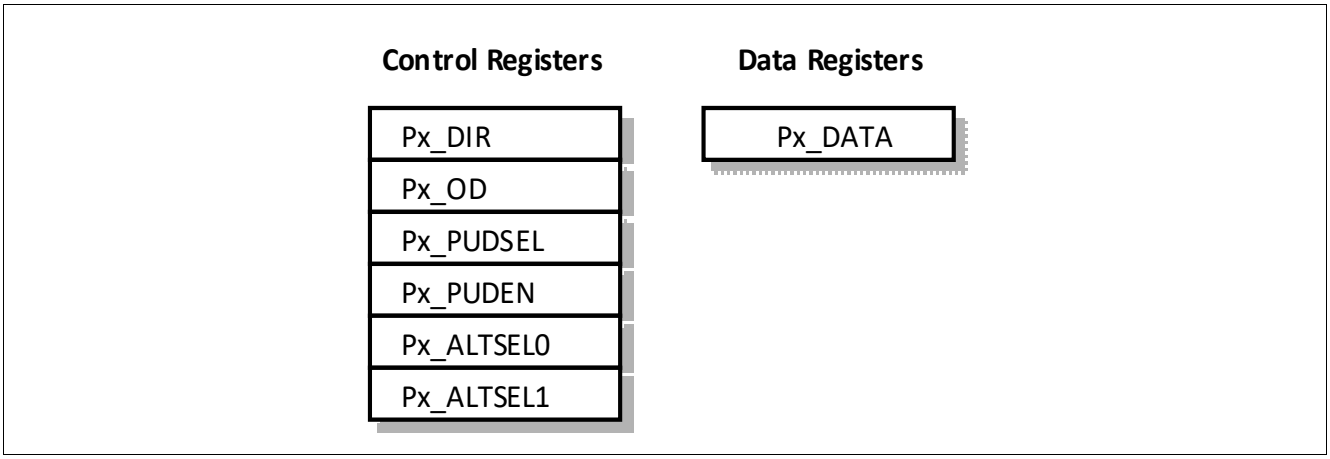


Figure 75 Port Registers

Table 77 Port Registers

Register Short Name	Register Long Name	Page Number
Px_DATA	Port x Data Register	395
Px_DIR	Port x Direction Register	395
Px_OD	Port x Open Drain Control Register	396
Px_PUDSEL	Port x Pull-Up/Pull-Down Select Register	397
Px_PUDEN	Port x Pull-Up/Pull-Down Enable Register	397
Px_ALTSEL0	Port x Alternate Select Register 0	398
Px_ALTSEL1	Port x Alternate Select Register 1	398

Note: Not all the registers are implemented for each port.

Note: The Px_DATA registers are not read-modify-write save.

14.3.1 Port Data Register

If a port pin is used as general purpose output, output data is written into register Px_DATA of port x. When the port pin is used as general purpose input, the value at a port pin can be read through the register Px_DATA. The data register Px_DATA always contains a latched value of the assigned port pin.

Px_DATA
Port x Data Register **Reset Value: XX_H**

7	6	5	4	3	2	1	0
P7	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
Pn **(n = 0 - 7)**	n	rwh	**Portx Pin n Data Value** 0 Port x pin n data value = 0 1 Port x pin n data value = 1

Bit Px_DATA.n can only be written if the corresponding pin is set to output, i.e. Px_DIR.n = 1. The contents of Px_DATA.n are output on the assigned pin if the pin is assigned as GPIO pin and the direction is switched/set to output. A read operation of Px_DATA returns the register value and not the state of the Px_DATA pins.

14.3.2 Direction Register

The direction of bidirectional port pins is controlled by the respective direction register Px_DIR. For input-only port pins, register Px_DIR is used to enable or disable the input drivers.

Px_DIR
Port x Direction Register **Reset Value: XX_H**

7	6	5	4	3	2	1	0
P7	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn **(n = 0 - 7)**	n	rw	**Bidirectional: Port x Pin n Direction Control** 0 Direction is set to input (default) 1 Direction is set to output or **Input-only: Port x Pin n Driver Control** 0 Input driver is enabled (default) 1 Input driver is disabled

14.3.3 Open Drain Control Register

Each pin in output mode can be switched to Open Drain Mode. If driven with 1, no driver will be activated and the pin output state depends on the internal pull-up/pull-down device setting; if driven with 0, the driver's pull-down transistor will be activated.

The open drain mode is controlled by the register Px_OD.

Px_OD
Port x Open Drain Control Register **Reset Value: XX_H**

7	6	5	4	3	2	1	0
P7	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Py **(y = 0 - 7)**	y	rw	**Port x Pin y Open Drain Mode** 0 Normal Mode, output is actively driven for 0 and 1 state (default) 1 Open Drain Mode, output is actively driven only for 0 state

14.3.4 Pull-Up/Pull-Down Device Register

Internal pull-up/pull-down devices can be optionally applied to a port pin. This offers the possibility to configure the following input characteristics:

- tristate
- high-impedance with a weak pull-up device
- high-impedance with a weak pull-down device

and the following output characteristics:

- push/pull (optional pull-up/pull-down)
- open drain with internal pull-up
- open drain with external pull-up

The pull-up/pull-down device can be fixed or controlled via the registers Px_PUDSEL and Px_PUDEN. Register Px_PUDSEL selects the type of pull-up/pull-down device, while register Px_PUDEN enables or disables it. The pull-up/pull-down device can be selected pinwise.

Note: The selected pull-up/pull-down device is enabled by setting the respective bit in the Px_PUDEN register.

Px_PUDSEL
Port x Pull-Up/Pull-Down Select Register **Reset Value: XX_H**

7	6	5	4	3	2	1	0
P7	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Py **(y = 0 - 7)**	y	rw	**Pull-Up/Pull-Down Select Port x Bit y** 0 Pull-down device is selected 1 Pull-up device is selected

Px_PUDEN
Port x Pull-Up/Pull-Down Enable Register **Reset Value: XX_H**

7	6	5	4	3	2	1	0
P7	**P6**	**P5**	**P4**	**P3**	**P2**	**P1**	**P0**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Py **(y = 0 - 7)**	y	rw	**Pull-Up/Pull-Down Enable at Port x Bit y** 0 Pull-up or Pull-down device is disabled 1 Pull-up or Pull-down device is enabled

14.3.5 Alternate Input Functions

The number of alternate functions that uses a pin for input is not limited. Each port control logic of an I/O pin provides several input paths:

- Digital input value via register
- Direct digital input value

14.3.6 Alternate Output Functions

Alternate functions are selected via an output multiplexer which can select up to four output lines. This multiplexer can be controlled by the following signals:

- Register Px_ALTSEL0
- Register Px_ALTSEL1

Selection of alternate functions is defined in registers Px_ALTSEL0 and Px_ALTSEL1.

Px_ALTSELy (y = 0, 1)
Port x Alternate Select Register **Reset Value: XX_H**

7	6	5	4	3	2	1	0
0	**0**	**0**	**P4**	**P3**	**P2**	**P1**	**P0**
rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Pn (n = 0 - 4)	n	rw	See **Table 78**
0	5,6,7	rw	

Table 78 Function of Bits Px_ALTSEL0.Pn and Px_ALTSEL1.Pn

Px_ALTSEL0.Pn	Px_ALTSEL1.Pn	Function
0	0	Normal GPIO
1	0	Alternate Select 1
0	1	Alternate Select 2
1	1	Alternate Select 3

14.3.7 Register Map

Table 79 shows the Ports module base addresses.

Table 80 lists the addresses of the Ports SFRs and SCU SFRs.

Table 79 Register Address Space

Module	Base Address	End Address	Note
PORT	48028000_H	$48029FFF_H$	
SCU	50005000_H	$50005FFF_H$	

Table 80 Register Overview Ports

Register Short Name	Register Long Name	Offset Address	Page Number
P0_DATA	Port 0 Data Register	00_H	**403**
P0_DIR	Port 0 Direction Register	04_H	**403**
P1_DATA	Port 1 Data Register	08_H	**410**
P1_DIR	Port 1 Direction Register	$0C_H$	**410**
P2_DATA	Port 2 Data Register	10_H	**416**
P2_DIR	Port 2 Direction Register	14_H	**417**
P0_PUDSEL	Port 0 Pull-Up/Pull-Down Select Register	18_H	**404**
P0_PUDEN	Port 0 Pull-Up/Pull-Down Enable Register	$1C_H$	**405**
P1_PUDSEL	Port 1 Pull-Up/Pull-Down Select Register	20_H	**412**
P1_PUDEN	Port 1 Pull-Up/Pull-Down Enable Register	24_H	**412**
P2_PUDSEL	Port 2 Pull-Up/Pull-Down Select Register	28_H	**418**
P2_PUDEN	Port 2 Pull-Up/Pull-Down Enable Register	$2C_H$	**419**
P0_ALTSELy (y = 0-1)	Port 0 Alternate Select Register 0	30_H	**406**
P0_ALTSELy (y = 0-1)	Port 0 Alternate Select Register 1	34_H	**406**
P1_ALTSELy (y = 0-1)	Port 1 Alternate Select Register 0	38_H	**413**
P1_ALTSELy (y = 0-1)	Port 1 Alternate Select Register 1	$3C_H$	**413**
P0_OD	Port 0 Open Drain Control Register	40_H	**404**
P1_OD	Port 1 Open Drain Control Register	44_H	**411**

Table 81 Register Overview SCU Module

Register Short Name	Register Long Name	Offset Address	Page Number
P0_POCON0	P0 Control Register	$0E8_H$	**151**
P0_POCON1	P0 Control Register	$0EC_H$	**152**
P0_POCON2	P0 Control Register	$0F0_H$	**153**
P1_POCON0	P1 Control Register	$0F8_H$	**154**
P1_POCON1	P1 Control Register	$0FC_H$	**155**
P1_POCON2	P1 Control Register	100_H	**156**

14.4 TLE987x port module

14.4.1 Port 0

14.4.1.1 Overview

Port 0 is a general purpose bidirectional port. The port registers of Port 0 are shown in **Table 76**.

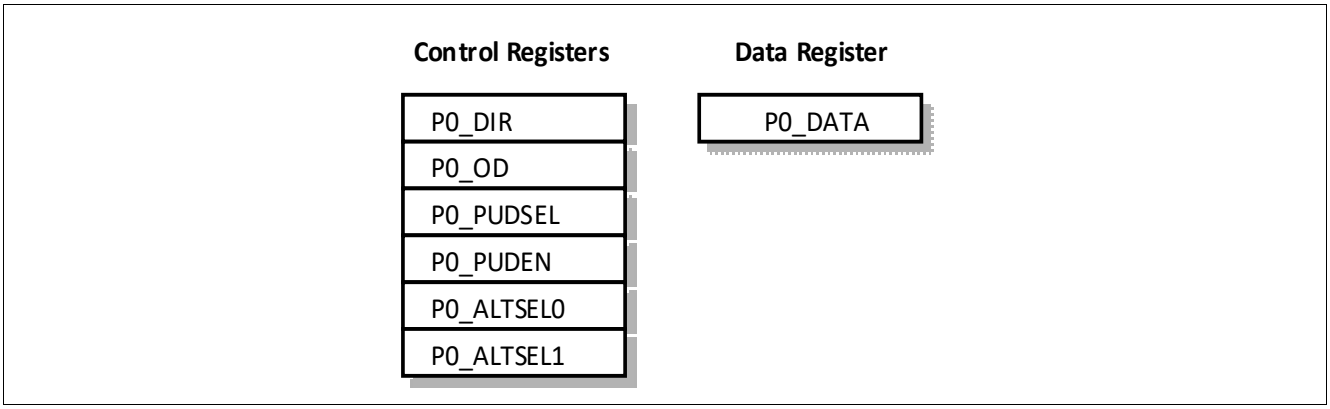


Figure 76 Port 0 Registers

Table 82 Port 0 Registers

Register Short Name	Register Long Name
P0_DATA	Port 0 Data Register
P0_DIR	Port 0 Direction Register
P0_OD	Port 0 Open Drain Control Register
P0_PUDSEL	Port 0 Pull-Up/Pull-Down Select Register
P0_PUDEN	Port 0 Pull-Up/Pull-Down Enable Register
P0_ALTSEL0	Port 0 Alternate Select Register 0
P0_ALTSEL1	Port 0 Alternate Select Register 1

14.4.1.2 Port 0 functions

Table 83 Port 0 input/output functions

Port pin	Input/output	Select	Connected signals	From/to module
P0.0	Input	GPI	P0_DATA.P0	–
		INP1	SWCLK / TCK_0	SW
		INP2	T12HR_0	CCU6
		INP3	T4INA	GPT12T4
		INP4	T2_0	Timer2
		INP5	–	–
		INP6	EXINT2_3	SCU
	Output	GPO	P0_DATA.P0	–
		ALT1	T3OUT	GPT12T3
		ALT2	EXF21_0	Timer21
		ALT3	RXDO_2	UART2
P0.1	Input	GPI	P0_DATA.P1	–
		INP1	T13HR_0	CCU6
		INP2	TxD1	LIN_TxD
		INP3	CAPINA	GPT12CAP
		INP4	T21_0	Timer21
		INP5	T4INC	GPT12T4
		INP6	MRST_1_2	SSC1
		INP7	EXINT0_2	SCU
	Output	GPO	P0_DATA.P1	–
		ALT1	TxD1	UART1 / LIN_TxD
		ALT2	–	–
		ALT3	T6OUT	GPT12T6

Table 83 Port 0 input/output functions (cont'd)

Port pin	Input/output	Select	Connected signals	From/to module
P0.2	Input	GPI	P0_DATA.P2	–
		INP1	CCPOS2_1	CCU6
		INP2	T2EUDA	GPT12T2
		INP3	MTSR_1	SSC1
		INP4	T21EX_0	Timer21
		INP5	T6INA	GPT12T6
	Output	GPO	P0_DATA.P2	–
		ALT1	COUT60_0	CCU6
		ALT2	MTSR_1	SSC1
		ALT3	EXF2_0	Timer2
P0.3	Input	GPI	P0_DATA.P3	–
		INP1	SCK_1	SSC1
		INP2	CAPINB	GPT12
		INP3	T5INA	GPT12T5
		INP4	T4EUDA	GPT12T4
		INP5	CCPOS0_1	CCU6
	Output	GPO	P0_DATA.P3	–
		ALT1	SCK_1	SSC1
		ALT2	EXF21_2	Timer21
		ALT3	T6OUT	GPT12T6
P0.4	Input	GPI	P0_DATA.P4	–
		INP1	MRST_1_0	SSC1
		INP2	CC60_0	CCU6
		INP3	T21_2	Timer21
		INP4	EXINT2_2	SCU
		INP5	T3EUDA	GPT12T3
		INP6	CCPOS1_1	CCU6
	Output	GPO	P0_DATA.P4	–
		ALT1	MRST_1_0	SSC1
		ALT2	CC60_0	CCU6
		ALT3	CLKOUT_0	SCU

14.4.1.3 Port 0 Register Description

Data Register

P0_DATA
Port 0 Data Register **(00_H)** **Reset Value: XX_H**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rwh	**Port 0 Pin x Data Value** 0_B Port 0 pin x data value = 0 1_B Port 0 pin x data value = 1
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Direction Register

P0_DIR
Port 0 Direction Register **(04_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Port 0 Pin x Direction Control** 0_B Direction is set to input (default) 1_B Direction is set to output
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Open Drain Control Register

P0_OD
Port 0 Open Drain Control Register **(40_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Port 0 Pin x Open Drain Mode** 0_B Normal Mode, output is actively driven for 0 and 1 state (default) 1_B Open Drain Mode, output is actively driven only for 0 state
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Pull-Up/Pull-Down Device Register

P0_PUDSEL
Port 0 Pull-Up/Pull-Down Select Register **(18_H)** **Reset Value: $0B_H$**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Pull-Up/Pull-Down Select Port 0 Bit x** 0_B Pull-down device is selected 1_B Pull-up device is selected
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

P0_PUDEN
Port 0 Pull-Up/Pull-Down Enable Register **($1C_H$)** **Reset Value: $1F_H$**

7	6	5	4	3	2	1	0
RES			**P4**	**P3**	**P2**	**P1**	**P0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Pull-Up/Pull-Down Enable at Port 0 Bit x** 0_B Pull-up or Pull-down device is disabled 1_B Pull-up or Pull-down device is enabled
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Alternate Output Select Register

P0_ALTSELy (y = 0-1)
Port 0 Alternate Select Register **(30_H+y*4_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Alternate Select Port 0 Bit x** See **Table 84**
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Table 84 Function of Bits P0_ALTSEL0.Px and P0_ALTSEL1.Px

P0_ALTSEL0.Px	P0_ALTSEL1.Px	Function
0	0	Normal GPIO
1	0	Alternate Select 1
0	1	Alternate Select 2
1	1	Alternate Select 3

14.4.2 Port 1

14.4.2.1 Overview

Port 1 is a general purpose bidirectional port. The port registers of Port 1 are shown in **Table 77**.

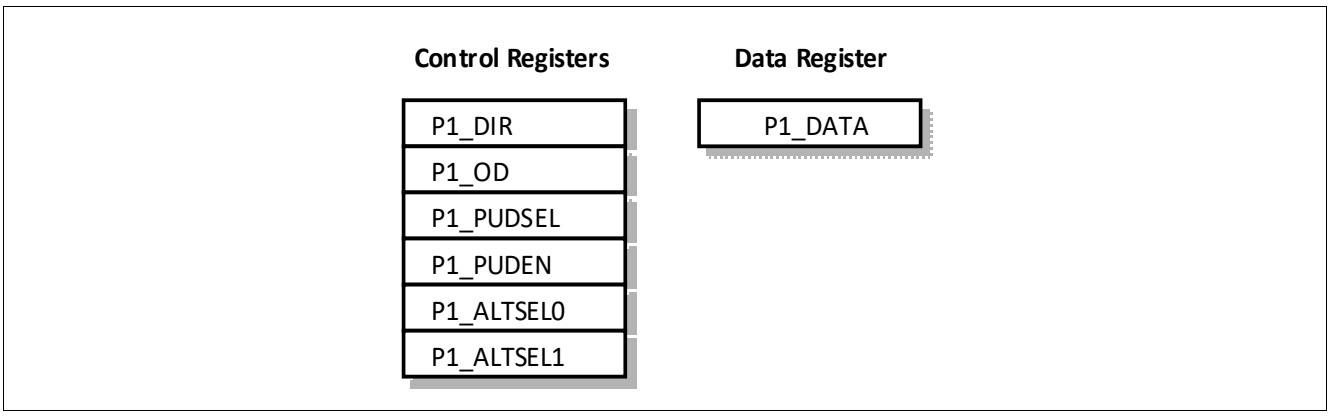


Figure 77 Port 1 Registers

Table 85 Port 1 Registers

Register Short Name	Register Long Name
P1_DATA	Port 1 Data Register
P1_DIR	Port 1 Direction Register
P1_OD	Port 1 Open Drain Control Register
P1_PUDSEL	Port 1 Pull-Up/Pull-Down Select Register
P1_PUDEN	Port 1 Pull-Up/Pull-Down Enable Register
P1_ALTSEL0	Port 1 Alternate Select Register 0
P1_ALTSEL1	Port 1 Alternate Select Register 1

14.4.2.2 Port 1 functions

Table 86 Port 1 input/output functions

Port pin	Input/output	Select	Connected signals	From/to module
P1.0	Input	GPI	P1_DATA.P0	–
		INP1	T3INC	GPT12T3
		INP2	T4EUDB	GPT12T4
		INP3	CC61_0	CCU6
		INP4	SCK_2	SSC2
		INP5	EXINT1_2	SCU
	Output	GPO	P1_DATA.P0	–
		ALT1	SCK_2	SSC2
		ALT2	CC61_0	CCU6
		ALT3	EXF21_3	Timer21

Table 86 Port 1 input/output functions (cont'd)

Port pin	Input/output	Select	Connected signals	From/to module
P1.1	Input	GPI	P1_DATA.P1	–
		INP1	–	–
		INP2	T6EUDA	GPT12T6
		INP3	–	–
		INP4	MTSR_2	SSC2
		INP5	T21_1	Timer21
		INP6	EXINT1_0	SCU
	Output	GPO	P1_DATA.P1	–
		ALT1	MTSR_2	SSC2
		ALT2	COUT61_0	CCU6
		ALT3	TXD2_0	UART2
P1.2	Input	GPI	P1_DATA.P2	–
		INP1	T2INA	GPT12T2
		INP2	T2EX_1	Timer2
		INP3	T21EX_3	Timer21
		INP4	MRST_2_0	SSC2
		INP5	RXD2_0	UART2
		INP6	CCPOS2_2	CCU6
		INP7	EXINT0_1	SCU
	Output	GPO	P1_DATA.P2	–
		ALT1	MRST_2_0	SSC2
		ALT2	COUT63_0	CCU6
		ALT3	T3OUT	GPT12T3

Table 86 Port 1 input/output functions (cont'd)

Port pin	Input/output	Select	Connected signals	From/to module
P1.3	Input	GPI	P1_DATA.P3	–
		INP1	T6INB	GPT12T6
		INP2	–	–
		INP3	CC62_0	CCU6
		INP4	T6EUDB	GPT12T6
		INP5	–	–
		INP6	CCPOS0_2	CCU6
		INP7	EXINT1_1	SCU
	Output	GPO	P1_DATA.P3	–
		ALT1	EXF21_1	Timer21
		ALT2	CC62_0	CCU6
		ALT3	TXD2_1	UART2
P1.4	Input	GPI	P1_DATA.P4	–
		INP1	EXINT2_1	SCU
		INP2	T21EX_1	Timer21
		INP3	T5EUDA	GPT12T5
		INP4	RxD1	UART1
		INP5	T2INB	GPT12T2
		INP6	CCPOS1_2	CCU6
		INP7	MRST_1_3	SSC1
	Output	GPO	P1_DATA.P4	–
		ALT1	CLKOUT_1	SCU
		ALT2	COUT62_0	CCU6
		ALT3	RxD1	UART1 / LIN_RxD

14.4.2.3 Port 1 Register Description

Data Register

P1_DATA
Port 1 Data Register **(08_H)** **Reset Value: XX_H**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rwh	**Port 1 Pin n Data Value** 0_B Port 1 pin x data value = 0 1_B Port 1 pin x data value = 1
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Direction Register

P1_DIR
Port 1 Direction Register **($0C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
	RES		**P4**	**P3**	**P2**	**P1**	**P0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Port 1 Pin x Direction Control** 0_B Direction is set to input (default) 1_B Direction is set to output
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Open Drain Control Register

P1_OD
Port 1 Open Drain Control Register **(44_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**P4**	**P3/P2**	**P1**	**P0**	**RES**
r			rw	rw	rw	rw	r

Field	Bits	Type	Description
RES	0	r	reserved
P0	1	rw	**P1.0 Open Drain Mode** 0_B Normal Mode, output is actively driven for 0 and 1 state (default) 1_B Open Drain Mode, output is actively driven only for 0 state
P1	2	rw	**P1.1 Open Drain Mode** 0_B Normal Mode, output is actively driven for 0 and 1 state (default) 1_B Open Drain Mode, output is actively driven only for 0 state
P3/P2	3	rw	**P1.3 and P1.2 Open Drain Mode** 0_B Normal Mode, output is actively driven for 0 and 1 state (default) 1_B Open Drain Mode, output is actively driven only for 0 state
P4	4	rw	**P1.4 Open Drain Mode** 0_B Normal Mode, output is actively driven for 0 and 1 state (default) 1_B Open Drain Mode, output is actively driven only for 0 state
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Pull-Up/Pull-Down Device Register

P1_PUDSEL
Port 1 Pull-Up/Pull-Down Select Register **(20_H)** **Reset Value: $1F_H$**

7	6	5	4	3	2	1	0
RES			**P4**	**P3**	**P2**	**P1**	**P0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Pull-Up/Pull-Down Select Port 1 Bit x** 0_B Pull-down device is selected 1_B Pull-up device is selected (default)
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

P1_PUDEN
Port 1 Pull-Up/Pull-Down Enable Register **(24_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**P4**	**P3**	**P2**	**P1**	**P0**
r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Pull-Up/Pull-Down Enable at Port 1 Bit x** 0_B Pull-up or Pull-down device is disabled (default) 1_B Pull-up or Pull-down device is enabled
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Alternate Output Select Register

P1_ALTSELy (y = 0-1)
Port 1 Alternate Select Register **(38_H+y*4_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES			**P4**	**P3**	**P2**	**P1**	**P0**
	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Px **(x = 0 - 4)**	x	rw	**Alternate Select Port 1 Bit x** See **Table 87**
RES	[7:5]	r	**Reserved** Returns 0 if read; should be written with 0.

Table 87 Function of Bits P1_ALTSEL0.Px and P1_ALTSEL1.Px

P1_ALTSEL0.Px	P1_ALTSEL1.Px	Function
0	0	Normal GPIO
1	0	Alternate Select 1
0	1	Alternate Select 2
1	1	Alternate Select 3

14.4.3 Port 2

14.4.3.1 Overview

Port 2 is a general purpose input-only port. The port registers of Port 2 are shown in **Table 78**.

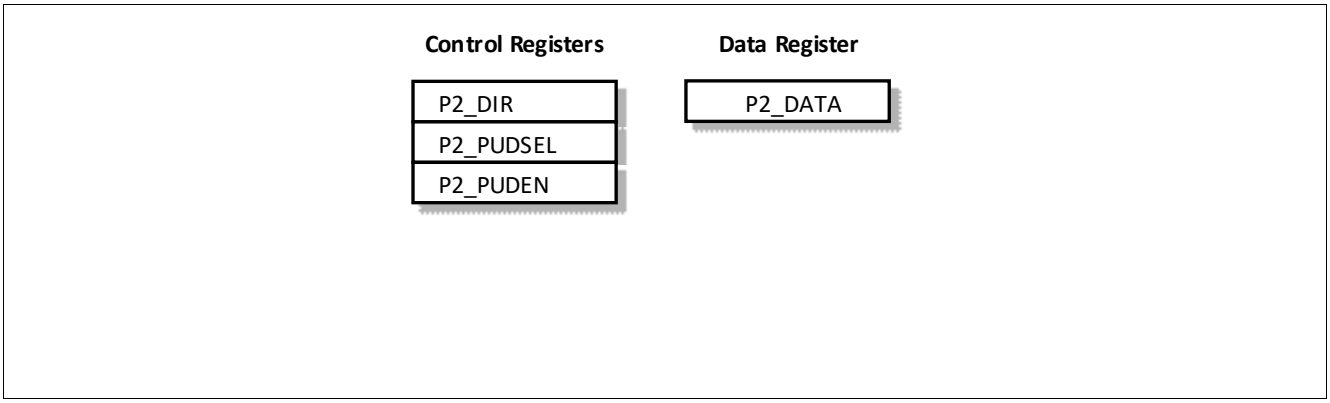


Figure 78 Port 2 Registers

Table 88 Port 2 Registers

Register Short Name	Register Long Name
P2_DATA	Port 2 Data Register
P2_DIR	Port 2 Direction Register
P2_PUDSEL	Port 2 Pull-Up/Pull-Down Select Register
P2_PUDEN	Port 2 Pull-Up/Pull-Down Enable Register

14.4.3.2 Port 2 functions

Table 89 Port 2 input functions

Port pin	Input/output	Select	Connected signals	From/to module
P2.0	Input	GPI	P2_DATA.P0	–
		INP1	CCPOS0_3	CCU6
		INP2	-	–
		INP3	T12HR_2	CCU6
		INP4	EXINT0_0	SCU
		INP5	CC61_2	CCU6
		ANALOG	AN0	ADC1
			XTAL (in)	XTAL[1)]
		IN	ADC3.P	SDADC[2)]

Table 89 Port 2 input functions (cont'd)

Port pin	Input/output	Select	Connected signals	From/to module
P2.2	Input	GPI	P2_DATA.P2	–
		INP1	CCPOS2_3	CCU6
		INP2	T13HR_2	CCU6
		INP3	–	–
		INP4	CC62_2	CCU6
		ANALOG	AN2	ADC1
		OUT	XTAL (out)	XTAL[1)]
		IN	ADC3.N	SDADC[2)]
P2.3	Input	GPI	P2_DATA.P3	–
		INP1	CCPOS1_0	CCU6
		INP2	$\overline{\text{CTRAP\#_1}}$	CCU6
		INP3	T21EX_2	Timer21
		INP4	CC60_1	CCU6
		INP5	EXINT0_3	SCU
		ANALOG	AN3	ADC1
P2.4	Input	GPI	P2_DATA.P4	–
		INP1	$\overline{\text{CTRAP\#_0}}$	CCU6
		INP2	T2EUDB	GPT12T2
		INP3	MRST_1_1	SSC1
		INP4	EXINT1_3	SCU
		ANALOG	AN4	ADC1
		IN	ADC4.P	SDADC[2)]
P2.5	Input	GPI	P2_DATA.P5	–
		INP1	RXD2_1	UART2
		INP2	T3EUDB	GPT12T3
		INP3	MRST_2_1	SSC2
		INP4	T2_1	Timer2
		ANALOG	AN5	ADC1
		IN	ADC4.N	SDADC[2)]

1) Available only for products with XTAL support. See the datasheet for details.
2) Available only for products with SDADC. See the datasheet for details.

14.4.3.3 Port 2 Register Description

Data Register

P2_DATA
Port 2 Data Register **(10_H)** **Reset Value: XX_H**

7	6	5	4	3	2	1	0
RES	**RES**	**P5**	**P4**	**P3**	**P2**	**RES**	**P0**
rh	r	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
P0	0	rh	**Port 2 Pin 0 Data Value** 0_B P2.0 data value = 0 1_B P2.0 data value = 1
RES	1	rh	**Reserved** Returns 0 if read
P2	2	rh	**Port 2 Pin 2 Data Value** 0_B P2.2 data value = 0 1_B P2.2 data value = 1
P3	3	rh	**Port 2 Pin 3 Data Value** 0_B P2.3 data value = 0 1_B P2.3 data value = 1
P4	4	rh	**Port 2 Pin 4 Data Value** 0_B P2.4 data value = 0 1_B P2.4 data value = 1
P5	5	rh	**Port 2 Pin 5 Data Value** 0_B P2.5 data value = 0 1_B P2.5 data value = 1
RES	6	r	**Reserved** Returns 0 if read
RES	7	rh	**Reserved** Returns 0 if read

Direction Register

This register is being initialized by the BootROM during the boot-up process to a value of $8F_H$.

P2_DIR
Port 2 Direction Register **(14_H)** **Reset Value: BF_H**

7	6	5	4	3	2	1	0
RES	**RES**	**P5**	**P4**	**P3**	**P2**	**RES**	**P0**
rw	r	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
P0	0	rw	**Port 2 Pin 0 Input Control** 0_B Input receiver enabled 1_B Input receiver disabled
RES	1	rw	**Reserved** Returns 0 if read; should be written with 0.
P2	2	rw	**Port 2 Pin 2 Input Control** 0_B Input receiver enabled 1_B Input receiver disabled
P3	3	rw	**Port 2 Pin 3 Input Control** 0_B Input receiver enabled 1_B Input receiver disabled
P4	4	rw	**Port 2 Pin 4 Input Control** 0_B Input receiver enabled 1_B Input receiver disabled
P5	5	rw	**Port 2 Pin 5 Input Control** 0_B Input receiver enabled 1_B Input receiver disabled
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.
RES	7	rw	**Reserved** Returns 0 if read; should be written with 0.

Pull-Up/Pull-Down Device Register

P2_PUDSEL
Port 2 Pull-Up/Pull-Down Select Register **(28_H)** **Reset Value: BF_H**

7	6	5	4	3	2	1	0
RES	**RES**	**P5**	**P4**	**P3**	**P2**	**RES**	**P0**
r	r	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
P0	0	rw	**Port 2 Pin 0 Pull-Up/Pull-Down select** 0_B Pull-Down selected 1_B Pull-Up selected
RES	1	rw	**Reserved** Returns 0 if read; should be written with 0.
P2	2	rw	**Port 2 Pin 2 Pull-Up/Pull-Down select** 0_B Pull-Down selected 1_B Pull-Up selected
P3	3	rw	**Port 2 Pin 3 Pull-Up/Pull-Down select** 0_B Pull-Down selected 1_B Pull-Up selected
P4	4	rw	**Port 2 Pin 4 Pull-Up/Pull-Down select** 0_B Pull-Down selected 1_B Pull-Up selected
P5	5	rw	**Port 2 Pin 5 Pull-Up/Pull-Down select** 0_B Pull-Down selected 1_B Pull-Up selected
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.

P2_PUDEN
Port 2 Pull-Up/Pull-Down Enable Register **($2C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**RES**	**P5**	**P4**	**P3**	**P2**	**RES**	**P0**
r	r	rw	rw	rw	rw	r	rw

Field	Bits	Type	Description
P0	0	rw	**Port 2 Pin 0 Pull-Up/Pull-Down Device enable** 0_B Pull-Device disabled 1_B Pull-Device enabled
RES	1	r	**Reserved** Returns 0 if read; should be written with 0.
P2	2	rw	**Port 2 Pin 2 Pull-Up/Pull-Down Device enable** 0_B Pull-Device disabled 1_B Pull-Device enabled
P3	3	rw	**Port 2 Pin 3 Pull-Up/Pull-Down Device enable** 0_B Pull-Device disabled 1_B Pull-Device enabled
P4	4	rw	**Port 2 Pin 4 Pull-Up/Pull-Down Device enable** 0_B Pull-Device disabled 1_B Pull-Device enabled
P5	5	rw	**Port 2 Pin 5 Pull-Up/Pull-Down Device enable** 0_B Pull-Device disabled 1_B Pull-Device enabled
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.

14.5 Alternate Function Map

The following figure displays the mapping of the alternate function of the GPIOs.

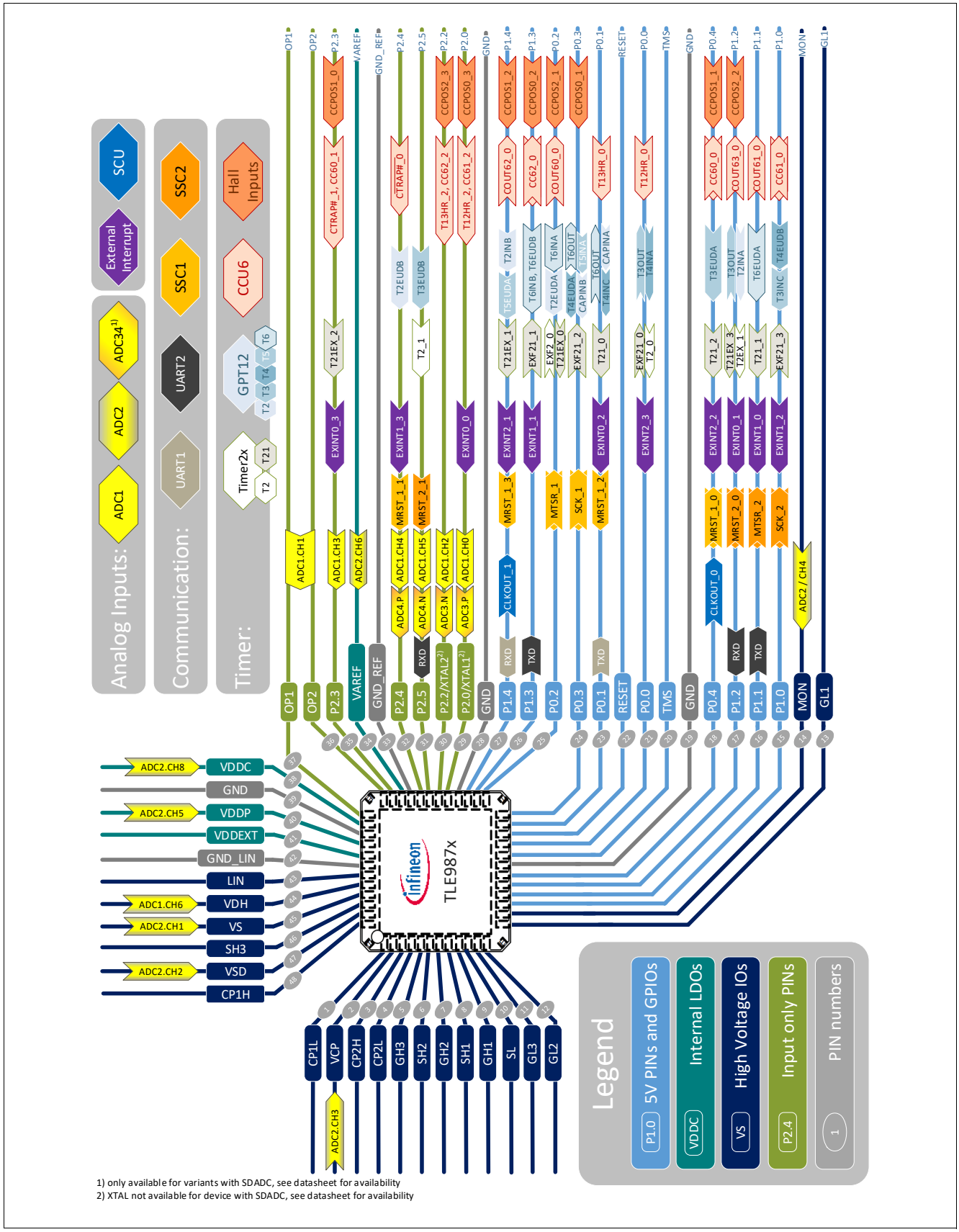


Figure 79 Alternate Function Map

15 General-purpose timer units (GPT12)

15.1 Features

15.1.1 Features of block GPT1

The following list summarizes the supported features:

- f_{GPT} is derived from PCLK
- $f_{GPT}/4$ maximum resolution
- 3 independent timers/counters
- Timers/counters can be concatenated
- 4 Operating modes:
  - Timer mode
  - Gated Timer mode
  - Counter mode
  - Incremental Interface mode
- Reload and capture functionality
- Shared interrupt: node 0

15.1.2 Features of block GPT2

The following list summarizes the supported features:

- f_{GPT} is derived from PCLK
- $f_{GPT}/2$ maximum resolution
- 2 independent timers/counters
- Timers/counters can be concatenated
- 3 Operating modes:
  - Timer mode
  - Gated Timer mode
  - Counter mode
- Extended capture/reload functions via 16-bit capture/reload register CAPREL
- Shared interrupt: node 1

15.2 Introduction

The general-purpose timer unit blocks GPT1 and GPT2 have very flexible multifunctional timer structures which may be used for timing, event counting, pulse-width measurement, pulse generation, frequency multiplication, and other purposes.

They incorporate five 16-bit timers that are grouped into the two timer blocks GPT1 and GPT2. Each timer in each block may operate independently in a number of different modes such as Gated Timer or Counter mode, or may be concatenated with another timer of the same block.

Each block has alternate input/output functions and specific interrupts associated with it. Input signals can be selected from several sources through the PISEL register.

The GPT module is clocked with clock f_{GPT}. f_{GPT} is a clock derived from PCLK.

15.2.1 Block diagram of GPT1

The GPT1 block contains three timers/counters: The core timer T3 and the two auxiliary timers T2 and T4. The maximum resolution is $f_{GPT}/4$. The auxiliary timers of GPT1 may optionally be configured as reload or capture registers for the core timer. These registers are listed in **Section 15.3.7.1**.

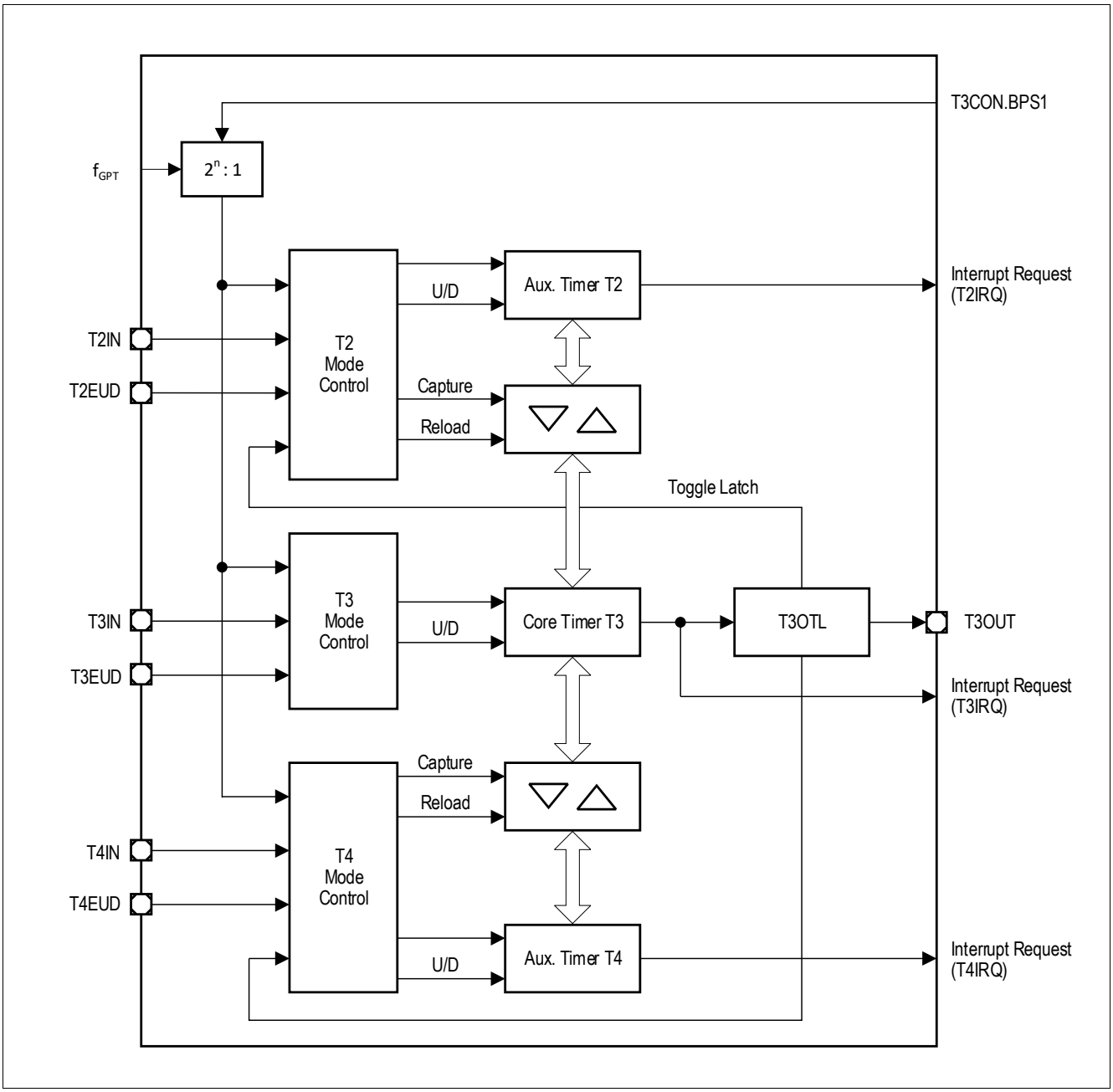


Figure 80 GPT1 block diagram

15.2.2 Block diagram of GPT2

The GPT2 block contains two timers/counters: The core timer T6 and the auxiliary timer T5. The maximum resolution is $f_{GPT}/2$. An additional capture/reload register (CAPREL) supports capture and reload operation with extended functionality. These registers are listed in **Section 15.4.8.1**.

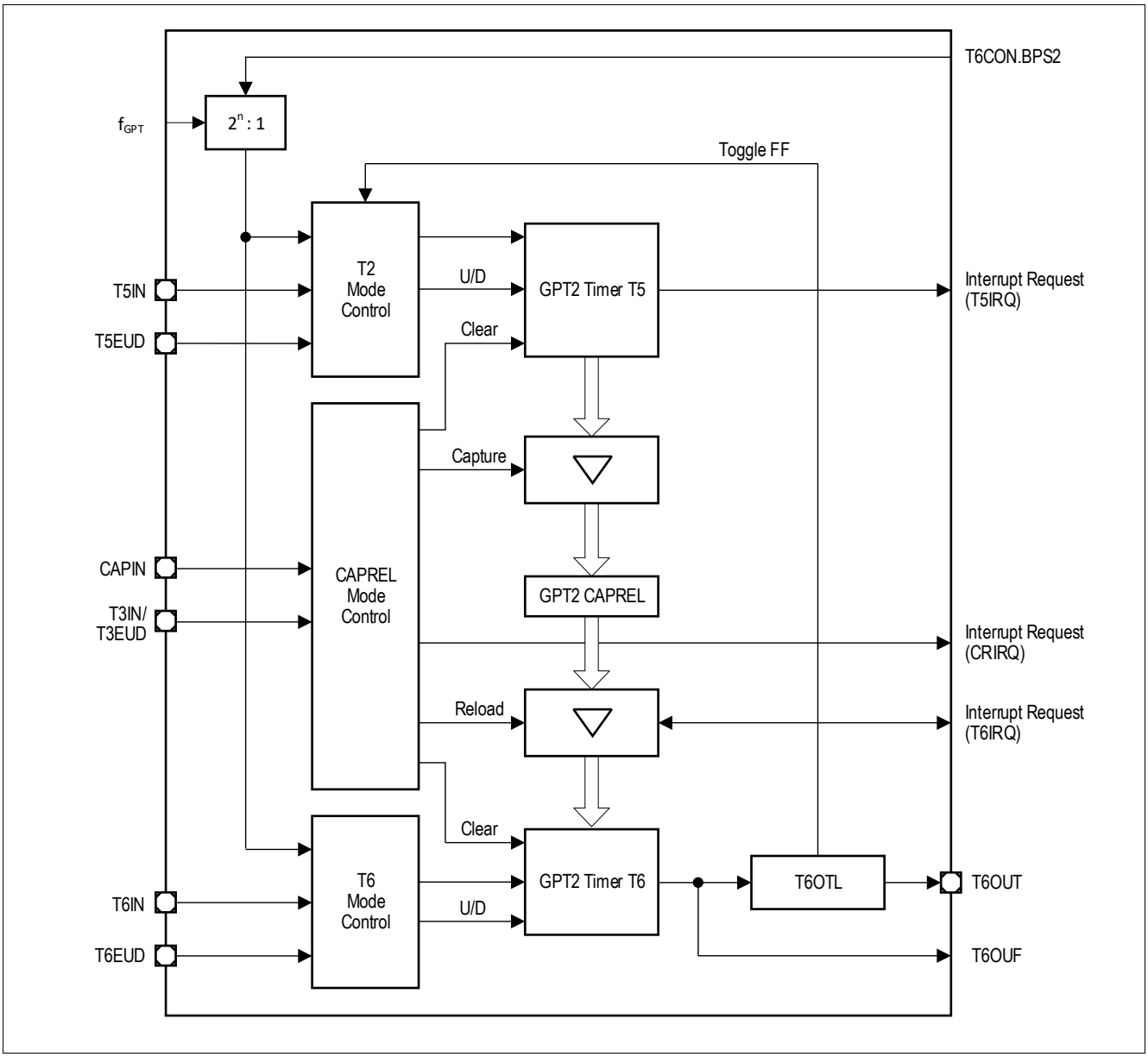


Figure 81 GPT2 block diagram

15.3 Timer Block GPT1

From a programmer's point of view, the GPT1 block is composed of a set of SFRs as summarized below. Those portions of port and direction registers which are used for alternate functions by the GPT1 block are shaded.

Data Registers	Control Registers	Interrupt Registers	Miscellaneous
T2	T2CON	GPT12IEN	GPT12PISEL
T3	T3CON	GPT12IRC	PISEL
T4	T4CON	GPT12ICLR	ID

Tx	GPT1 timer x data register
TxCON	GPT1 timer x control register
GPT12IEN	GPT12 interrupt enable register (located in SCU)
GPT12IRC	GPT12 interrupt status register (located in SCU)
GPT12ICLR	GPT12 interrupt status clear register (located in SCU)
GPT12PISEL	GPT12 extended pin select register (located in SCU)
PISEL	GPT12 local pin select register
ID	GPT module identification register

Figure 82 SFRs Associated with Timer Block GPT1

All three timers of block GPT1 (T2, T3, T4) can run in one of 4 basic modes: Timer Mode, Gated Timer Mode, Counter Mode, or Incremental Interface Mode. All timers can count up or down. Each timer of GPT1 is controlled by a separate control register TxCON.

Each timer has an input pin TxIN (alternate pin function) associated with it, which serves as the gate control in Gated Timer Mode, or as the count input in Counter Mode. The count direction (up/down) may be programmed via software or may be dynamically altered by a signal at the External Up/Down control input TxEUD (alternate pin function). An overflow/underflow of core timer T3 is indicated by the Output Toggle Latch T3OTL, whose state may be output on the associated pin T3OUT (alternate pin function). The auxiliary timers T2 and T4 may additionally be concatenated with the core timer T3 (through T3OTL) or may be used as capture or reload registers for the core timer T3.

The current contents of each timer can be read or modified by the CPU by accessing the corresponding timer count registers T2, T3, or T4, located in the non-bitaddressable SFR space (see **Section 15.3.7.1**). When any of the timer registers is written to by the CPU in the state immediately preceding a timer increment, decrement, reload, or capture operation, the CPU write operation has priority in order to guarantee correct results.

The interrupts of GPT1 are controlled through the SCU.**GPT12IEN** and SCU.**GPT12IRC**. These registers are not part of the GPT1 block.

The input and output lines of GPT1 are connected to pins. The control registers for the port functions are located in the respective port modules.

*Note: The timing requirements for external input signals can be found in **Section 15.3.5**, **Section 15.7.1** summarizes the module interface signals, including pins.*

15.3.1 GPT1 Core Timer T3 Control

The current contents of the core timer T3 are reflected by its count register T3. This register can also be written to by the CPU, for example, to set the initial start value.

The core timer T3 is configured and controlled via its control register T3CON.

Timer T3 Run Control

The core timer T3 can be started or stopped by software through bit T3R (Timer T3 Run Bit). This bit is relevant in all operating modes of T3. Setting bit T3R will start the timer, clearing bit T3R stops the timer.

In Gated Timer Mode, the timer will only run if T3R = 1 and the gate is active (high or low, as programmed).

Note: When bit T2RC or T4RC in timer control register T2CON or T4CON is set, bit T3R will also control (start and stop) the auxiliary timer(s) T2 and/or T4.

Count Direction Control

The count direction of the GPT1 timers (core timer and auxiliary timers) can be controlled either by software or by the external input pin TxEUD (Timer Tx External Up/Down Control Input). These options are selected by bits TxUD and TxUDE in the respective control register TxCON. When the up/down control is provided by software (bit TxUDE = 0), the count direction can be altered by setting or clearing bit TxUD. When bit TxUDE = 1, pin TxEUD is selected to be the controlling source of the count direction. However, bit TxUD can still be used to reverse the actual count direction, as shown in **Table 94**. The count direction can be changed regardless of whether or not the timer is running.

Note: When pin TxEUD is used as external count direction control input, it must be configured as input.

Timer T3 Output Toggle Latch

The overflow/underflow signal of timer T3 is connected to a block named 'Toggle Latch', shown in the Timer Mode diagrams. **Figure 83** illustrates the details of this block. An overflow or underflow of T3 will clock two latches: The first latch represents bit T3OTL in control register T3CON. The second latch is an internal latch toggled by T3OTL's output. Both latch outputs are connected to the input control blocks of the auxiliary timers T2 and T4. The output level of the shadow latch will match the output level of T3OTL, but is delayed by one clock cycle. When the T3OTL value changes, this will result in a temporarily different output level from T3OTL and the shadow latch, which can trigger the selected count event in T2 and/or T4.

When software writes to T3OTL, both latches are set or cleared simultaneously. In this case, both signals to the auxiliary timers carry the same level and no edge will be detected. Bit T3OE (overflow/underflow output enable) in register T3CON enables the state of T3OTL to be monitored via an external pin T3OUT. When T3OTL is linked to an external port pin (must be configured as output), T3OUT can be used to control external HW. If T3OE = 1, pin T3OUT outputs the state of T3OTL. If T3OE = 0, pin T3OUT outputs a high level (as long as the T3OUT alternate function is selected for the port pin).

The trigger signals can serve as an input for the counter function or as a trigger source for the reload function of the auxiliary timers T2 and T4.

As can be seen from **Figure 83**, when latch T3OTL is modified by software to determine the state of the output line, also the internal shadow latch is set or cleared accordingly. Therefore, no trigger condition is detected by T2/T4 in this case.

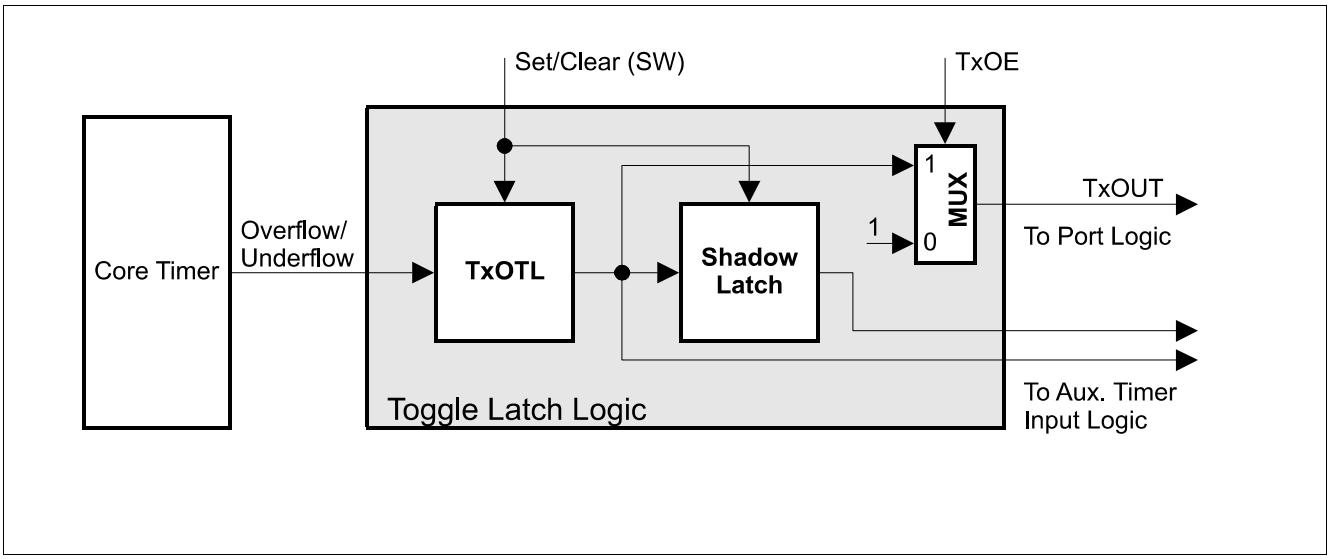


Figure 83 Block Diagram of the Toggle Latch Logic of Core Timer T3 (x = 3)

15.3.2 GPT1 Core Timer T3 Operating Modes

Timer T3 can operate in one of several modes.

Timer T3 in Timer Mode

Timer mode for the core timer T3 is selected by setting bitfield T3M in register T3CON to 000_B. In Timer Mode, T3 is clocked with the module's input clock f_{GPT} divided by two programmable prescalers controlled by bitfields BPS1 and T3I in register T3CON. Please see **Section 15.3.5** for details on the input clock options.

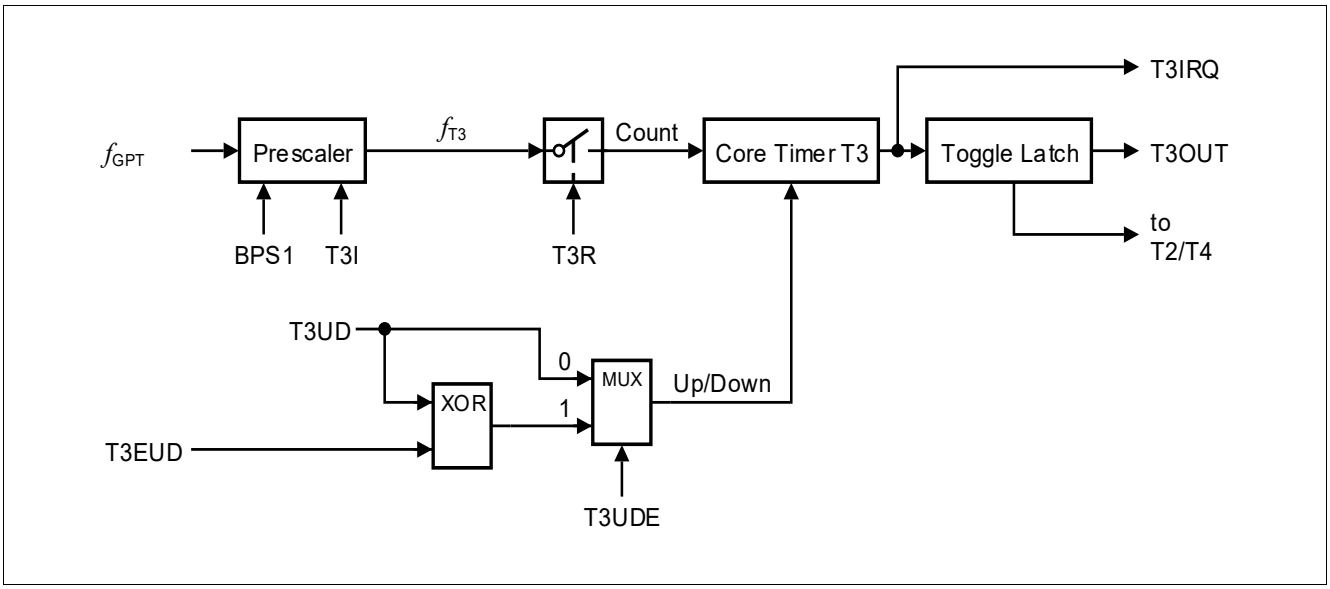


Figure 84 Block Diagram of Core Timer T3 in Timer Mode

Timer T3 in Gated Timer Mode

Gated Timer Mode for the core timer T3 is selected by setting bitfield T3M in register T3CON to 010_B or 011_B. Bit T3M.0 (T3CON.3) selects the active level of the gate input. The same options for the input frequency are available in Gated Timer Mode as in Timer Mode (see **Section 15.3.5**). However, the input clock to the timer in this mode is gated by the external input pin T3IN (Timer T3 External Input).
To enable this operation, the associated pin T3IN must be configured as input.

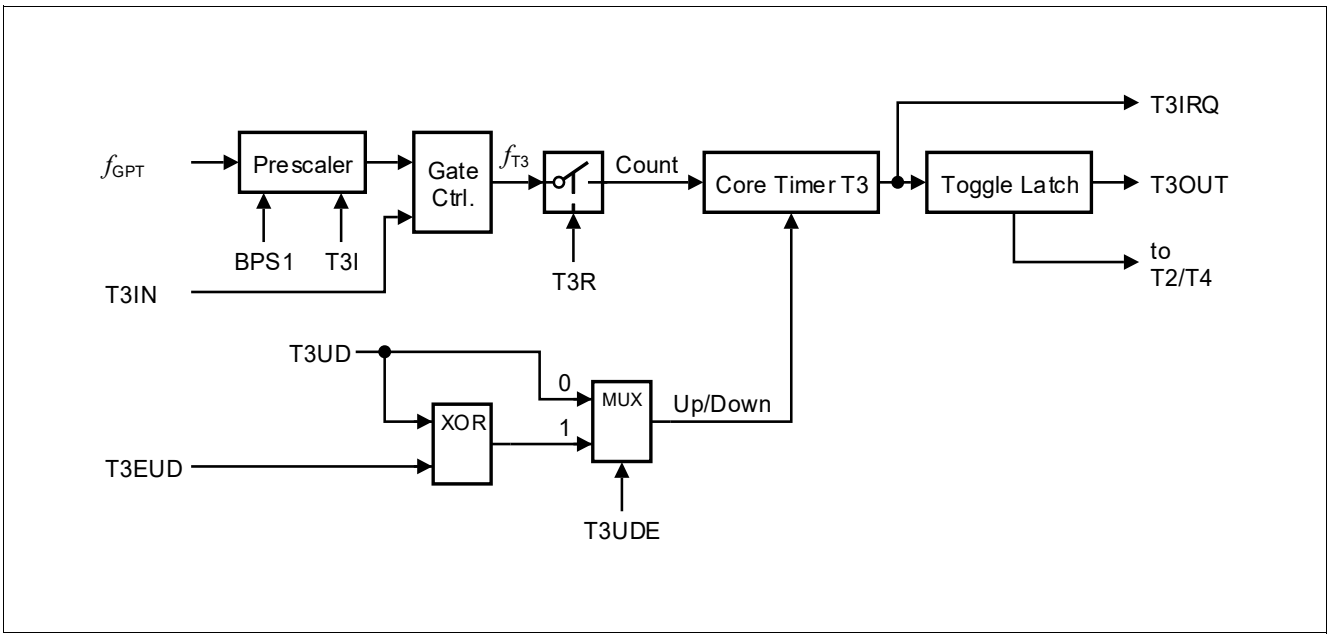


Figure 85 Block Diagram of Core Timer T3 in Gated Timer Mode

If T3M = 010_B, the timer is enabled when T3IN shows a low level. A high level at this line stops the timer. If T3M = 011_B, line T3IN must have a high level in order to enable the timer. Additionally, the timer can be turned on or off by software using bit T3R. The timer will only run if T3R is 1 and the gate is active. It will stop if either T3R is 0 or the gate is inactive.

Note: A transition of the gate signal at pin T3IN does not cause an interrupt request.

Timer T3 in Counter Mode

Counter Mode for the core timer T3 is selected by setting bitfield T3M in register T3CON to 001_B. In Counter Mode, timer T3 is clocked by a transition at the external input pin T3IN. The event causing an increment or decrement of the timer can be a positive, a negative, or both a positive and a negative transition at this line. Bitfield T3I in control register T3CON selects the triggering transition (see **Table 96**).

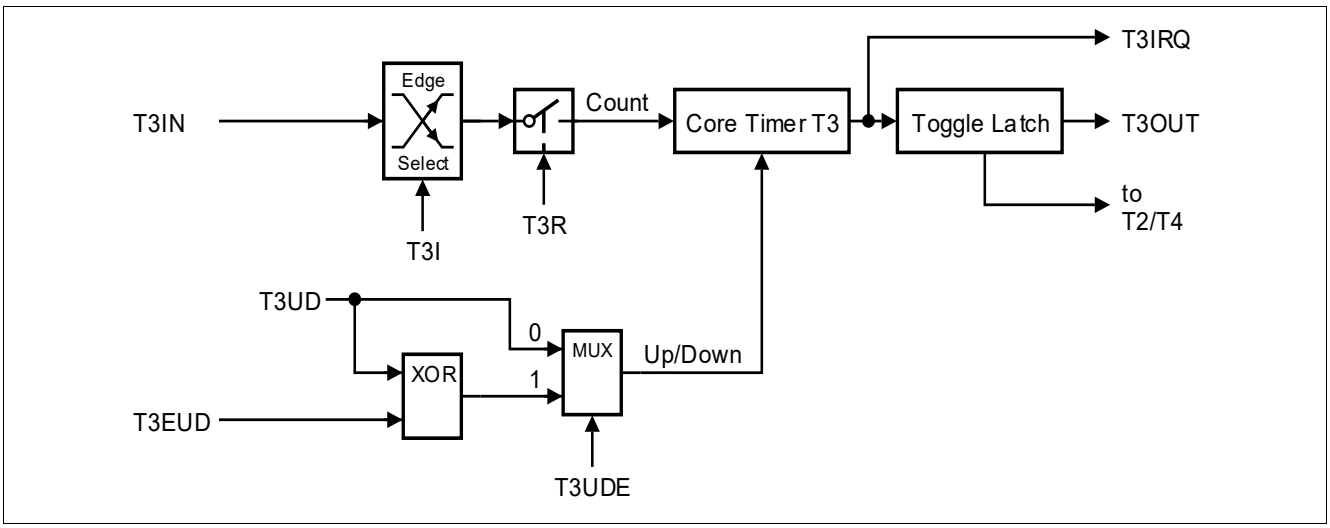


Figure 86 Block Diagram of Core Timer T3 in Counter Mode

For Counter Mode operation, pin T3IN must be configured as input. The maximum input frequency allowed in Counter Mode depends on the selected prescaler value. To ensure that a transition of the count input signal applied to T3IN is recognized correctly, its level must be held high or low for a minimum number of module clock cycles before it changes. This information can be found in **Section 15.3.5**.

Timer T3 in Incremental Interface Mode

Incremental interface mode for the core timer T3 is selected by setting bitfield T3M in register T3CON to 110_B or 111_B. In Incremental Interface Mode, the two inputs associated with core timer T3 (T3IN, T3EUD) are used to interface to an incremental encoder. T3 is clocked by each transition on one or both of the external input pins to provide 2-fold or 4-fold resolution of the encoder input.

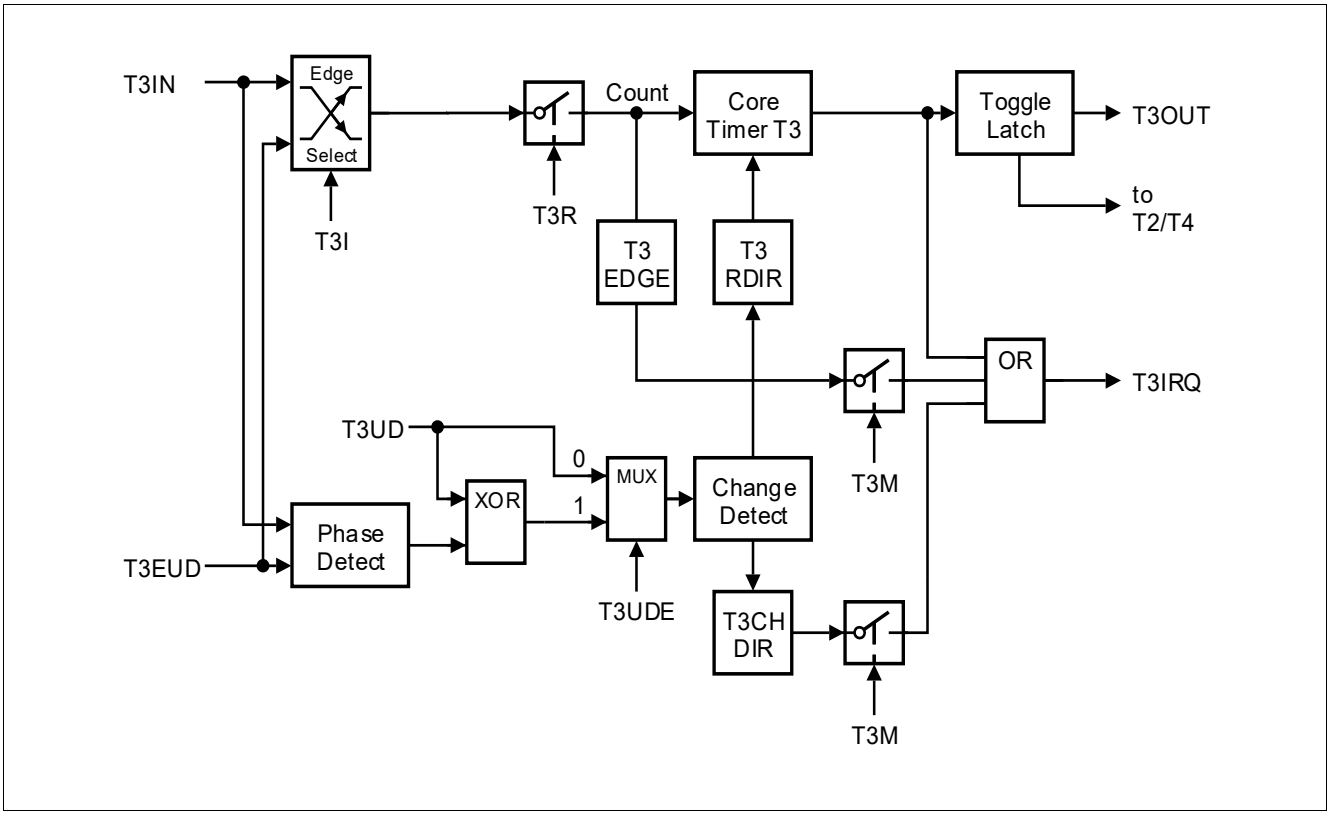


Figure 87 Block Diagram of Core Timer T3 in Incremental Interface Mode

Bitfield T3I in control register T3CON selects the triggering transitions (see **Table 99**). The sequence of the transitions of the two input signals is evaluated and generates count pulses as well as the direction signal. So T3 is modified automatically according to the speed and the direction of the incremental encoder and, therefore, its contents always represent the encoder's current position.

The interrupt request (T3IRQ) generation mode can be selected: In Rotation Detection Mode (T3M = 110_B), an interrupt request is generated each time the count direction of T3 changes. In Edge Detection Mode (T3M = 111_B), an interrupt request is generated each time a count edge for T3 is detected. Count direction, changes in the count direction, and count requests are monitored by status bits T3RDIR, T3CHDIR, and T3EDGE in register T3CON.

The incremental encoder can be connected directly to the TLE987x without external interface logic. In a standard system, however, comparators will be employed to convert the encoder's differential outputs (such as A, $\overline{A}$) to digital signals (such as A). This greatly increases noise immunity.

Note: The third encoder output T0, which indicates the mechanical zero position, may be connected to an external interrupt input and trigger a reset of timer T3.
If input T4IN is available, T0 can be connected there and clear T3 automatically without requiring an interrupt.

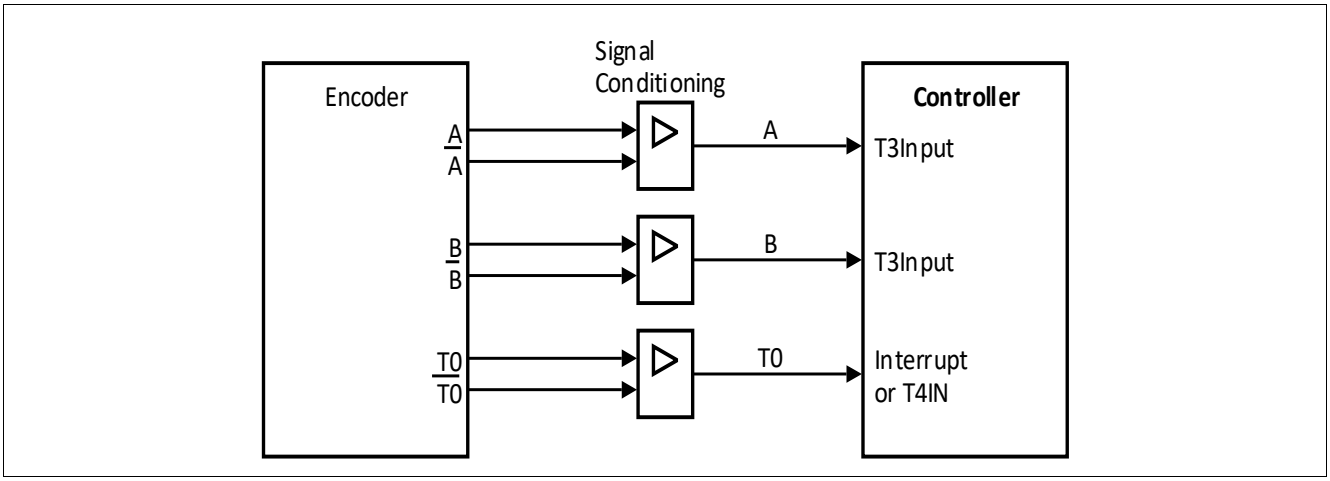


Figure 88 Connection of the Encoder to the TLE987x

For incremental interface operation, the following conditions must be met:

- Bitfield T3M must be 110_B or 111_B.
- Both pins T3IN and T3EUD must be configured as input.
- Pin T4IN must be configured as input, if used for T0.
- Bit T3UDE must be 1 to enable automatic external direction control.

The maximum count frequency allowed in Incremental Interface Mode depends on the selected prescaler value. To ensure that a transition of any input signal is recognized correctly, its level must be held high or low for a minimum number of module clock cycles before it changes. This information can be found in **Section 15.3.5**.

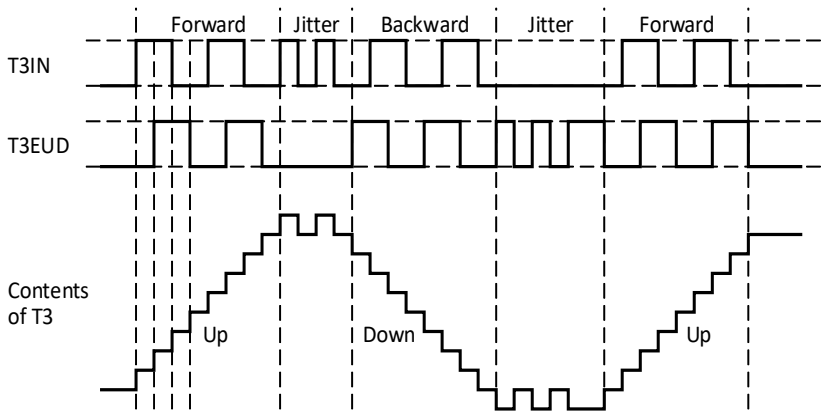
As in Incremental Interface Mode two input signals with a 90° phase shift are evaluated, their maximum input frequency can be half the maximum count frequency.

In Incremental Interface Mode, the count direction is automatically derived from the sequence in which the input signals change, which corresponds to the rotation direction of the connected sensor. **Table 90** summarizes the possible combinations.

Table 90 GPT1 Core Timer T3 (Incremental Interface Mode) Count Direction

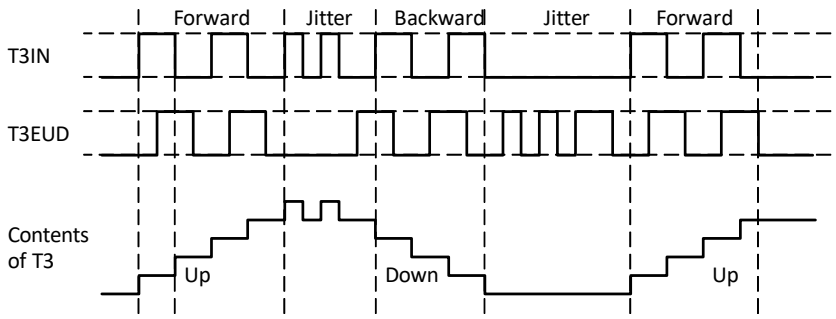
Level on Respective other Input	T3IN Input		T3EUD Input	
	Rising ↑	**Falling ↓**	**Rising ↑**	**Falling ↓**
High	Down	Up	Up	Down
Low	Up	Down	Down	Up

Figure 89 and **Figure 90** give examples of T3's operation, visualizing count signal generation and direction control. They also show how input jitter is compensated, which might occur if the sensor rests near to one of its switching points.



Note: This example shows the timer behaviour assuming that T3 counts upon any transition on input, i.e. T3I = '011_B'.

Figure 89 Evaluation of Incremental Encoder Signals, 2 Count Inputs



Note: This example shows the timer behaviour assuming that T3 counts upon any transition on input T3IN, i.e. T3I = '001_B'.

Figure 90 Evaluation of Incremental Encoder Signals, 1 Count Input

Note: Timer T3 operating in Incremental Interface Mode automatically provides information on the sensor's current position. Dynamic information (speed, acceleration, deceleration) may be obtained by measuring the incoming signal periods (see ***"Combined Capture Modes" on Page 471****).*

15.3.3 GPT1 Auxiliary Timers T2/T4 Control

Auxiliary timers T2 and T4 have exactly the same functionality. They can be configured for Timer Mode, Gated Timer Mode, Counter Mode, or Incremental Interface Mode with the same options for the timer frequencies and the count signal as the core timer T3. In addition to these 4 counting modes, the auxiliary timers can be concatenated with the core timer, or they may be used as reload or capture registers in conjunction with the core timer. The start/stop function of the auxiliary timers can be remotely controlled by the T3 run control bit. Several timers may thus be controlled synchronously.

The current contents of an auxiliary timer are reflected by its count register T2 or T4, respectively. These registers can also be written to by the CPU, for example, to set the initial start value.

The individual configurations for timers T2 and T4 are determined by their control registers T2CON and T4CON, which are organized identically. Note that functions which are present in all 3 timers of block GPT1 are controlled in the same bit positions and in the same manner in each of the specific control registers.

Note: The auxiliary timers have no output toggle latch and no alternate output function.

Timer T2/T4 Run Control

Each of the auxiliary timers T2 and T4 can be started or stopped by software in two different ways:

- Through the associated timer run bit (T2R or T4R). In this case it is required that the respective control bit TxRC = 0.
- Through the core timer's run bit (T3R). In this case the respective remote control bit must be set (TxRC = 1).

The selected run bit is relevant in all operating modes of T2/T4. Setting the bit will start the timer, clearing the bit stops the timer.

In Gated Timer Mode, the timer will only run if the selected run bit is set and the gate is active (high or low, as programmed).

Note: If remote control is selected T3R will start/stop timer T3 and the selected auxiliary timer(s) synchronously.

Count Direction Control

The count direction of the GPT1 timers (core timer and auxiliary timers) is controlled in the same way, either by software or by the external input pin TxEUD. Please refer to the description in **Table 94**.

Note: When pin TxEUD is used as external count direction control input, it must be configured as input.

15.3.4 GPT1 Auxiliary Timers T2/T4 Operating Modes

The operation of the auxiliary timers in the basic operating modes is almost identical with the core timer's operation, with very few exceptions. Additionally, some combined operating modes can be selected.

Timers T2 and T4 in Timer Mode

Timer mode for an auxiliary timer Tx is selected by setting its bitfield TxM in register TxCON to 000_B.

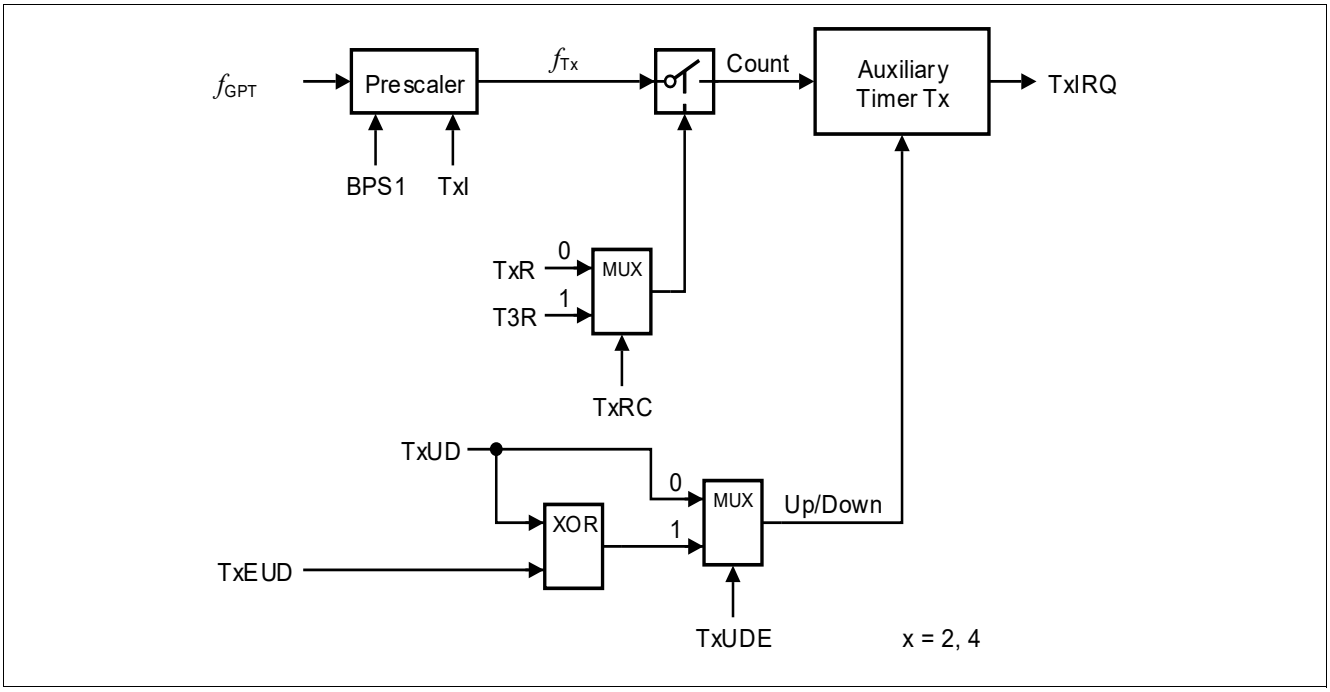


Figure 91 Block Diagram of an Auxiliary Timer in Timer Mode

Timers T2 and T4 in Gated Timer Mode

Gated Timer Mode for an auxiliary timer Tx is selected by setting bitfield TxM in register TxCON to 010_B or 011_B. Bit TxM.0 (TxCON.3) selects the active level of the gate input.

Note: A transition of the gate signal at line TxIN does not cause an interrupt request.

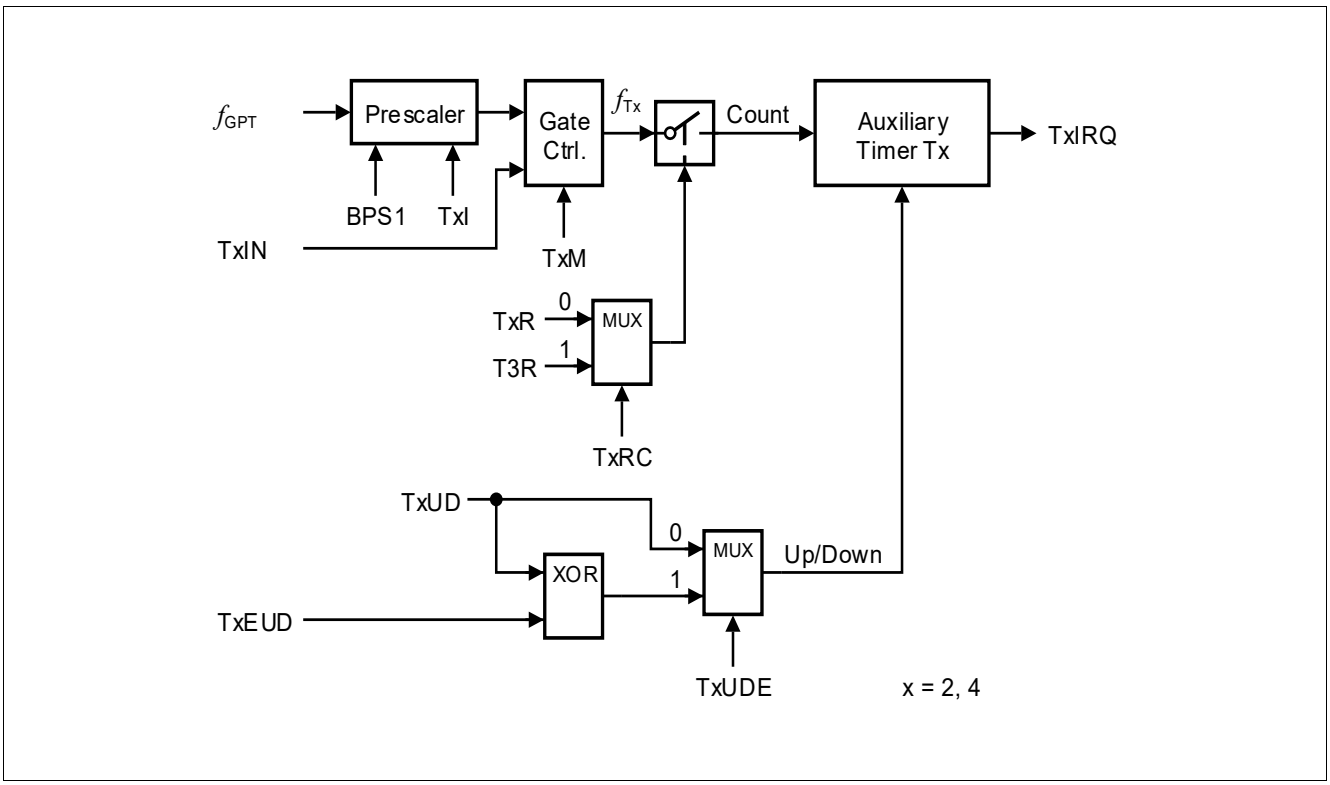


Figure 92 Block Diagram of an Auxiliary Timer in Gated Timer Mode

Note: There is no output toggle latch for T2 and T4.
Start/stop of an auxiliary timer can be controlled locally or remotely.

Timers T2 and T4 in Counter Mode

Counter Mode for an auxiliary timer Tx is selected by setting bitfield TxM in register TxCON to 001_B. In Counter Mode, an auxiliary timer can be clocked either by a transition at its external input line TxIN, or by a transition of timer T3's toggle latch T3OTL. The event causing an increment or decrement of a timer can be a positive, a negative, or both a positive and a negative transition at either the respective input pin or at the toggle latch. Bitfield TxI in control register TxCON selects the triggering transition (see **Table 98**).

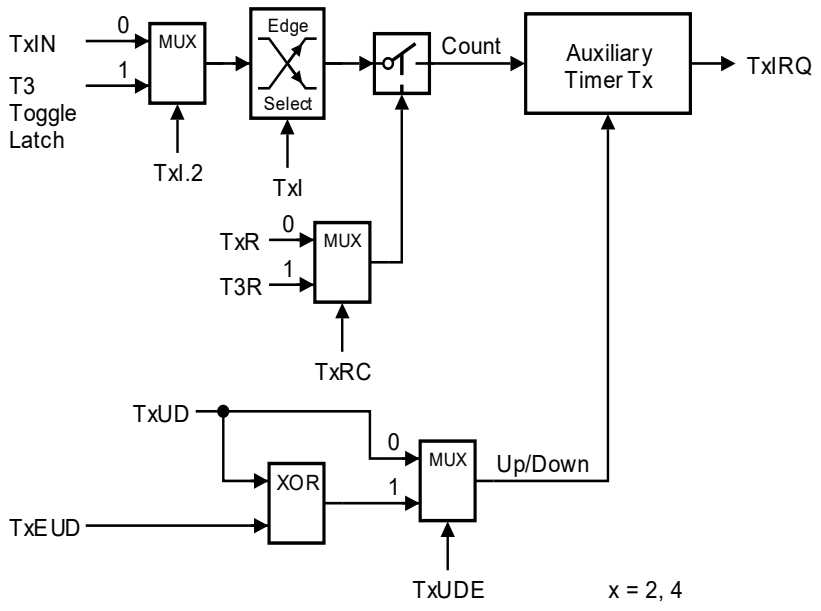


Figure 93 Block Diagram of an Auxiliary Timer in Counter Mode

Note: Only state transitions of T3OTL which are caused by the overflows/underflows of T3 will trigger the counter function of T2/T4. Modifications of T3OTL via software will NOT trigger the counter function of T2/T4.

For counter operation, pin TxIN must be configured as input. The maximum input frequency allowed in Counter Mode depends on the selected prescaler value. To ensure that a transition of the count input signal applied to TxIN is recognized correctly, its level must be held high or low for a minimum number of module clock cycles before it changes. This information can be found in **Section 15.3.5**.

Timer Concatenation

Using the toggle bit T3OTL as a clock source for an auxiliary timer in Counter Mode concatenates the core timer T3 with the respective auxiliary timer. This concatenation forms either a 32-bit or a 33-bit timer/counter, depending on which transition of T3OTL is selected to clock the auxiliary timer.

- **32-bit Timer/Counter:** If both a positive and a negative transition of T3OTL are used to clock the auxiliary timer, this timer is clocked on every overflow/underflow of the core timer T3. Thus, the two timers form a 32-bit timer.
- **33-bit Timer/Counter:** If either a positive or a negative transition of T3OTL is selected to clock the auxiliary timer, this timer is clocked on every second overflow/underflow of the core timer T3. This configuration forms a 33-bit timer (16-bit core timer + T3OTL + 16-bit auxiliary timer).

As long as bit T3OTL is not modified by software, it represents the state of the internal toggle latch, and can be regarded as part of the 33-bit timer.

The count directions of the two concatenated timers are not required to be the same. This offers a wide variety of different configurations.

T3, which represents the low-order part of the concatenated timer, can operate in Timer Mode, Gated Timer Mode or Counter Mode in this case.

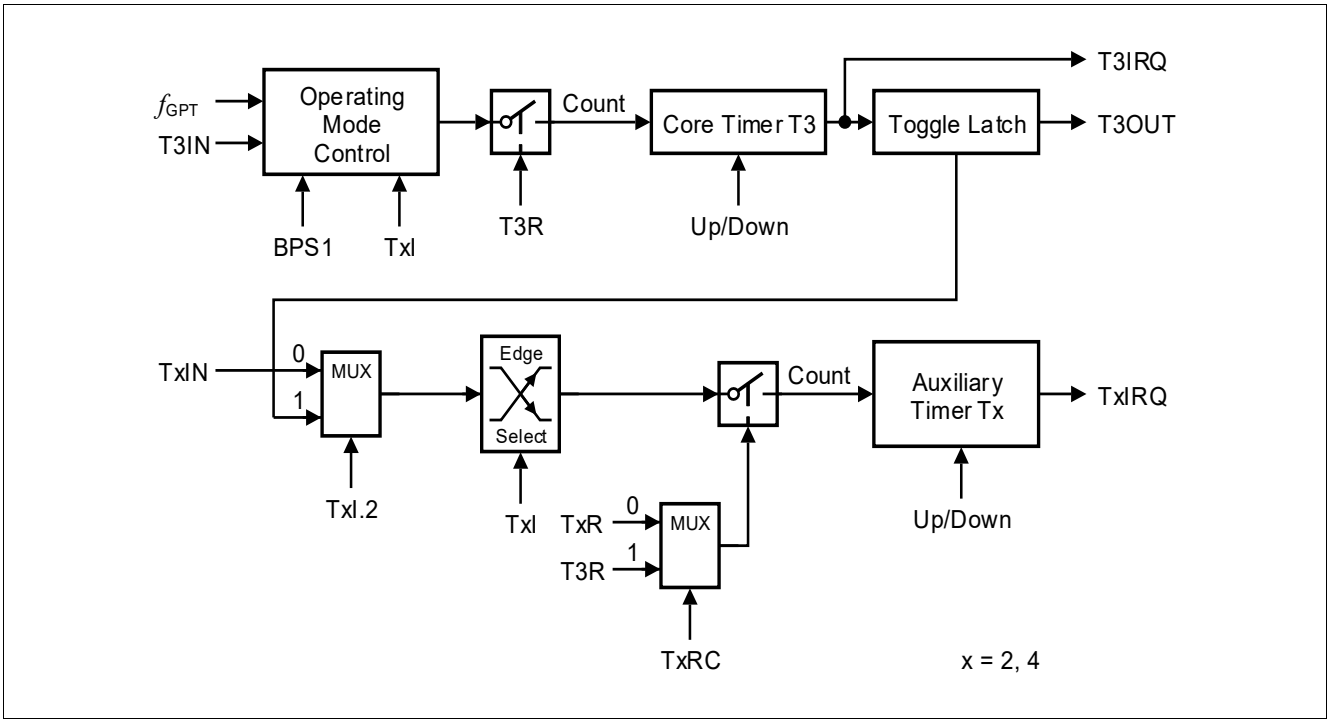


Figure 94 Concatenation of Core Timer T3 and an Auxiliary Timer

Timers T2 and T4 in Capture Mode

Capture mode for an auxiliary timer Tx is selected by setting bitfield TxM in the respective register TxCON to 101_B. In capture mode, the contents of the core timer T3 are latched into an auxiliary timer register in response to a signal transition at the respective auxiliary timer's external input pin TxIN. The capture trigger signal can be a positive, a negative, or both a positive and a negative transition.

The two least significant bits of bitfield TxI select the active transition (see **Table 98**). Bit 2 of TxI is irrelevant for capture mode and must be cleared (TxI.2 = 0).

Note: When programmed for capture mode, the respective auxiliary timer (T2 or T4) stops independently of its run flag T2R or T4R.

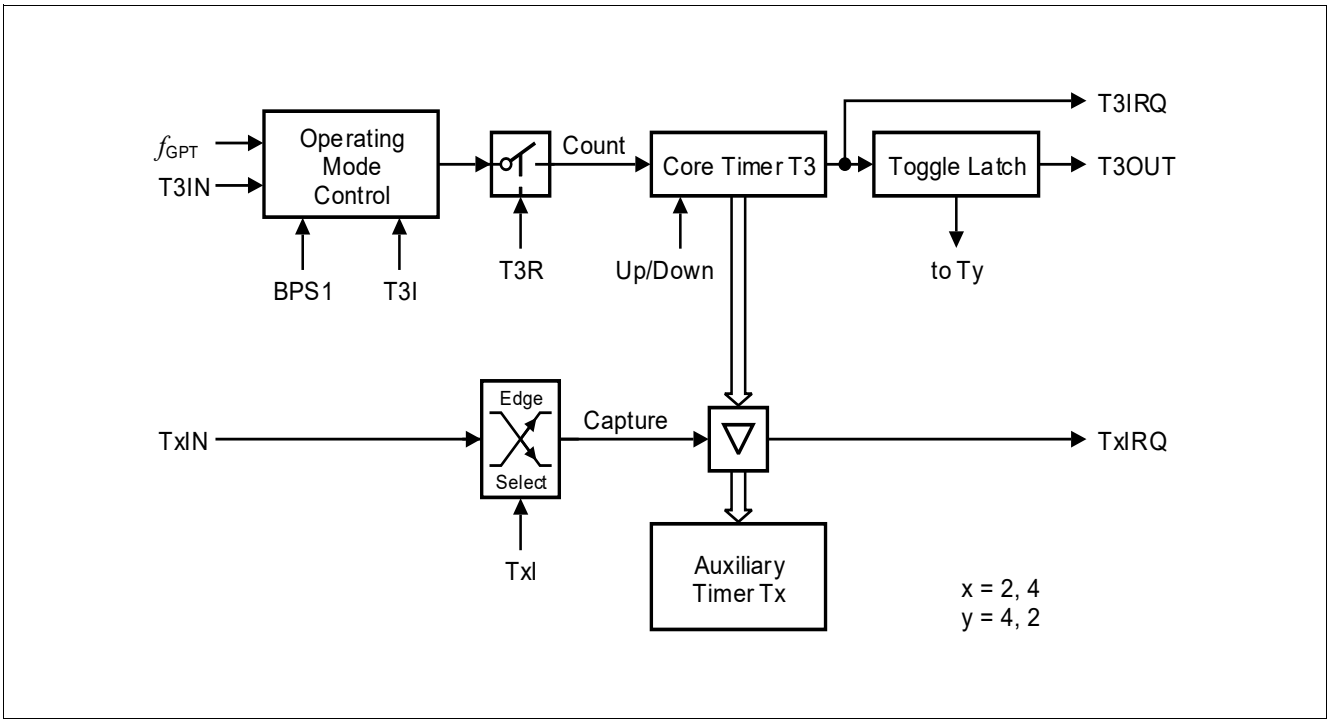


Figure 95 GPT1 Auxiliary Timer in Capture Mode

Upon a trigger (selected transition) at the corresponding input pin TxIN the contents of the core timer are loaded into the auxiliary timer register and the associated interrupt request flag TxIR will be set.

For capture mode operation, the respective timer input pin TxIN must be configured as input. To ensure that a transition of the capture input signal applied to TxIN is recognized correctly, its level must be held high or low for a minimum number of module clock cycles, detailed in **Section 15.3.5**.

Timers T2 and T4 in Incremental Interface Mode

Incremental interface mode for an auxiliary timer Tx is selected by setting bitfield TxM in the respective register TxCON to 110_B or 111_B. In Incremental Interface Mode, the two inputs associated with an auxiliary timer Tx (TxIN, TxEUD) are used to interface to an incremental encoder. Tx is clocked by each transition on one or both of the external input pins to provide 2-fold or 4-fold resolution of the encoder input.

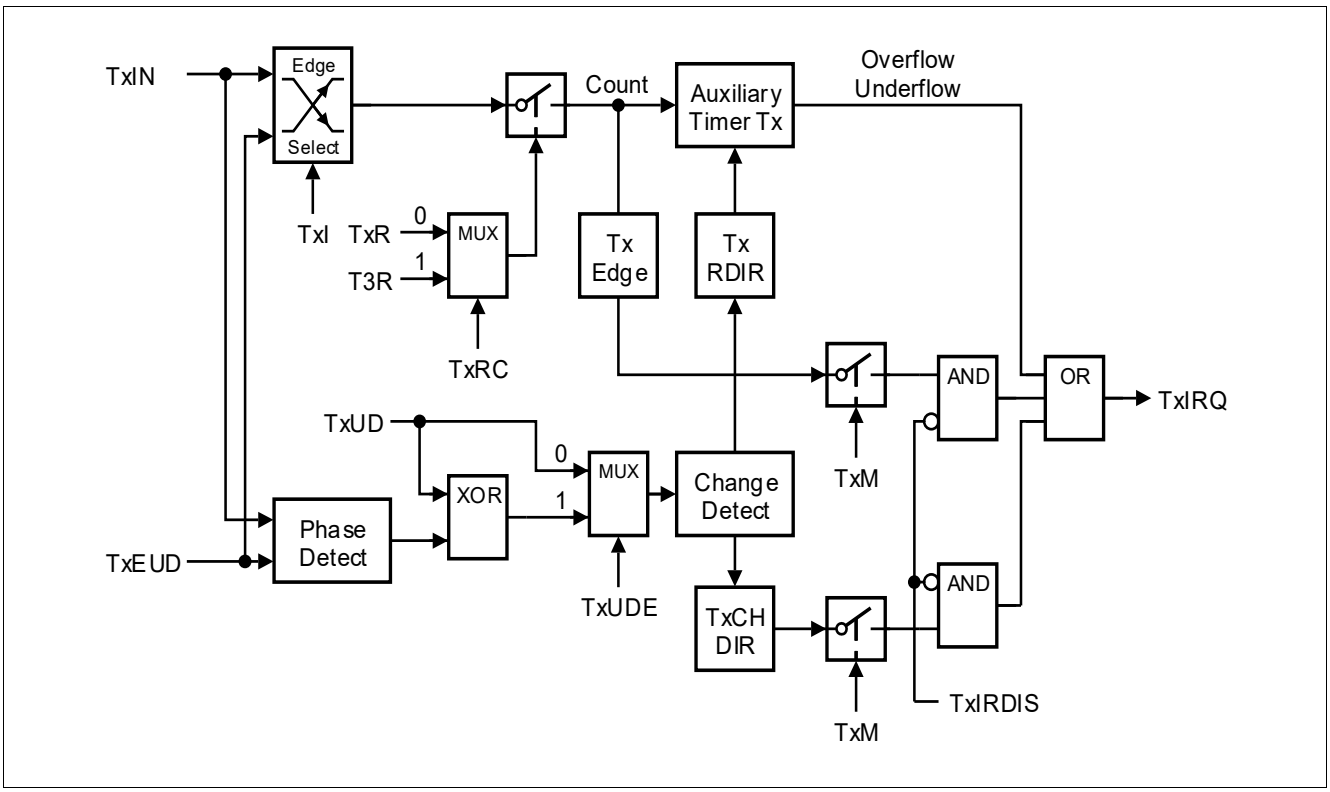


Figure 96 Block Diagram of an Auxiliary Timer in Incremental Interface Mode

The operation of the auxiliary timers T2 and T4 in Incremental Interface Mode and the interrupt generation are the same as described for the core timer T3. The descriptions, figures and tables apply accordingly.

Note: Timers T2 and T4 operating in Incremental Interface Mode automatically provide information on the sensor's current position. For dynamic information (speed, acceleration, deceleration) see ***"Combined Capture Modes" on Page 471****).*

Timers T2 and T4 in Reload Mode

Reload Mode for an auxiliary timer Tx is selected by setting bitfield TxM in the respective register TxCON to 100_B. In reload mode, the core timer T3 is reloaded with the contents of an auxiliary timer register, triggered by one of two different signals. The trigger signal is selected the same way as the clock source for Counter Mode (see **Table 98**), i.e. a transition of the auxiliary timer's input TxIN or the toggle latch T3OTL may trigger the reload.

Note: When programmed for reload mode, the respective auxiliary timer (T2 or T4) stops independently of its run flag T2R or T4R.
The timer input pin TxIN must be configured as input if it shall trigger a reload operation.

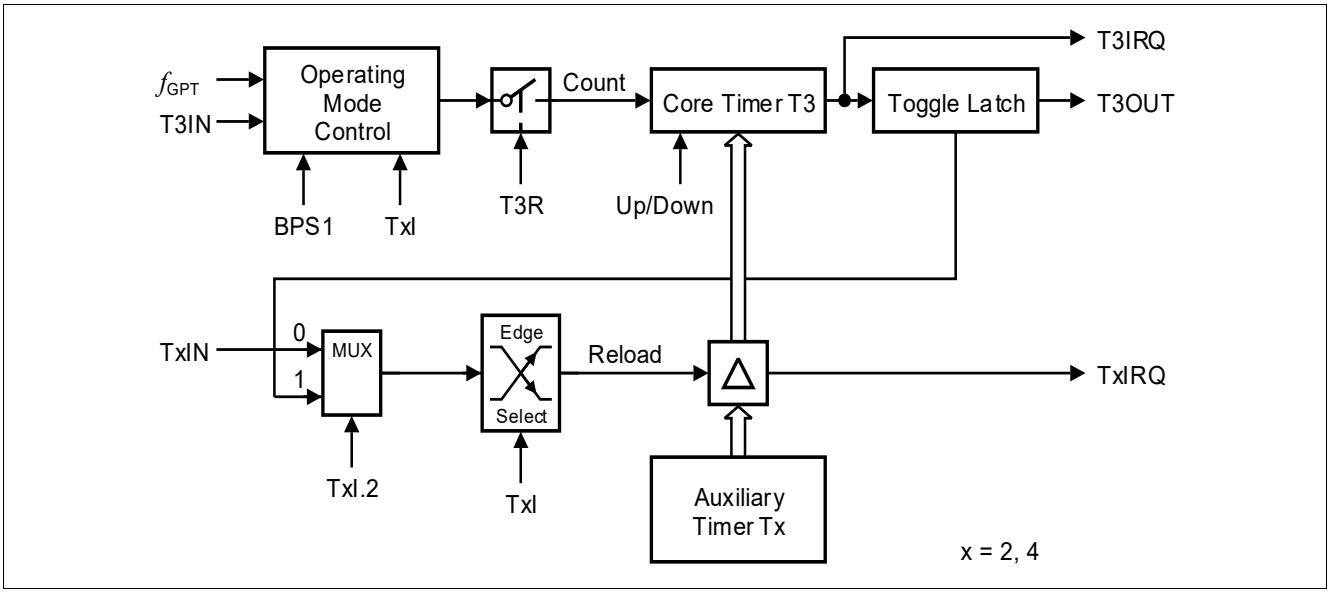


Figure 97 GPT1 Auxiliary Timer in Reload Mode

Upon a trigger signal, T3 is loaded with the contents of the respective timer register (T2 or T4) and the respective interrupt request flag (T2IR or T4IR) is set.

Note: When a T3OTL transition is selected for the trigger signal, the interrupt request flag T3IR will also be set upon a trigger, indicating T3's overflow or underflow. Modifications of T3OTL via software will NOT trigger the counter function of T2/T4.

To ensure that a transition of the reload input signal applied to TxIN is recognized correctly, its level must be held high or low for a minimum number of module clock cycles, detailed in **Section 15.3.5**.

The reload mode triggered by the T3 toggle latch can be used in a number of different configurations. The following functions can be performed, depending on the selected active transition:

- If both a positive and a negative transition of T3OTL are selected to trigger a reload, the core timer will be reloaded with the contents of the auxiliary timer each time it overflows or underflows. This is the standard reload mode (reload on overflow/underflow).
- If either a positive or a negative transition of T3OTL is selected to trigger a reload, the core timer will be reloaded with the contents of the auxiliary timer on every second overflow or underflow.
- Using this "single-transition" mode for both auxiliary timers allows to perform very flexible Pulse Width Modulation (PWM). One of the auxiliary timers is programmed to reload the core timer on a positive transition of T3OTL, the other is programmed for a reload on a negative transition of T3OTL. With this combination the core timer is alternately reloaded from the two auxiliary timers.

Figure 98 shows an example for the generation of a PWM signal using the “single-transition” reload mechanism. T2 defines the high time of the PWM signal (reloaded on positive transitions) and T4 defines the low time of the PWM signal (reloaded on negative transitions). The PWM signal can be output on pin T3OUT if T3OE = 1. With this method, the high and low time of the PWM signal can be varied in a wide range.

Note: The output toggle latch T3OTL is accessible via software and may be changed, if required, to modify the PWM signal.
However, this will NOT trigger the reloading of T3.

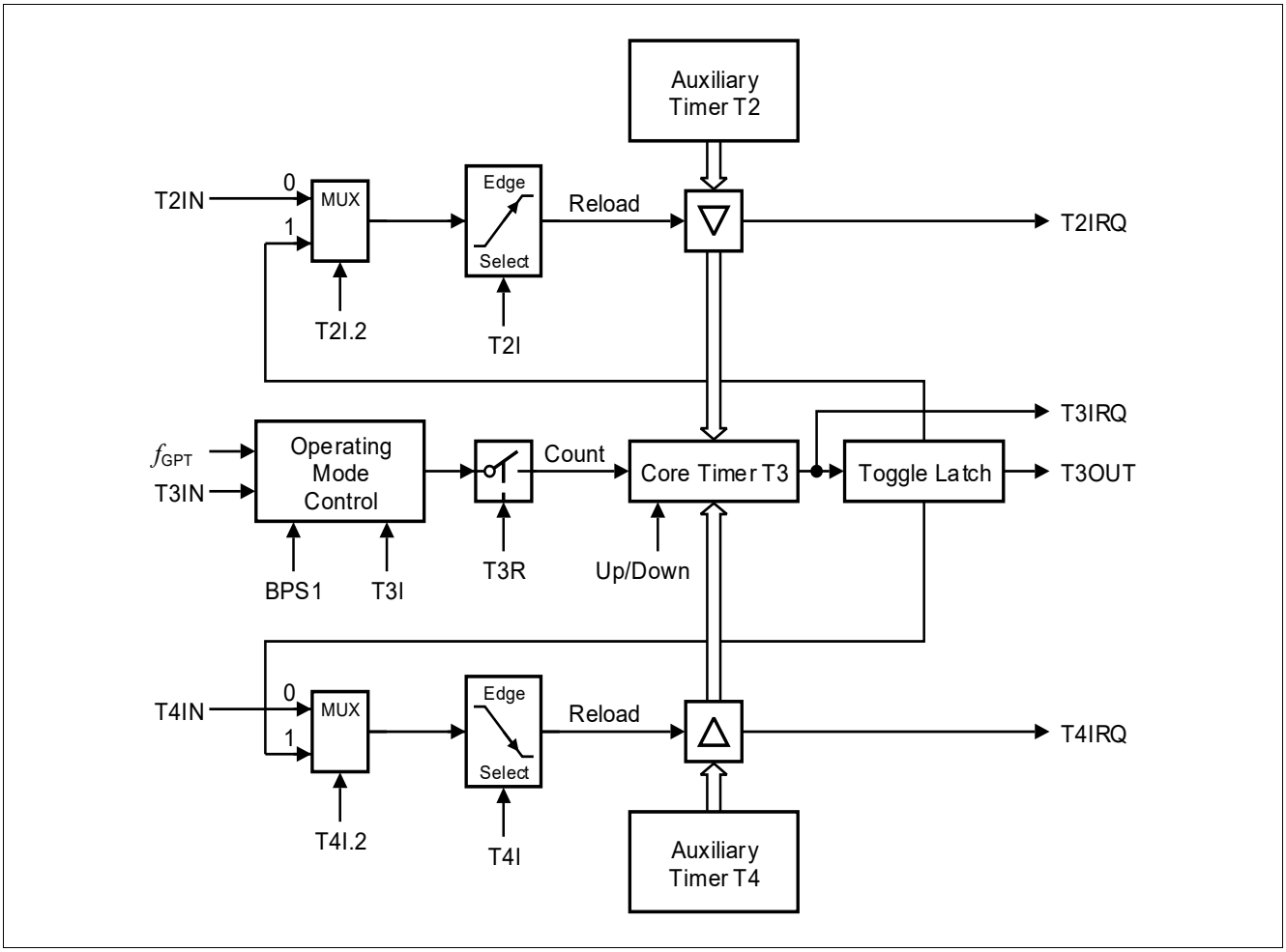


Figure 98 GPT1 Timer Reload Configuration for PWM Generation

Note: Although possible, selecting the same reload trigger event for both auxiliary timers should be avoided. In such a case, both reload registers would try to load the core timer at the same time. If this combination is selected, T2 is disregarded and the contents of T4 is reloaded.

15.3.5 GPT1 Clock Signal Control

All actions within the timer block GPT1 are triggered by transitions of its basic clock. This basic clock is derived from the system clock by a basic block prescaler, controlled by bitfield BPS1 in register T3CON (see **Figure 80**). The count clock can be generated in two different ways:

- **Internal count clock**, derived from GPT1's basic clock via a programmable prescaler, is used for (gated) Timer Mode.
- **External count clock**, derived from the timer's input pin(s), is used for Counter Mode.

For both ways, the basic clock determines the maximum count frequency and the timer's resolution:

Table 91 Basic Clock Selection for Block GPT1

Block Prescaler[1)]	**BPS1 = 01_B**	**BPS1 = 00_B**[2)]	**BPS1 = 11_B**	**BPS1 = 10_B**
Prescaling Factor for GPT1: F(BPS1)	F(BPS1) = 4	F(BPS1) = 8	F(BPS1) = 16	F(BPS1) = 32
Maximum External Count Frequency	$f_{GPT}/8$	$f_{GPT}/16$	$f_{GPT}/32$	$f_{GPT}/64$
Input Signal Stable Time	$4 \times t_{GPT}$	$8 \times t_{GPT}$	$16 \times t_{GPT}$	$32 \times t_{GPT}$

1) Please note the non-linear encoding of bitfield BPS1.
2) Default after reset.

Note: When initializing the GPT1 block, and the block prescaler BPS1 in register T3CON needs to be set to a value different from its reset value (00_B), it must be initialized first before any mode involving external trigger signals is configured. These modes include counter, incremental interface, capture, and reload mode. Otherwise, unintended count/capture/reload events may occur.
In this case (e.g. when changing BPS1 during operation of the GPT1 block), disable related interrupts before modification of BPS1, and afterwards clear the corresponding service request flags and re-initialize those registers (T2, T3, T4) that might be affected by a count/capture/reload event.

Internal Count Clock Generation

In Timer Mode and Gated Timer Mode, the count clock for each GPT1 timer is derived from the GPT1 basic clock by a programmable prescaler, controlled by bitfield TxI in the respective timer's control register TxCON.

The count frequency f_{Tx} for a timer Tx and its resolution r_{Tx} are scaled linearly with lower clock frequencies, as can be seen from the following formula:

$$f_{Tx} = \frac{f_{GPT}}{F(BPS1) \times 2^{<TxI>}} \qquad r_{Tx}[\mu s] = \frac{F(BPS1) \times 2^{<TxI>}}{f_{GPT}[MHz]} \tag{15.1}$$

The effective count frequency depends on the common module clock prescaler factor F(BPS1) as well as on the individual input prescaler factor $2^{<TxI>}$. **Table 95** summarizes the resulting overall divider factors for a GPT1 timer that result from these cascaded prescalers.

Table 92 lists GPT1 timer's parameters (such as count frequency, resolution, and period) resulting from the selected overall prescaler factor and the module clock f_{GPT}. Note that some numbers may be rounded.

Table 92 GPT1 Timer Parameters

Module Clock f_{GPT} = 10 MHz			Overall Prescaler Factor	Module Clock f_{GPT} = 40 MHz		
Frequency	**Resolution**	**Period**		**Frequency**	**Resolution**	**Period**
2.5 MHz	400 ns	26.21 ms	4	10.0 MHz	100 ns	6.55 ms
1.25 MHz	800 ns	52.43 ms	8	5.0 MHz	200 ns	13.11 ms
625.0 kHz	1.6 µs	104.9 ms	16	2.5 MHz	400 ns	26.21 ms
312.5 kHz	3.2 µs	209.7 ms	32	1.25 MHz	800 ns	52.43 ms
156.25 kHz	6.4 µs	419.4 ms	64	625.0 kHz	1.6 µs	104.9 ms
78.125 kHz	12.8 µs	838.9 ms	128	312.5 kHz	3.2 µs	209.7 ms
39.06 kHz	25.6 µs	1.678 s	256	156.25 kHz	6.4 µs	419.4 ms
19.53 kHz	51.2 µs	3.355 s	512	78.125 kHz	12.8 µs	838.9 ms
9.77 kHz	102.4 µs	6.711 s	1024	39.06 kHz	25.6 µs	1.678 s
4.88 kHz	204.8 µs	13.42 s	2048	19.53 kHz	51.2 µs	3.355 s
2.44 kHz	409.6 µs	26.84 s	4096	9.77 kHz	102.4 µs	6.711 s

External Count Clock Input

The external input signals of the GPT1 block are sampled with the GPT1 basic clock (see **Figure 80**). To ensure that a signal is recognized correctly, its current level (high or low) must be held active for at least one complete sampling period, before changing. A signal transition is recognized if two subsequent samples of the input signal represent different levels. Therefore, a minimum of two basic clock periods are required for the sampling of an external input signal. Thus, the maximum frequency of an input signal must not be higher than half the basic clock.

Table 93 summarizes the resulting requirements for external GPT1 input signals.

Table 93 GPT1 External Input Signal Limits

GPT1 Basic Clock = 10 MHz		Input Frequ. Factor	GPT1 Divider BPS1	Input Phase Duration	GPT1 Basic Clock = 40 MHz	
Max. Input Frequency	**Min. Level Hold Time**				**Max. Input Frequency**	**Min. Level Hold Time**
1.25 MHz	400 ns	$f_{GPT}/8$	01_B	$4 \times t_{GPT}$	5.0 MHz	100 ns
625.0 kHz	800 ns	$f_{GPT}/16$	00_B	$8 \times t_{GPT}$	2.5 MHz	200 ns
312.5 kHz	1.6 µs	$f_{GPT}/32$	11_B	$16 \times t_{GPT}$	1.25 MHz	400 ns
156.25 kHz	3.2 µs	$f_{GPT}/64$	10_B	$32 \times t_{GPT}$	625.0 kHz	800 ns

These limitations are valid for all external input signals to GPT1, including the external count signals in Counter Mode and Incremental Interface Mode, the gate input signals in Gated Timer Mode, and the external direction signals.

15.3.6 Interrupt Control for GPT1 Timers

When a timer overflows from $FFFF_H$ to 0000_H (when counting up), or when it underflows from 0000_H to $FFFF_H$ (when counting down), its interrupt request flag in register GPT12E_T2, GPT12E_T3, or GPT12E_T4 will be set. This will cause an interrupt to the respective timer interrupt vector, if the respective interrupt enable bit is set.

In **Reload Mode**, upon a trigger signal, T3 is loaded with the contents of the respective timer (T2 or T4) and the respective interrupt request flag in register GPT12E_T2 or GPT12E_T4 is set.

In **Incremental Interface Mode**, the interrupt request generation can be selected as follows:

- In Rotation Detection Mode (T3M = 110_B), an interrupt request is generated each time the count direction of T3 changes.
- In Edge Detection Mode (T3M = 111_B), an interrupt request is generated each time a count edge for T3 is detected.

In **Capture Mode**, upon a trigger (selected transition) at the corresponding input pin the content of the core timer T3 are loaded into the auxiliary timer register Tx and the associated interrupt request flag in register GPTE12_T2 or GPT12E_T4 will be set.

15.3.7 GPT1 Registers

15.3.7.1 GPT1 Timer Registers

T2

Timer T2 Count Register **(20_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T2															
rwh															

Field	Bits	Typ	Description
T2	[15:0]	rwh	**Timer T2 Current Value** Contains the current value of the timer T2

T3

Timer T3 Count Register **(24_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T3															
rwh															

Field	Bits	Typ	Description
T3	[15:0]	rwh	**Timer T3 Current Value** Contains the current value of the timer T3

T4

Timer T4 Count Register **(28_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T4															
rwh															

Field	Bits	Typ	Description
T4	[15:0]	rwh	**Timer T4 Current Value** Contains the current value of the timer T4

15.3.7.2 GPT1 Timer Control Registers

15.3.7.2.1 GPT1 Core Timer T3 Control Register

T3CON
Timer T3 Control Register **($0C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T3RDIR	**T3CHDIR**	**T3EDGE**	**BPS1**		**T3OTL**	**T3OE**	**T3UDE**	**T3UD**	**T3R**	**T3M**			**T3I**		
rh	rwh	rwh	rw		rwh	rw	rw	rw	rw	rw			rw		

Field	Bits	Type	Description
T3I	[2:0]	rw	**Timer T3 Input Parameter Selection** Depends on the operating mode, see respective sections for encoding: **Table 95** for Timer Mode and Gated Timer Mode **Table 96** for Counter Mode **Table 99** for Incremental Interface Mode
T3M	[5:3]	rw	**Timer T3 Mode Control** 000_B Timer Mode 001_B Counter Mode 010_B Gated Timer Mode with gate active low 011_B Gated Timer Mode with gate active high 100_B Reserved. Do not use this combination 101_B Reserved. Do not use this combination 110_B Incremental Interface Mode (Rotation Detection Mode) 111_B Incremental Interface Mode (Edge Detection Mode)
T3R	6	rw	**Timer T3 Run Bit** 0_B Timer T3 stops 1_B Timer T3 runs
T3UD	7	rw	**Timer T3 Up/Down Control**[1)] 0_B Timer T3 counts up 1_B Timer T3 counts down *Note: This bit only controls count direction of T3 if bit T3UDE = 0.*
T3UDE	8	rw	**Timer T3 External Up/Down Enable**[1)] 0_B Count direction is controlled by bit T3UD; input T3EUD is disconnected 1_B Count direction is controlled by input T3EUD
T3OE	9	rw	**Overflow/Underflow Output Enable** 0_B Alternate Output Function Disabled 1_B State of T3 toggle latch is output on pin T3OUT

Field	Bits	Type	Description
T3OTL	10	rwh	**Timer T3 Overflow Toggle Latch** Toggles on each overflow/underflow of T3. Can be set or cleared by software (see separate description)
BPS1	[12:11]	rw	**GPT1 Block Prescaler Control** Selects the basic clock for block GPT1 (see also **Section 15.3.5**) 00_B $f_{GPT}/8$ 01_B $f_{GPT}/4$ 10_B $f_{GPT}/32$ 11_B $f_{GPT}/16$
T3EDGE	13	rwh	**Timer T3 Edge Detection Flag** The bit is set each time a count edge is detected. T3EDGE must be cleared by software. 0_B No count edge was detected 1_B A count edge was detected
T3CHDIR	14	rwh	**Timer T3 Count Direction Change Flag** This bit is set each time the count direction of timer T3 changes. T3CHDIR must be cleared by software. 0_B No change of count direction was detected 1_B A change of count direction was detected
T3RDIR	15	rh	**Timer T3 Rotation Direction Flag** 0_B Timer T3 counts up 1_B Timer T3 counts down

1) See **Table 104** for encoding of bits T3UD and T3UDE.

15.3.7.2.2 GPT1 Auxiliary Timers T2/T4 Control Registers

T2CON
Timer T2 Control Register **(08_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T2RDIR	**T2CHDIR**	**T2EDGE**	**T2IRDIS**	**RES**		**T2RC**	**T2UDE**	**T2UD**	**T2R**	**T2M**			**T2I**		
rh	rwh	rwh	rw	r		rw	rw	rw	rw	rw			rw		

Field	Bits	Type	Description
T2I	[2:0]	rw	**Timer T2 Input Parameter Selection** Depends on the operating mode, see respective sections for encoding: **Table 95** for Timer Mode and Gated Timer Mode **Table 98** for Counter Mode **Table 99** for Incremental Interface Mode **Table 98** for Reload Mode **Table 97** for Capture Mode
T2M	[5:3]	rw	**Timer T2 Mode Control (Basic Operating Mode)** 000_B Timer Mode 001_B Counter Mode 010_B Gated Timer Mode with gate active low 011_B Gated Timer Mode with gate active high 100_B Reload Mode 101_B Capture Mode 110_B Incremental Interface Mode (Rotation Detection Mode) 111_B Incremental Interface Mode (Edge Detection Mode)
T2R	6	rw	**Timer T2 Run Bit** 0_B Timer T2 stops 1_B Timer T2 runs *Note: This bit only controls timer T2 if bit T2RC = 0.*
T2UD	7	rw	**Timer T2 Up/Down Control**[1)] 0_B Timer T2 counts up 1_B Timer T2 counts down *Note: This bit only controls count direction of T2 if bit T2UDE = 0.*
T2UDE	8	rw	**Timer T2 External Up/Down Enable**[1)] 0_B Count direction is controlled by bit T2UD; input T2EUD is disconnected 1_B Count direction is controlled by input T2EUD

Field	Bits	Type	Description
T2RC	9	rw	**Timer T2 Remote Control** 0_B Timer T2 is controlled by its own run bit T2R 1_B Timer T2 is controlled by the run bit T3R of core timer T3, not by bit T2R
T2IRDIS	12	rw	**Timer T2 Interrupt Disable** 0_B Interrupt generation for T2CHDIR and T2EDGE interrupts in Incremental Interface Mode is enabled 1_B Interrupt generation for T2CHDIR and T2EDGE interrupts in Incremental Interface Mode is disabled
T2EDGE	13	rwh	**Timer T2 Edge Detection** The bit is set each time a count edge is detected. T2EDGE must be cleared by software. 0_B No count edge was detected 1_B A count edge was detected
T2CHDIR	14	rwh	**Timer T2 Count Direction Change** The bit is set each time the count direction of timer T2 changes. T2CHDIR must be cleared by software. 0_B No change in count direction was detected 1_B A change in count direction was detected
T2RDIR	15	rh	**Timer T2 Rotation Direction** 0_B Timer T2 counts up 1_B Timer T2 counts down
RES	[11:10]	r	**Reserved** Read as 0; should be written with 0.

1) See **Table 104** for encoding of bits T2UD and T2UDE.

T4CON
Timer T4 Control Register **(10_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T4RDIR	T4CHDIR	T4EDGE	T4IRDIS	CLRT3EN	CLRT2EN	T4RC	T4UDE	T4UD	T4R	T4M			T4I		
rh	rwh	rwh	rw	rw	rw	rw	rw	rw	rw	rw			rw		

Field	Bits	Type	Description
T4I	[2:0]	rw	**Timer T4 Input Parameter Selection** Depends on the operating mode, see respective sections for encoding: **Table 95** for Timer Mode and Gated Timer Mode **Table 98** for Counter Mode **Table 99** for Incremental Interface Mode **Table 98** for Reload Mode **Table 97** for Capture Mode
T4M	[5:3]	rw	**Timer T4 Mode Control (Basic Operating Mode)** 000_B Timer Mode 001_B Counter Mode 010_B Gated Timer Mode with gate active low 011_B Gated Timer Mode with gate active high 100_B Reload Mode 101_B Capture Mode 110_B Incremental Interface Mode (Rotation Detection Mode) 111_B Incremental Interface Mode (Edge Detection Mode)
T4R	6	rw	**Timer T4 Run Bit** 0_B Timer T4 stops 1_B Timer T4 runs *Note: This bit only controls timer T4 if bit T4RC = 0.*
T4UD	7	rw	**Timer T4 Up/Down Control**[1)] 0_B Timer T4 counts up 1_B Timer T4 counts down *Note: This bit only controls count direction of T4 if bit T4UDE = 0.*
T4UDE	8	rw	**Timer T4 External Up/Down Enable**[1)] 0_B Count direction is controlled by bit T4UD; input T4EUD is disconnected 1_B Count direction is controlled by input T4EUD
T4RC	9	rw	**Timer T4 Remote Control** 0_B Timer T4 is controlled by its own run bit T4R 1_B Timer T4 is controlled by the run bit T3R of core timer T3, but not by bit T4R

Field	Bits	Type	Description
CLRT2EN	10	rw	**Clear Timer T2 Enable** Enables the automatic clearing of timer T2 upon a falling edge of the selected T4EUD input. 0_B No effect of T4EUD on timer T2 1_B A falling edge on T4EUD clears timer T2
CLRT3EN	11	rw	**Clear Timer T3 Enable** Enables the automatic clearing of timer T3 upon a falling edge of the selected T4IN input. 0_B No effect of T4IN on timer T3 1_B A falling edge on T4IN clears timer T3
T4IRDIS	12	rw	**Timer T4 Interrupt Disable** 0_B Interrupt generation for T4CHDIR and T4EDGE interrupts in Incremental Interface Mode is enabled 1_B Interrupt generation for T4CHDIR and T4EDGE interrupts in Incremental Interface Mode is disabled
T4EDGE	13	rwh	**Timer T4 Edge Detection** The bit is set each time a count edge is detected. T4EDGE has to be cleared by software. 0_B No count edge was detected 1_B A count edge was detected
T4CHDIR	14	rwh	**Timer T4 Count Direction Change** The bit is set each time the count direction of timer T4 changes. T4CHDIR must be cleared by software. 0_B No change in count direction was detected 1_B A change in count direction was detected
T4RDIR	15	rh	**Timer T4 Rotation Direction** 0_B Timer T4 counts up 1_B Timer T4 counts down

1) See **Table 104** for encoding of bits T4UD and T4UDE.

15.3.7.2.3 Encoding of GPT1 Timer Count Direction Control

Table 94 GPT1 Timer Count Direction Control

Pin TxEUD	Bit TxUDE	Bit TxUD	Count Direction	Bit TxRDIR
X	0	0	Count Up	0
X	0	1	Count Down	1
0	1	0	Count Up	0
1	1	0	Count Down	1
0	1	1	Count Down	1
1	1	1	Count Up	0

15.3.7.2.4 Timer Mode and Gated Timer Mode: Encoding of GPT1 Overall Prescaler Factor

Table 95 GPT1 Overall Prescaler Factors for Internal Count Clock (Timer Mode and Gated Timer Mode)

Individual Prescaler for Tx	Common Prescaler for Module Clock[1)]			
	BPS1 = 01_B	**BPS1 = 00_B**	**BPS1 = 11_B**	**BPS1 = 10_B**
TxI = 000_B	4	8	16	32
TxI = 001_B	8	16	32	64
TxI = 010_B	16	32	64	128
TxI = 011_B	32	64	128	256
TxI = 100_B	64	128	256	512
TxI = 101_B	128	256	512	1024
TxI = 110_B	256	512	1024	2048
TxI = 111_B	512	1024	2048	4096

1) Please note the non-linear encoding of bitfield BPS1.

15.3.7.2.5 Counter Mode: Encoding of GPT1 Input Edge Selection

Table 96 GPT1 Core Timer T3 Input Edge Selection (Counter Mode)

T3I	Triggering Edge for Counter Increment/Decrement
000_B	None. Counter T3 is disabled
001_B	Positive transition (rising edge) on T3IN
010_B	Negative transition (falling edge) on T3IN
011_B	Any transition (rising or falling edge) on T3IN
$1XX_B$	Reserved. Do not use this combination

Table 97 GPT1 Auxiliary Timers T2/T4 Input Edge Selection (Capture Mode)

T2I/T4I	Triggering Edge for Counter Increment/Decrement
000_B	None. Counter Tx is disabled
001_B	Positive transition (rising edge) on TxIN
010_B	Negative transition (falling edge) on TxIN
011_B	Any transition (rising or falling edge) on TxIN
$1XX_B$	Reserved. Do not use this combination

Table 98 GPT1 Auxiliary Timers T2/T4 Input Edge Selection (Counter Mode, Reload Mode)

T2I/T4I	Triggering Edge for Counter Increment/Decrement
$X00_B$	None. Counter Tx is disabled
001_B	Positive transition (rising edge) on TxIN
010_B	Negative transition (falling edge) on TxIN
011_B	Any transition (rising or falling edge) on TxIN
101_B	Positive transition (rising edge) of T3 toggle latch T3OTL
110_B	Negative transition (falling edge) of T3 toggle latch T3OTL
111_B	Any transition (rising or falling edge) of T3 toggle latch T3OTL

15.3.7.2.6 Incremental Interface Mode: Encoding of Input Edge Selection

Table 99 GPT1 Core Timer T3 Input Edge Selection (Incremental Interface Mode)

T3I	Triggering Edge for Counter Increment/Decrement
000_B	None. Counter T3 stops.
001_B	Any transition (rising or falling edge) on T3IN.
010_B	Any transition (rising or falling edge) on T3EUD.
011_B	Any transition (rising or falling edge) on any T3 input (T3IN or T3EUD).
$1XX_B$	Reserved. Do not use this combination.

15.3.7.3 GPT1 Timer Interrupt Control Registers

The Interrupt Control and Status register are located in the SCU.**GPT12IEN** and SCU.**GPT12IRC**.

15.4 Timer Block GPT2

From a programmer's point of view, the GPT2 block is represented by a set of SFRs as summarized below. Those portions of port and direction registers which are used for alternate functions by the GPT2 block are shaded.

Data Registers	Control Registers	Interrupt Registers	Miscellaneous
T5	T5CON	GPT12IEN	GPT12PISEL
T6	T6CON	GPT12IRC	PISEL
CAPREL		GPT12ICLR	ID

Tx	GPT2 timer x data register
TxCON	GPT2 timer x control register
CAPREL	GPT2 capture and reload register
GPT12IEN	GPT12 interrupt enable register (located in SCU)
GPT12IRC	GPT12 interrupt status register (located in SCU)
GPT12ICLR	GPT12 interrupt status clear register (located in SCU)
GPT12PISEL	GPT12 extended pin select register (located in SCU)
PISEL	GPT12 local pin select register
ID	GPT module identification register

Figure 99 SFRs Associated with Timer Block GPT2

Both timers of block GPT2 (T5, T6) can run in one of 3 basic modes: Timer Mode, Gated Timer Mode, or Counter Mode. All timers can count up or down. Each timer of GPT2 is controlled by a separate control register TxCON.

Each timer has an input pin TxIN (alternate pin function) associated with it, which serves as the gate control in Gated Timer Mode, or as the count input in Counter Mode. The count direction (up/down) may be programmed via software or may be dynamically altered by a signal at the External Up/Down control input TxEUD (alternate pin function). An overflow/underflow of core timer T6 is indicated by the Output Toggle Latch T6OTL, whose state may be output on the associated pin T6OUT (alternate pin function). The auxiliary timer T5 may additionally be concatenated with core timer T6 (through T6OTL).

The Capture/Reload register CAPREL can be used to capture the contents of timer T5, or to reload timer T6. A special mode facilitates the use of register CAPREL for both functions at the same time. This mode allows frequency multiplication. The capture function is triggered by the input pin CAPIN, or by GPT1 timer's T3 input lines T3IN and T3EUD. The reload function is triggered by an overflow or underflow of timer T6. Overflows/underflows of timer T6 may also clock the timers of the CAPCOM units.

The current contents of each timer can be read or modified by the CPU by accessing the corresponding timer count registers T5 or T6, located in the SFR space (see **Section 15.4.8.1**). When any of the timer registers is written to by the CPU in the state immediately preceding a timer increment, decrement, reload, or capture operation, the CPU write operation has priority in order to guarantee correct results.

The interrupts of GPT2 are controlled through the SCU.**GPT12IEN** and SCU.**GPT12IRC**. These registers are not part of the GPT2 block.

The input and output lines of GPT2 are connected to pins. The control registers for the port functions are located in the respective port modules.

*Note: The timing requirements for external input signals can be found in **Section 15.4.6**, **Section 15.7.1** summarizes the module interface signals, including pins.*

15.4.1 GPT2 Core Timer T6 Control

The current contents of the core timer T6 are reflected by its count register T6. This register can also be written to by the CPU, for example, to set the initial start value.

The core timer T6 is configured and controlled via its control register T6CON.

Timer T6 Run Control

The core timer T6 can be started or stopped by software through bit T6R (timer T6 run bit). This bit is relevant in all operating modes of T6. Setting bit T6R will start the timer, clearing bit T6R stops the timer.

In Gated Timer Mode, the timer will only run if T6R = 1 and the gate is active (high or low, as programmed).

Note: When bit T5RC in timer control register T5CON is set, bit T6R will also control (start and stop) the Auxiliary Timer T5.

Count Direction Control

The count direction of the GPT2 timers (core timer and auxiliary timer) can be controlled either by software or by the external input pin TxEUD (Timer Tx External Up/Down Control Input). These options are selected by bits TxUD and TxUDE in the respective control register TxCON. When the up/down control is provided by software (bit TxUDE = 0), the count direction can be altered by setting or clearing bit TxUD. When bit TxUDE = 1, pin TxEUD is selected to be the controlling source of the count direction. However, bit TxUD can still be used to reverse the actual count direction, as shown in **Table 104**. The count direction can be changed regardless of whether or not the timer is running.

Note: When pin TxEUD is used as external count direction control input, it must be configured as input.

Timer T6 Output Toggle Latch

The overflow/underflow signal of timer T6 is connected to a block named 'Toggle Latch', shown in the Timer Mode diagrams. **Figure 100** illustrates the details of this block. An overflow or underflow of T6 will clock two latches: The first latch represents bit T6OTL in control register T6CON. The second latch is an internal latch toggled by T6OTL's output. Both latch outputs are connected to the input control block of the auxiliary timer T5. The output level of the shadow latch will match the output level of T6OTL, but is delayed by one clock cycle. When the T6OTL value changes, this will result in a temporarily different output level from T6OTL and the shadow latch, which can trigger the selected count event in T5.

When software writes to T6OTL, both latches are set or cleared simultaneously. In this case, both signals to the auxiliary timers carry the same level and no edge will be detected. Bit T6OE (overflow/underflow output enable) in register T6CON enables the state of T6OTL to be monitored via an external pin T6OUT. When T6OTL is linked to an external port pin (must be configured as output), T6OUT can be used to control external HW. If T6OE = 1, pin T6OUT outputs the state of T6OTL. If T6OE = 0, pin T6OUT outputs a high level (while it selects the timer output signal).

As can be seen from **Figure 100**, when latch T6OTL is modified by software to determine the state of the output line, also the internal shadow latch is set or cleared accordingly. Therefore, no trigger condition is detected by T5 in this case.

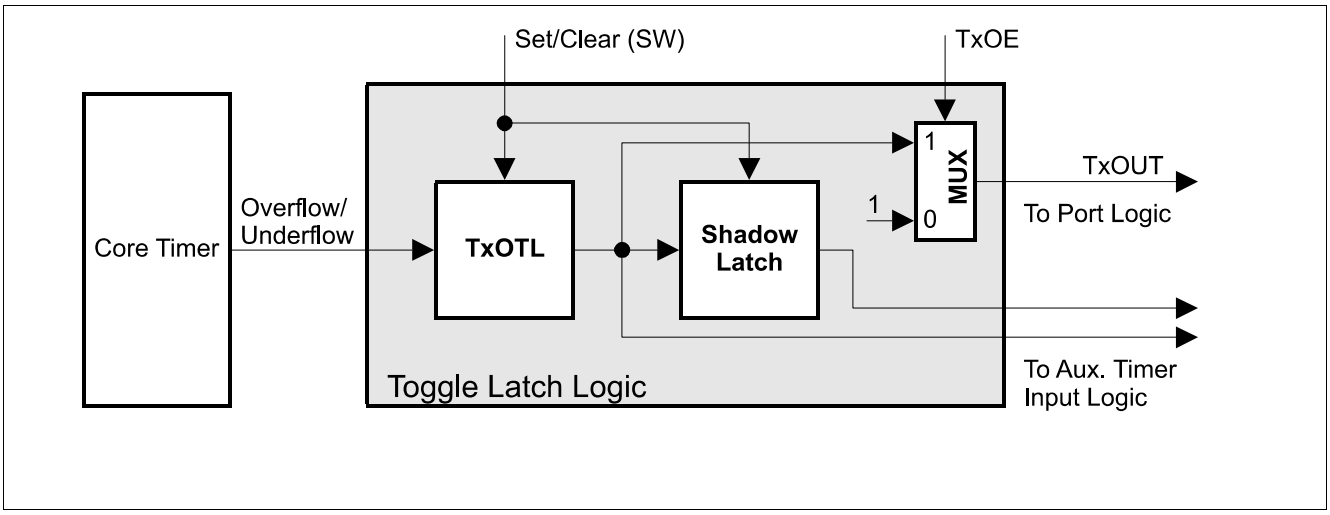


Figure 100 Block Diagram of the Toggle Latch Logic of Core Timer T6 (x = 6)

Note: T6 is also used to clock the timers in the CAPCOM units. For this purpose, there is a direct internal connection between the T6 overflow/underflow line and the CAPCOM timers (signal T6OUF).

15.4.2 GPT2 Core Timer T6 Operating Modes

Timer T6 can operate in one of several modes.

Timer T6 in Timer Mode

Timer mode for the core timer T6 is selected by setting bitfield T6M in register T6CON to 000_B. In this mode, T6 is clocked with the module's input clock f_{GPT} divided by two programmable prescalers controlled by bitfields BPS2 and T6I in register T6CON. Please see **Section 15.4.6** for details on the input clock options.

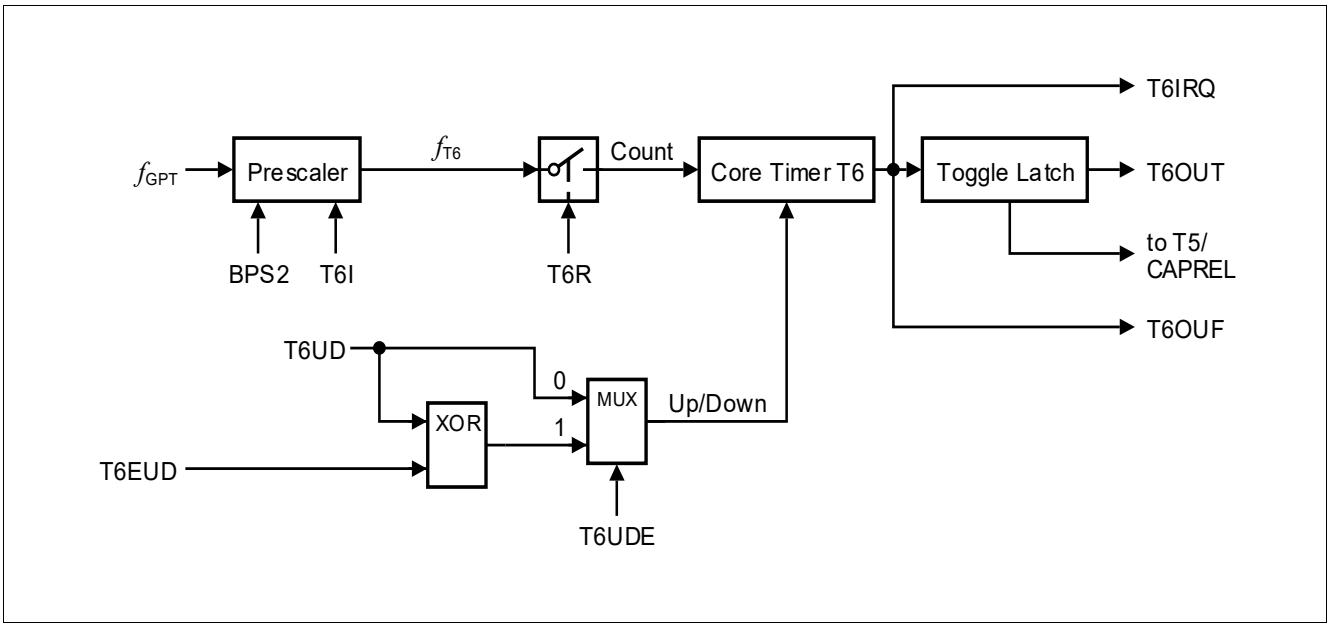


Figure 101 Block Diagram of Core Timer T6 in Timer Mode

Timer T6 in Gated Timer Mode

Gated Timer Mode for the core timer T6 is selected by setting bitfield T6M in register T6CON to 010_B or 011_B. Bit T6M.0 (T6CON.3) selects the active level of the gate input. The same options for the input frequency are available in Gated Timer Mode as in Timer Mode (see **Section 15.4.6**). However, the input clock to the timer in this mode is gated by the external input pin T6IN (Timer T6 External Input).
To enable this operation, the associated pin T6IN must be configured as input.

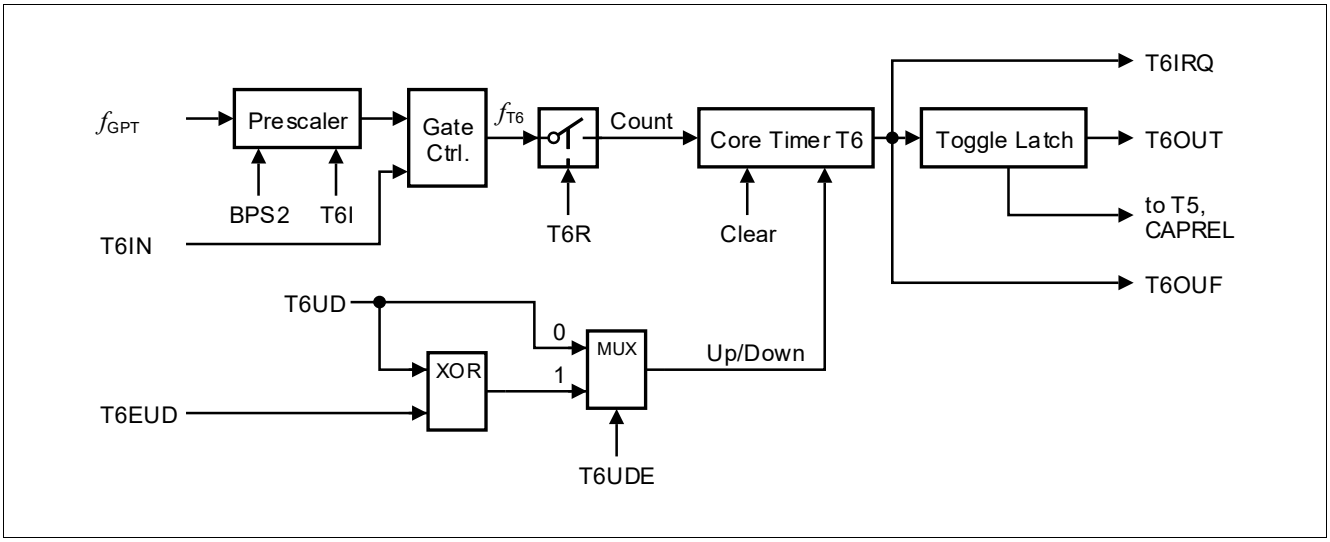


Figure 102 Block Diagram of Core Timer T6 in Gated Timer Mode

If T6M = 010_B, the timer is enabled when T6IN shows a low level. A high level at this line stops the timer. If T6M = 011_B, line T6IN must have a high level in order to enable the timer. Additionally, the timer can be turned on or off by software using bit T6R. The timer will only run if T6R is 1 and the gate is active. It will stop if either T6R is 0 or the gate is inactive.

Note: A transition of the gate signal at pin T6IN does not cause an interrupt request.

Timer T6 in Counter Mode

Counter Mode for the core timer T6 is selected by setting bitfield T6M in register T6CON to 001_B. In Counter Mode, timer T6 is clocked by a transition at the external input pin T6IN. The event causing an increment or decrement of the timer can be a positive, a negative, or both a positive and a negative transition at this line. Bitfield T6I in control register T6CON selects the triggering transition (see **Table 106**).

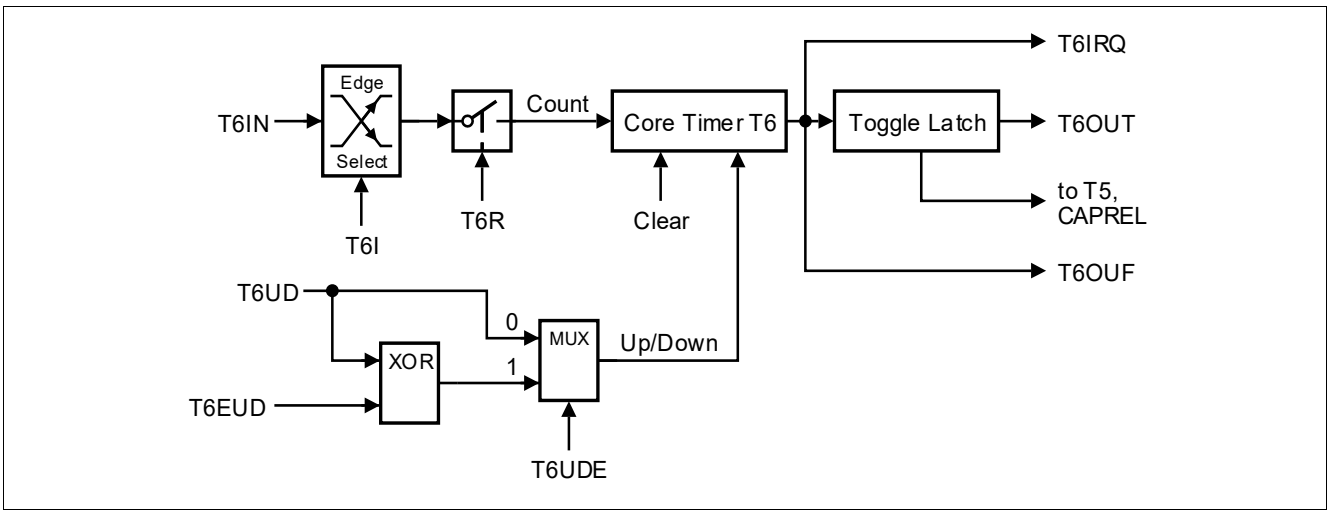


Figure 103 Block Diagram of Core Timer T6 in Counter Mode

For Counter Mode operation, pin T6IN must be configured as input. The maximum input frequency allowed in Counter Mode depends on the selected prescaler value. To ensure that a transition of the count input signal applied to T6IN is recognized correctly, its level must be held high or low for a minimum number of module clock cycles before it changes. This information can be found in **Section 15.4.6**.

15.4.3 GPT2 Auxiliary Timer T5 Control

Auxiliary timer T5 can be configured for Timer Mode, Gated Timer Mode, or Counter Mode with the same options for the timer frequencies and the count signal as the core timer T6. In addition to these 3 counting modes, the auxiliary timer can be concatenated with the core timer. The contents of T5 may be captured to register CAPREL upon an external or an internal trigger. The start/stop function of the auxiliary timers can be remotely controlled by the T6 run control bit. Several timers may thus be controlled synchronously.

The current contents of the auxiliary timer are reflected by its count register T5. This register can also be written to by the CPU, for example, to set the initial start value.

The individual configurations for timer T5 are determined by its control register T5CON. Some bits in this register also control the function of the CAPREL register. Note that functions which are present in all timers of block GPT2 are controlled in the same bit positions and in the same manner in each of the specific control registers.

Note: The auxiliary timer has no output toggle latch and no alternate output function.

Timer T5 Run Control

The auxiliary timer T5 can be started or stopped by software in two different ways:

- Through the associated timer run bit (T5R). In this case it is required that the respective control bit T5RC = 0.
- Through the core timer's run bit (T6R). In this case the respective remote control bit must be set (T5RC = 1).

The selected run bit is relevant in all operating modes of T5. Setting the bit will start the timer, clearing the bit stops the timer.

In Gated Timer Mode, the timer will only run if the selected run bit is set and the gate is active (high or low, as programmed).

Note: If remote control is selected T6R will start/stop timer T6 and the auxiliary timer T5 synchronously.

15.4.4 GPT2 Auxiliary Timer T5 Operating Modes

The operation of the auxiliary timer in the basic operating modes is almost identical with the core timer's operation, with very few exceptions. Additionally, some combined operating modes can be selected.

Timer T5 in Timer Mode

Timer Mode for the auxiliary timer T5 is selected by setting its bitfield T5M in register T5CON to 000_B.

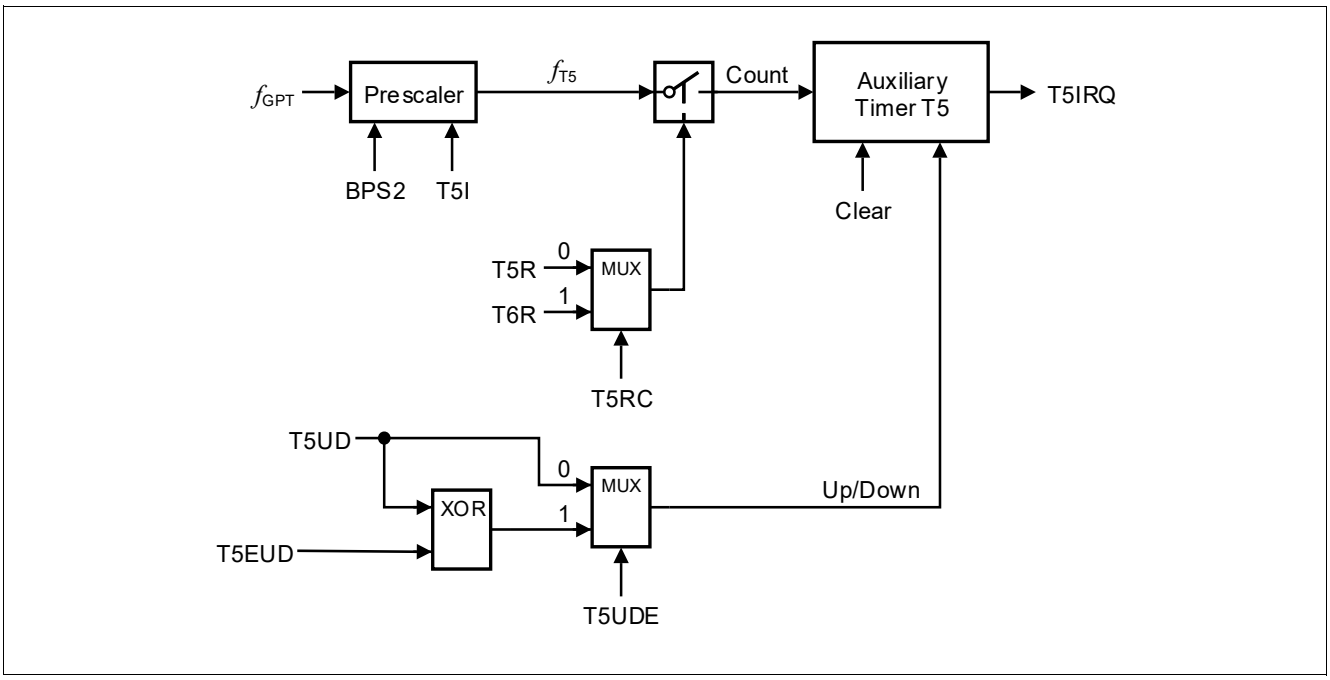


Figure 104 Block Diagram of Auxiliary Timer T5 in Timer Mode

Timer T5 in Gated Timer Mode

Gated Timer Mode for the auxiliary timer T5 is selected by setting bitfield T5M in register T5CON to 010_B or 011_B. Bit T5M.0 (T5CON.3) selects the active level of the gate input.

Note: A transition of the gate signal at line T5IN does not cause an interrupt request.

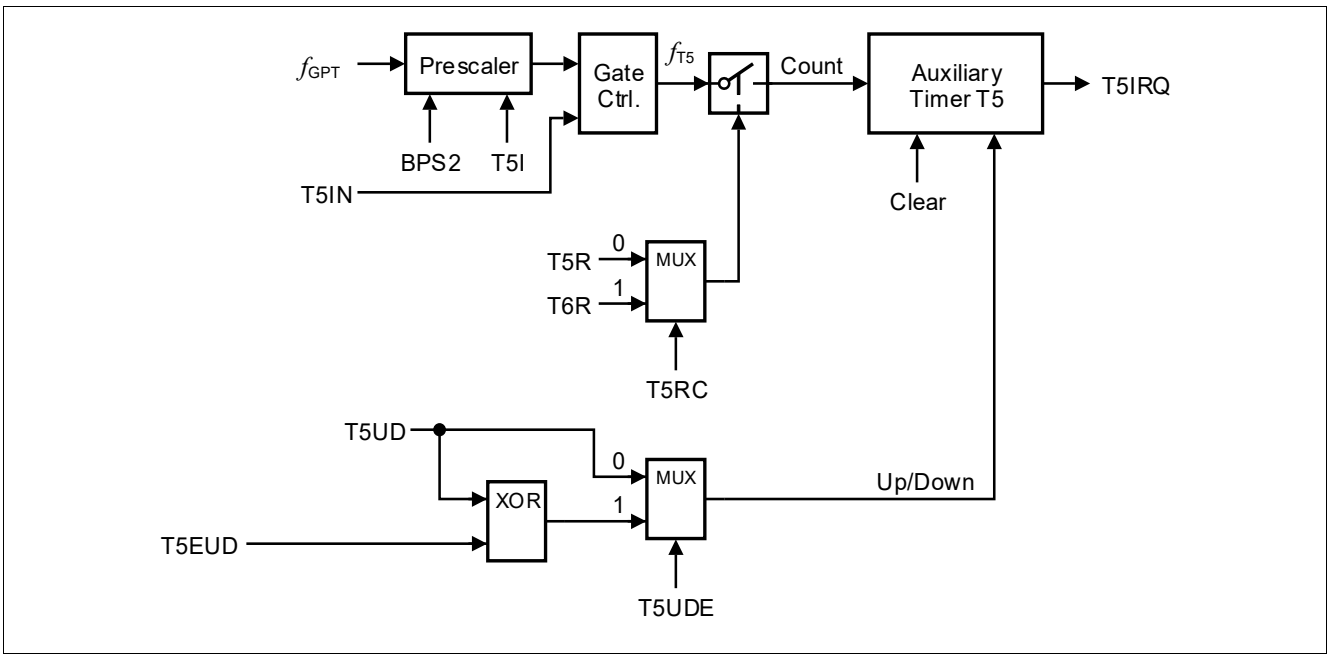


Figure 105 Block Diagram of Auxiliary Timer T5 in Gated Timer Mode

Note: There is no output toggle latch for T5.
Start/stop of the auxiliary timer can be controlled locally or remotely.

Timer T5 in Counter Mode

Counter Mode for auxiliary timer T5 is selected by setting bitfield T5M in register T5CON to 001_B. In Counter Mode, the auxiliary timer can be clocked either by a transition at its external input line T5IN, or by a transition of timer T6's toggle latch T6OTL. The event causing an increment or decrement of a timer can be a positive, a negative, or both a positive and a negative transition at either the respective input pin or at the toggle latch. Bitfield T5I in control register T5CON selects the triggering transition (see **Table 107**).

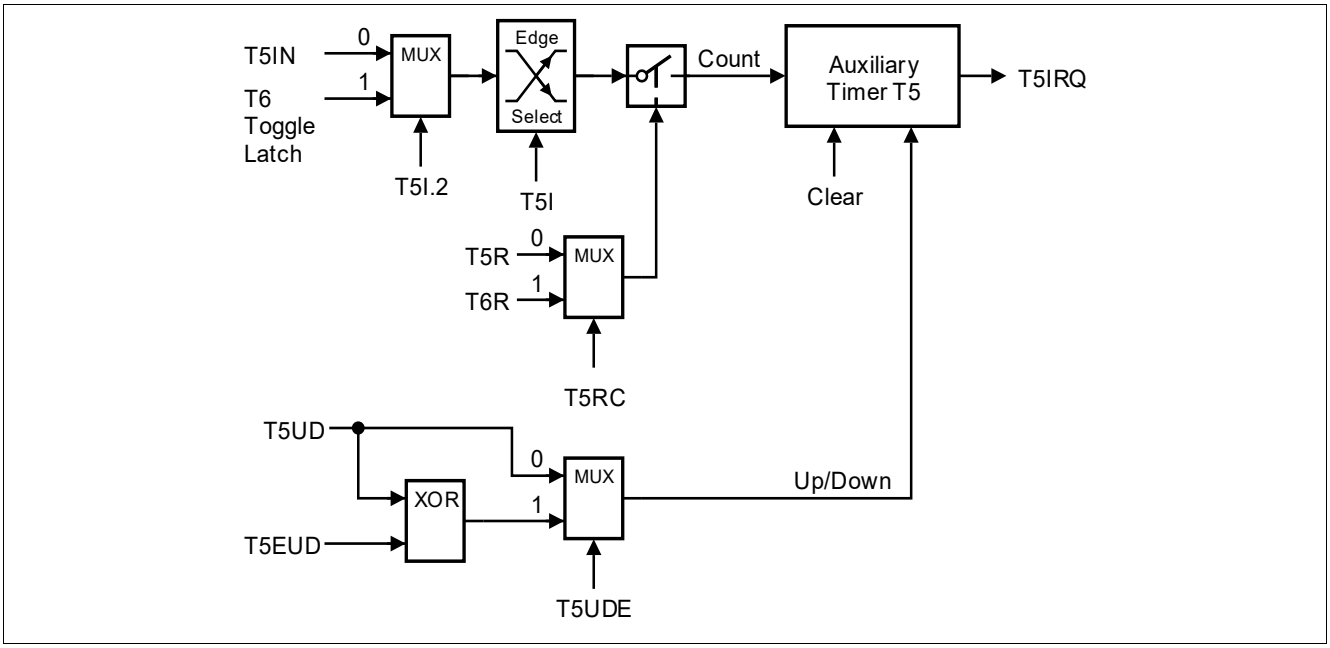


Figure 106 Block Diagram of Auxiliary Timer T5 in Counter Mode

Note: Only state transitions of T6OTL which are caused by the overflows/underflows of T6 will trigger the counter function of T5. Modifications of T6OTL via software will NOT trigger the counter function of T5.

For counter operation, pin T5IN must be configured as input. The maximum input frequency allowed in Counter Mode depends on the selected prescaler value. To ensure that a transition of the count input signal applied to T5IN is recognized correctly, its level must be held high or low for a minimum number of module clock cycles before it changes. This information can be found in **Section 15.4.6**.

Timer Concatenation

Using the toggle bit T6OTL as a clock source for the auxiliary timer in Counter Mode concatenates the core timer T6 with the auxiliary timer T5. This concatenation forms either a 32-bit or a 33-bit timer/counter, depending on which transition of T6OTL is selected to clock the auxiliary timer.

- **32-bit Timer/Counter:** If both a positive and a negative transition of T6OTL are used to clock the auxiliary timer, this timer is clocked on every overflow/underflow of the core timer T6. Thus, the two timers form a 32-bit timer.
- **33-bit Timer/Counter:** If either a positive or a negative transition of T6OTL is selected to clock the auxiliary timer, this timer is clocked on every second overflow/underflow of the core timer T6. This configuration forms a 33-bit timer (16-bit core timer + T6OTL + 16-bit auxiliary timer).
  As long as bit T6OTL is not modified by software, it represents the state of the internal toggle latch, and can be regarded as part of the 33-bit timer.

The count directions of the two concatenated timers are not required to be the same. This offers a wide variety of different configurations.

T6, which represents the low-order part of the concatenated timer, can operate in Timer Mode, Gated Timer Mode or Counter Mode in this case.

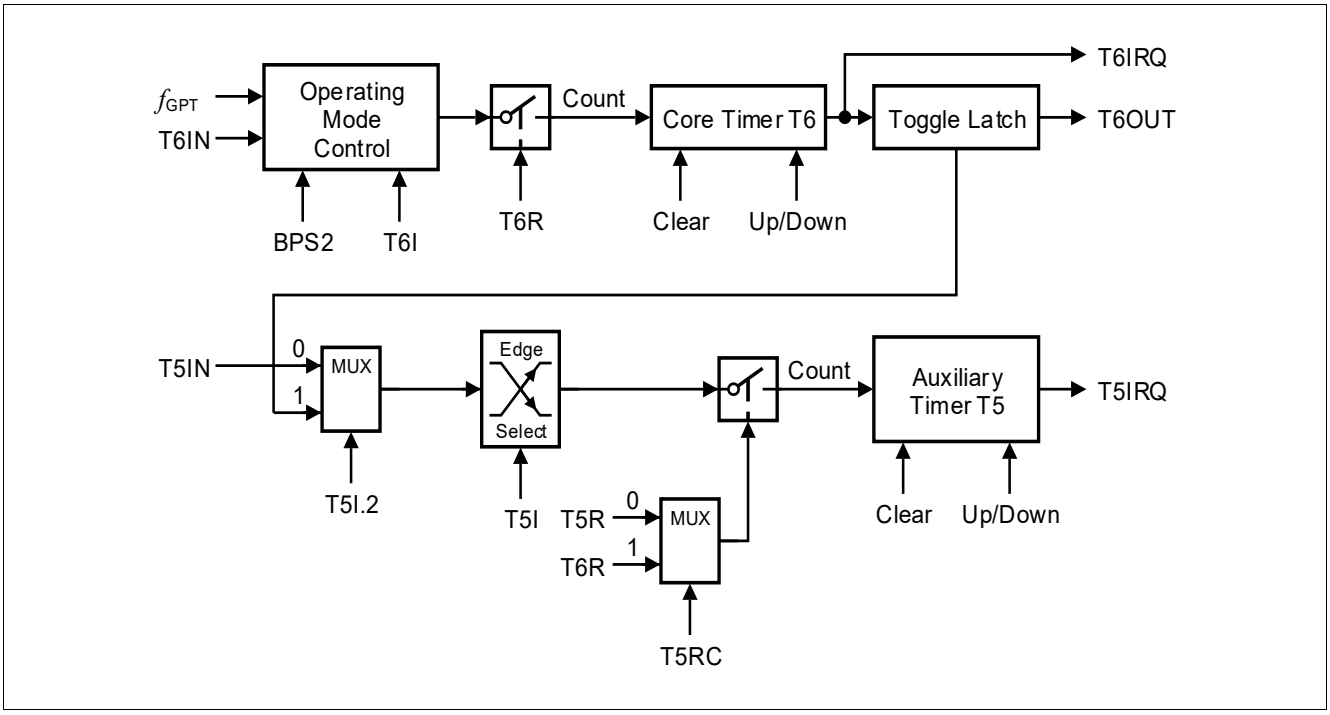


Figure 107 Concatenation of Core Timer T6 and Auxiliary Timer T5

15.4.5 GPT2 Register CAPREL Operating Modes

The Capture/Reload register CAPREL can be used to capture the contents of timer T5, or to reload timer T6. A special mode facilitates the use of register CAPREL for both functions at the same time. This mode allows frequency multiplication. The capture function is triggered by CAPIN, by T3IN and T3EUD, or by read GPT1 timers. The reload function is triggered by an overflow or underflow of timer T6.

In addition to the capture function, the capture trigger signal can also be used to clear the contents of timers T5 and T6 individually.

The functions of register CAPREL are controlled via several bit(field)s in the timer control registers T5CON and T6CON.

Capture/Reload Register CAPREL in Capture Mode

Capture mode for register CAPREL is selected by setting bit T5SC in control register T5CON (set bitfield CI in register T5CON to a non-zero value to select a trigger signal). In capture mode, the contents of the auxiliary timer T5 are latched into register CAPREL in response to a signal transition at the selected external input pin(s). Bit CT3 selects the external input line CAPIN or the input lines T3IN and/or T3EUD of GPT1 timer T3 as the source for a capture trigger. Either a positive, a negative, or both a positive and a negative transition at line CAPIN can be selected to trigger the capture function, or transitions on input T3IN or input T3EUD or both inputs, T3IN and T3EUD. The active edge is controlled by bitfield CI in register T5CON. **Table 100** summarizes these options.

Table 100 CAPREL Register Input Edge Selection

CT3	CI	Triggering Signal/Edge for Capture Mode
X	00_B	None. Capture Mode is disabled.
0	01_B	Positive transition (rising edge) on CAPIN.[1)]
0	10_B	Negative transition (falling edge) on CAPIN.
0	11_B	Any transition (rising or falling edge) on CAPIN.
1	01_B	Any transition (rising or falling edge) on T3IN.
1	10_B	Any transition (rising or falling edge) on T3EUD.
1	11_B	Any transition (rising or falling edge) on T3IN or T3EUD.

1) Rising edge must be selected if capturing is triggered by the internal GPT1 read signals (see register PISEL and **"Combined Capture Modes" on Page 471**).

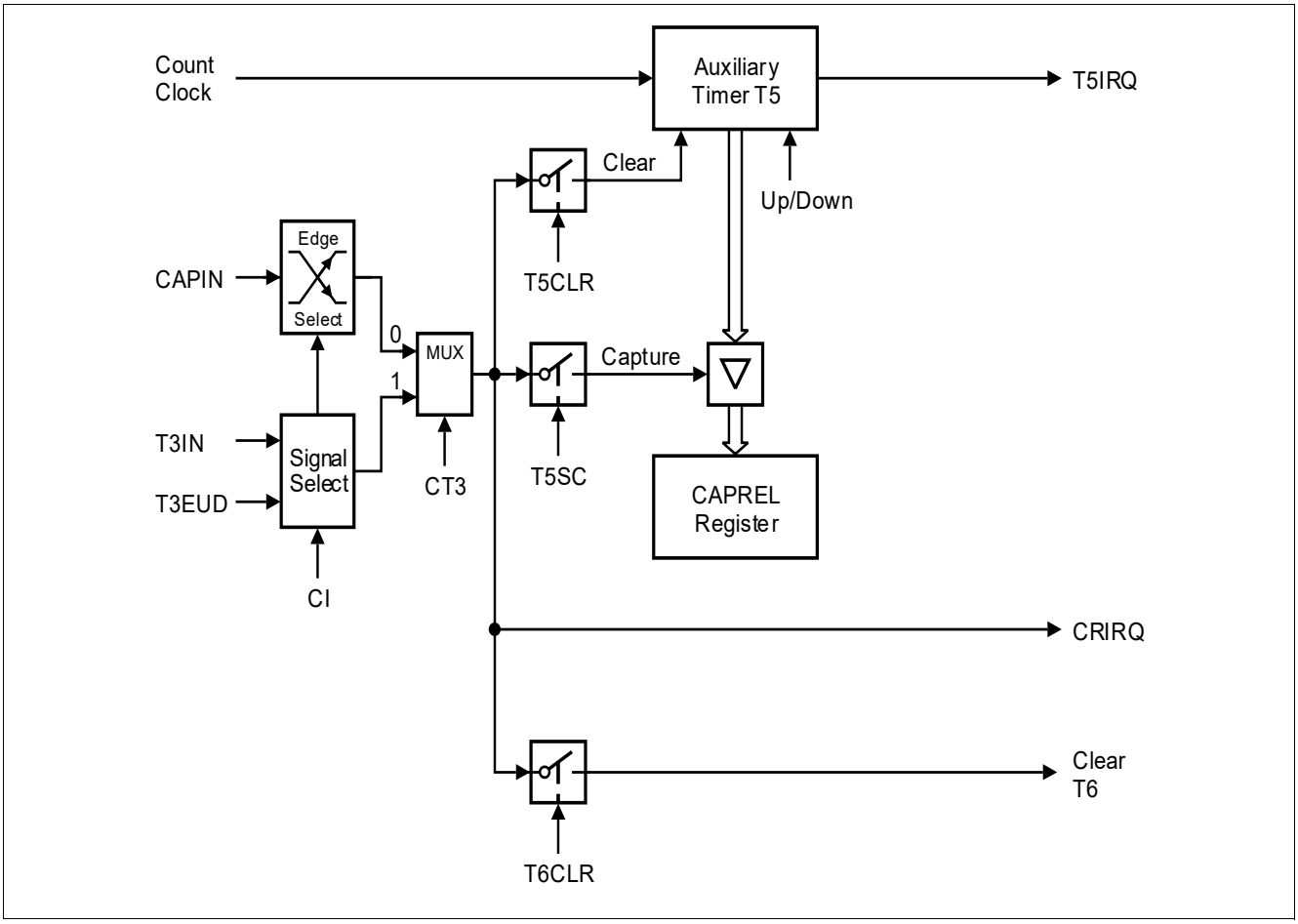


Figure 108 Capture/Reload Register CAPREL in Capture Mode

When a selected trigger is detected, the contents of the auxiliary timer T5 are latched into register CAPREL and the interrupt request line CRIRQ is activated. The same event can optionally clear timer T5 and/or timer T6. This option is enabled by bit T5CLR in register T5CON and bit T6CLR in register T6CON, respectively. If TxCLR = 0 the contents of timer Tx is not affected by a capture. If TxCLR = 1 timer Tx is cleared after the current timer T5 value has been latched into register CAPREL.

*Note: Bit T5SC only controls whether or not a capture is performed. If T5SC is cleared the external input pin(s) can still be used to clear timer T5 and/or T6, or as external interrupt input(s). This interrupt is controlled by the CAPREL interrupt control register SCU.**GPT12IEN** and SCU.**GPT12IRC**.*

When capture triggers T3IN or T3EUD are enabled (CT3 = 1), register CAPREL captures the contents of T5 upon transitions of the selected input(s). These values can be used to measure T3's input signals. This is useful, for example, when T3 operates in Incremental Interface Mode, in order to derive dynamic information (speed, acceleration) from the input signals.

For capture mode operation, the selected pins CAPIN, T3IN, or T3EUD must be configured as input. To ensure that a transition of a trigger input signal applied to one of these inputs is recognized correctly, its level must be held high or low for a minimum number of module clock cycles, detailed in **Section 15.4.6**.

Capture/Reload Register CAPREL in Reload Mode

Reload mode for register CAPREL is selected by setting bit T6SR in control register T6CON. In reload mode, the core timer T6 is reloaded with the contents of register CAPREL, triggered by an overflow or underflow of T6. This will not activate the interrupt request line CRIRQ associated with the CAPREL register. However, interrupt request line T6IRQ will be activated, indicating the overflow/underflow of T6.

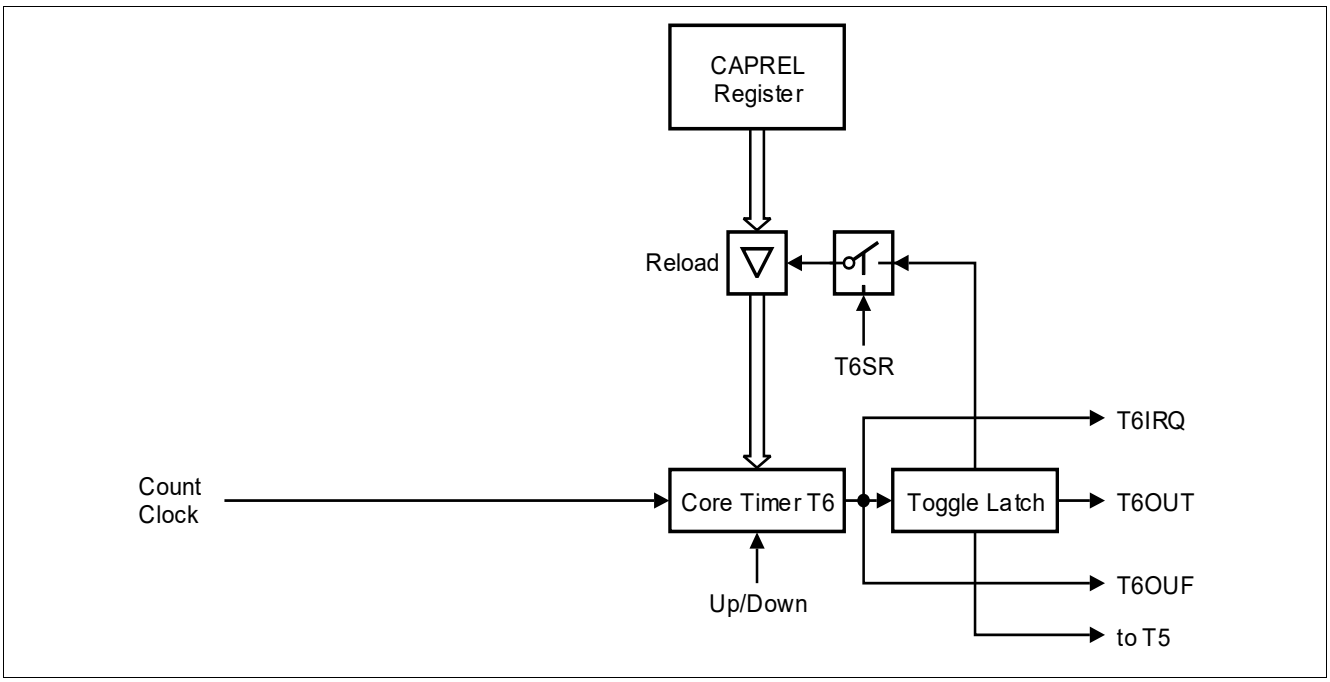


Figure 109 Capture/Reload Register CAPREL in Reload Mode

Capture/Reload Register CAPREL in Capture-And-Reload Mode

Since the reload function and the capture function of register CAPREL can be enabled individually by bits T5SC and T6SR, the two functions can be enabled simultaneously by setting both bits. This feature can be used to generate an output frequency that is a multiple of the input frequency.

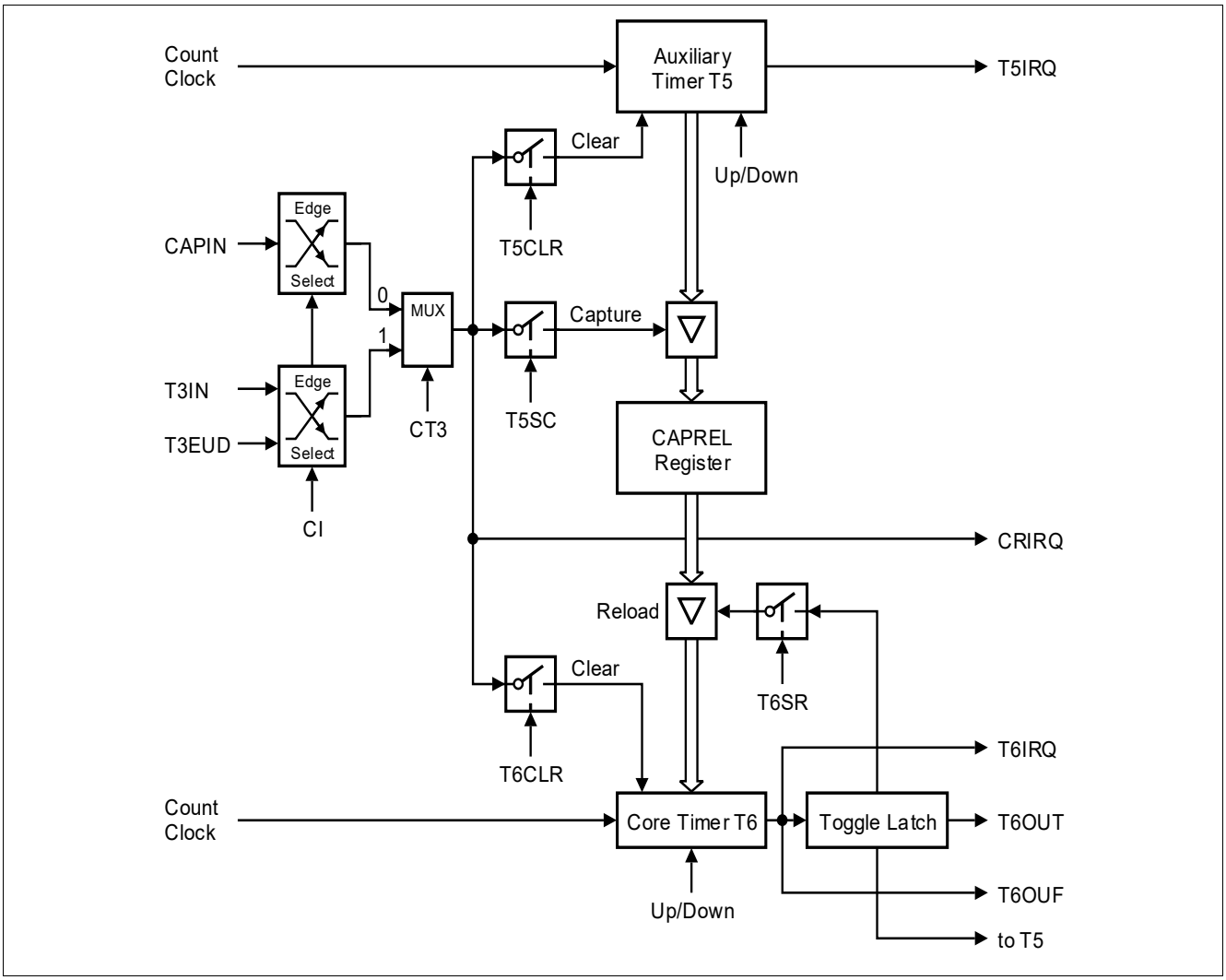


Figure 110 Capture/Reload Register CAPREL in Capture-And-Reload Mode

This combined mode can be used to detect consecutive external events which may occur aperiodically, but where a finer resolution, that means, more ‘ticks’ within the time between two external events is required.

For this purpose, the time between the external events is measured using timer T5 and the CAPREL register. Timer T5 runs in Timer Mode counting up with a frequency of e.g. $f_{GPT}/32$. The external events are applied to pin CAPIN. When an external event occurs, the contents of timer T5 are latched into register CAPREL and timer T5 is cleared (T5CLR = 1). Thus, register always contains the correct time between two events, measured in timer T5 increments. Timer T6, which runs in Timer Mode counting down with a frequency of e.g. $f_{GPT}/4$, uses the value in register CAPREL to perform a reload on underflow. This means, the value in register CAPREL represents the time between two underflows of timer T6, now measured in timer T6 increments. Since (in this example) timer T6 runs 8 times faster than timer T5, it will underflow 8 times within the time between two external events. Thus, the underflow signal of timer T6 generates 8 ‘ticks’. Upon each underflow, the interrupt request line T6IRQ will be activated and bit T6OTL will be toggled. The state of T6OTL may be output on pin T6OUT. This signal has 8 times more transitions than the signal which is applied to pin CAPIN.

Note: The underflow signal of Timer T6 can furthermore be used to clock one or more of the timers of the CAPCOM units, which gives the user the possibility to set compare events based on a finer resolution than that of the external events. This connection is accomplished via signal T6OUF.

Capture Correction

A certain deviation of the output frequency is generated by the fact that timer T5 will count actual time units (e.g. T5 running at 1 MHz will count up to the value $64_H/100_D$ for a 10 kHz input signal), while T6OTL will only toggle upon an underflow of T6 (i.e. the transition from 0000_H to $FFFF_H$). In the above mentioned example, T6 would count down from 64_H, so the underflow would occur after 101 timing ticks of T6. The actual output frequency then is 79.2 kHz, instead of the expected 80 kHz.

This deviation can be compensated for by using T6 overflows. In this case, T5 counts down and T6 counts up. Upon a signal transition on pin CAPIN, the count value in T5 is captured into CAPREL and T5 is cleared to 0000_H. In its next clock cycle, T5 underflows to $FFFF_H$, and continues to count down with the following clocks. T6 is reloaded from CAPREL upon an overflow, and continues to count up with its following clock cycles (8 times faster in the above example). In this case, T5 and T6 count the same number of steps with their respective internal count frequency.

In the above example, T5 running at 1 MHz will count down to the value $FF9C_H/-100_D$ for a 10 kHz input signal applied at CAPIN, while T6 counts up from $FF9C_H$ through $FFFF_H$ to 0000_H. So the overflow occurs after 100 timing ticks of T6, and the actual output frequency at T6OUT then is the expected 80 kHz.

However, in this case CAPREL does not directly contain the time between two CAPIN events, but rather its 2's complement. Software will have to convert this value, if it is required for the operation.

Combined Capture Modes

For incremental interface applications in particular, several timer features can be combined to obtain dynamic information such as speed, acceleration, or deceleration. The current position itself can be obtained directly from the timer register (T2, T3, T4).

The time information to determine the dynamic parameters is generated by capturing the contents of the free-running timer T5 into register CAPREL. Two trigger sources for this event can be selected:

- Capture trigger on sensor signal transitions
- Capture trigger on position read operations

Capturing on sensor signal transitions is available for timer T3 inputs. This mode is selected by setting bit CT3 and selecting the intended signal(s) via bitfield CI in register T5CON. CAPREL then indicates the time between two selected transitions (measured in T5 counts).

Capturing on position read operations is available for timers T2, T3, and T4. This mode is selected by clearing bit CT3 and selecting the rising edge via bitfield CI in register T5CON. Bitfield ISCAPIN in register PISEL then selects either a read access from T3 or a read access from any of T2 or T3 or T4. CAPREL then indicates the time between two read accesses.

These operating modes directly support the measurement of position and rotational speed. Acceleration and deceleration can then be determined by evaluating subsequent speed measurements.

15.4.6 GPT2 Clock Signal Control

All actions within the timer block GPT2 are triggered by transitions of its basic clock. This basic clock is derived from the module clock f_{GPT} by a basic block prescaler, controlled by bitfield BPS2 in register T6CON (see **Figure 81**). The count clock can be generated in two different ways:

- **Internal count clock**, derived from GPT2's basic clock via a programmable prescaler, is used for (gated) Timer Mode.
- **External count clock**, derived from the timer's input pin(s), is used for Counter Mode.

For both ways, the basic clock determines the maximum count frequency and the timer's resolution:

Table 101 Basic Clock Selection for Block GPT2

Block Prescaler[1)]	**BPS2 = 01_B**	**BPS2 = 00_B**[2)]	**BPS2 = 11_B**	**BPS2 = 10_B**
Prescaling Factor for GPT2: F(BPS2)	F(BPS2) = 2	F(BPS2) = 4	F(BPS2) = 8	F(BPS2) = 16
Maximum External Count Frequency	$f_{GPT}/4$	$f_{GPT}/8$	$f_{GPT}/16$	$f_{GPT}/32$
Input Signal Stable Time	$2 \times t_{GPT}$	$4 \times t_{GPT}$	$8 \times t_{GPT}$	$16 \times t_{GPT}$

1) Please note the non-linear encoding of bitfield BPS2.
2) Default after reset.

Note: When initializing the GPT2 block, and the block prescaler BPS2 in T6CON needs to be set to a value different from its reset value (00_B), it must be initialized first before any mode involving external trigger signals is configured. These modes include counter, capture, and reload mode. Otherwise, unintended count/capture/reload events may occur.
In this case (e.g. when changing BPS2 during operation of the GPT2 block), disable related interrupts before modification of BPS2, and afterwards clear the corresponding service request flags and re-initialize those registers (T5, T6, CAPREL) that might be affected by a count/capture/reload event.

Internal Count Clock Generation

In Timer Mode and Gated Timer Mode, the count clock for each GPT2 timer is derived from the GPT2 basic clock by a programmable prescaler, controlled by bitfield TxI in the respective timer's control register TxCON.

The count frequency f_{Tx} for a timer Tx and its resolution r_{Tx} are scaled linearly with lower clock frequencies, as can be seen from the following formula:

$$f_{Tx} = \frac{f_{GPT}}{F(BPS2) \times 2^{<TxI>}} \qquad r_{Tx}[\mu s] = \frac{F(BPS2) \times 2^{<TxI>}}{f_{GPT}[MHz]} \tag{15.2}$$

The effective count frequency depends on the common module clock prescaler factor F(BPS2) as well as on the individual input prescaler factor $2^{<TxI>}$. **Table 105** summarizes the resulting overall divider factors for a GPT2 timer that result from these cascaded prescalers.

Table 102 lists GPT2 timer's parameters (such as count frequency, resolution, and period) resulting from the selected overall prescaler factor and the module clock f_{GPT}. Note that some numbers may be rounded.

Table 102 GPT2 Timer Parameters

System Clock = 10 MHz			Overall Divider Factor	System Clock = 40 MHz		
Frequency	Resolution	Period		Frequency	Resolution	Period
5.0 MHz	200 ns	13.11 ms	2	20.0 MHz	50 ns	3.28 ms
2.5 MHz	400 ns	26.21 ms	4	10.0 MHz	100 ns	6.55 ms
1.25 MHz	800 ns	52.43 ms	8	5.0 MHz	200 ns	13.11 ms
625.0 kHz	1.6 µs	104.9 ms	16	2.5 MHz	400 ns	26.21 ms
312.5 kHz	3.2 µs	209.7 ms	32	1.25 MHz	800 ns	52.43 ms
156.25 kHz	6.4 µs	419.4 ms	64	625.0 kHz	1.6 µs	104.9 ms
78.125 kHz	12.8 µs	838.9 ms	128	312.5 kHz	3.2 µs	209.7 ms
39.06 kHz	25.6 µs	1.678 s	256	156.25 kHz	6.4 µs	419.4 ms
19.53 kHz	51.2 µs	3.355 s	512	78.125 kHz	12.8 µs	838.9 ms
9.77 kHz	102.4 µs	6.711 s	1024	39.06 kHz	25.6 µs	1.678 s
4.88 kHz	204.8 µs	13.42 s	2048	19.53 kHz	51.2 µs	3.355 s

External Count Clock Input

The external input signals of the GPT2 block are sampled with the GPT2 basic clock (see **Figure 81**). To ensure that a signal is recognized correctly, its current level (high or low) must be held active for at least one complete sampling period, before changing. A signal transition is recognized if two subsequent samples of the input signal represent different levels. Therefore, a minimum of two basic clock periods are required for the sampling of an external input signal. Thus, the maximum frequency of an input signal must not be higher than half the basic clock.

Table 103 summarizes the resulting requirements for external GPT2 input signals.

Table 103 GPT2 External Input Signal Limits

GPT2 Basic Clock = 10 MHz		Input Frequ. Factor	GPT2 Divider BPS2	Input Phase Duration	GPT2 Basic Clock = 40 MHz	
Max. Input Frequency	Min. Level Hold Time				Max. Input Frequency	Min. Level Hold Time
2.5 MHz	200 ns	$f_{GPT}/4$	01_B	$2 \times t_{GPT}$	10.0 MHz	50 ns
1.25 MHz	400 ns	$f_{GPT}/8$	00_B	$4 \times t_{GPT}$	5.0 MHz	100 ns
625.0 kHz	800 ns	$f_{GPT}/16$	11_B	$8 \times t_{GPT}$	2.5 MHz	200 ns
312.5 kHz	1.6 µs	$f_{GPT}/32$	10_B	$16 \times t_{GPT}$	1.25 MHz	400 ns

These limitations are valid for all external input signals to GPT2, including the external count signals in Counter Mode and the gate input signals in Gated Timer Mode.

15.4.7 Interrupt Control for GPT2 Timers and CAPREL

When a timer overflows from $FFFF_H$ to 0000_H (when counting up), or when it underflows from 0000_H to $FFFF_H$ (when counting down), its interrupt request flag in register GPT2_T5 or GPT2_T6I will be set. This will cause an interrupt to the respective timer interrupt vector, if the respective interrupt enable bit is set.

Whenever a transition according to the selection in bit field CI is detected at pin CAPIN, interrupt request flag in register GPT12_CR is set. Setting any request flag will cause an interrupt to the respective timer or CAPREL interrupt vector, if the respective interrupt enable bit is set.

There is an interrupt control register for each of the two timers (T5, T6) and for the CAPREL register. All interrupt control registers have the same structure described in section Interrupt Control.

15.4.8 GPT2 Registers

15.4.8.1 GPT2 Timer Registers

T5
Timer 5 Count Register **($2C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T5															
rwh															

Field	Bits	Typ	Description
T5	[15:0]	rwh	**Timer T5 Current Value** Contains the current value of the timer T5

T6
Timer T6 Count Register **(30_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T6															
rwh															

Field	Bits	Typ	Description
T6	[15:0]	rwh	**Timer T6 Current Value** Contains the current value of the timer T6

CAPREL
Capture/Reload Register **($1C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPREL															
rwh															

Field	Bits	Typ	Description
CAPREL	[15:0]	rwh	**Current reload value or Captured value** Contains the current value of the CAPREL register

15.4.8.2 GPT2 Timer Control Registers

15.4.8.2.1 GPT2 Core Timer T6 Control Register

T6CON
Timer T6 Control Register **(18_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T6SR	T6CLR	RES	BPS2		T6OTL	T6OE	T6UDE	T6UD	T6R	T6M			T6I		
rw	rw	r	rw		rwh	rw	rw	rw	rw	rw			rw		

Field	Bits	Type	Description
T6I	[2:0]	rw	**Timer T6 Input Parameter Selection** Depends on the operating mode, see respective sections for encoding: **Table 105** for Timer Mode and Gated Timer Mode **Table 106** for Counter Mode
T6M	[5:3]	rw	**Timer T6 Mode Control** (Basic Operating Mode) 000_B Timer Mode 001_B Counter Mode 010_B Gated Timer Mode with gate active low 011_B Gated Timer Mode with gate active high 100_B Reserved. Do not use this combination. 101_B Reserved. Do not use this combination. 110_B Reserved. Do not use this combination. 111_B Reserved. Do not use this combination.
T6R	6	rw	**Timer T6 Run Bit** 0_B Timer T6 stops 1_B Timer T6 runs
T6UD	7	rw	**Timer T6 Up/Down Control**[1)] 0_B Timer T6 counts up 1_B Timer T6 counts down *Note: This bit only controls count direction of T6 if bit T6UDE = 0.*
T6UDE	8	rw	**Timer T6 External Up/Down Enable**[1)] 0_B Count direction is controlled by bit T6UD; input T6EUD is disconnected 1_B Count direction is controlled by input T6EUD
T6OE	9	rw	**Overflow/Underflow Output Enable** 0_B Alternate Output Function Disabled 1_B State of timer T6 toggle latch is output on pin T6OUT

Field	Bits	Type	Description
T6OTL	10	rwh	**Timer T6 Overflow Toggle Latch** Toggles on each overflow/underflow of timer T6. Can be set or reset by software (see separate description)
BPS2	[12:11]	rw	**GPT2 Block Prescaler Control** Selects the basic clock for block GPT2 (see also **Section 15.4.6**) 00_B $f_{GPT}/4$ 01_B $f_{GPT}/2$ 10_B $f_{GPT}/16$ 11_B $f_{GPT}/8$
RES	13	r	**Reserved** Read as 0; should be written with 0.
T6CLR	14	rw	**Timer T6 Clear Enable Bit** 0_B Timer T6 is not cleared on a capture event 1_B Timer T6 is cleared on a capture event
T6SR	15	rw	**Timer T6 Reload Mode Enable** 0_B Reload from register CAPREL disabled 1_B Reload from register CAPREL enabled

1) See **Table 104** for encoding of bits T6UD and T6UDE.

15.4.8.2.2 GPT2 Auxiliary Timer T5 Control Registers

T5CON
Timer 5 Control Register **(14_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T5SC	T5CLR	CI		RES	CT3	T5RC	T5UDE	T5UD	T5R	RES	T5M		T5I		
rw	rw	rw		r	rw	rw	rw	rw	rw	r	rw		rw		

Field	Bits	Type	Description
T5I	[2:0]	rw	**Timer T5 Input Parameter Selection** Depends on the operating mode, see respective sections for encoding: **Table 105** for Timer Mode and Gated Timer Mode **Table 107** for Counter Mode
T5M	[4:3]	rw	**Timer T5 Mode Control** (Basic Operating Mode) 00_B Timer Mode 01_B Counter Mode 10_B Gated Timer Mode with gate active low 11_B Gated Timer Mode with gate active high
RES	5	r	**Reserved** Read as 0; should be written with 0.
T5R	6	rw	**Timer T5 Run Bit** 0_B Timer T5 stops 1_B Timer T5 runs *Note: This bit only controls timer T5 if bit T5RC = 0.*
T5UD	7	rw	**Timer T5 Up/Down Control**[1)] 0_B Timer T5 counts up 1_B Timer T5 counts down *Note: This bit only controls count direction of T5 if bit T5DE = 0.*
T5UDE	8	rw	**Timer T5 External Up/Down Enable**[1)] 0_B Count direction is controlled by bit T5UD; input T5EUD is disconnected 1_B Count direction is controlled by input T5EUD
T5RC	9	rw	**Timer T5 Remote Control** 0_B Timer T5 is controlled by its own run bit T5R 1_B Timer T5 is controlled by the run bit T6R of core timer T6, not by bit T5R
CT3	10	rw	**Timer T3 Capture Trigger Enable** 0_B Capture trigger from input line CAPIN 1_B Capture trigger from T3 input lines T3IN and/or T3EUD

Field	Bits	Type	Description
RES	11	r	**Reserved** Read as 0; should be written with 0.
CI	[13:12]	rw	**Register CAPREL Capture Trigger Selection**[2)] 00_B Capture disabled 01_B Positive transition (rising edge) on CAPIN[3)] or any transition on T3IN 10_B Negative transition (falling edge) on CAPIN or any transition on T3EUD 11_B Any transition (rising or falling edge) on CAPIN or any transition on T3IN or T3EUD
T5CLR	14	rw	**Timer T5 Clear Enable Bit** 0_B Timer T5 is not cleared on a capture event 1_B Timer T5 is cleared on a capture event
T5SC	15	rw	**Timer T5 Capture Mode Enable** 0_B Capture into register CAPREL disabled 1_B Capture into register CAPREL enabled

1) See **Table 104** for encoding of bits T5UD and T5UDE.
2) To define the respective trigger source signal, also bit CT3 must be regarded (see **Table 100**).
3) Rising edge must be selected if capturing is triggered by the internal GPT1 read signals (see register PISEL and **"Combined Capture Modes" on Page 471**).

15.4.8.2.3 Encoding of Timer Count Direction Control

Table 104 GPT2 Timer Count Direction Control

Pin TxEUD	Bit TxUDE	Bit TxUD	Count Direction
X	0	0	Count Up
X	0	1	Count Down
0	1	0	Count Up
1	1	0	Count Down
0	1	1	Count Down
1	1	1	Count Up

15.4.8.2.4 Timer Mode and Gated Timer Mode: Encoding of Overall Prescaler Factor

Table 105 GPT2 Overall Prescaler Factors for Internal Count Clock (Timer Mode and Gated Timer Mode)

Individual Prescaler for Tx	Common Prescaler for Module Clock[1)]			
	BPS2 = 01_B	BPS2 = 00_B	BPS2 = 11_B	BPS2 = 10_B
TxI = 000_B	2	4	8	16
TxI = 001_B	4	8	16	32
TxI = 010_B	8	16	32	64
TxI = 011_B	16	32	64	128
TxI = 100_B	32	64	128	256
TxI = 101_B	64	128	256	512
TxI = 110_B	128	256	512	1024
TxI = 111_B	256	512	1024	2048

1) Please note the non-linear encoding of bitfield BPS2.

15.4.8.2.5 Counter Mode: Encoding of Input Edge Selection

Table 106 GPT2 Core Timer T6 Input Edge Selection (Counter Mode)

T6I	Triggering Edge for Counter Increment/Decrement
000_B	None. Counter T6 is disabled
001_B	Positive transition (rising edge) on T6IN
010_B	Negative transition (falling edge) on T6IN
011_B	Any transition (rising or falling edge) on T6IN
$1XX_B$	Reserved. Do not use this combination

Table 107 GPT2 Auxiliary Timer T5 Input Edge Selection (Counter Mode)

T5I	Triggering Edge for Counter Increment/Decrement
$X00_B$	None. Counter T5 is disabled
001_B	Positive transition (rising edge) on T5IN
010_B	Negative transition (falling edge) on T5IN
011_B	Any transition (rising or falling edge) on T5IN
101_B	Positive transition (rising edge) of T6 toggle latch T6OTL
110_B	Negative transition (falling edge) of T6 toggle latch T6OTL
111_B	Any transition (rising or falling edge) of T6 toggle latch T6OTL

15.4.8.3 GPT2 Timer and CAPREL Interrupt Control Registers

The Interrupt control register for GPT2 and CAPREL are located in the SCU.**GPT12IEN** and SCU. **GPT12IRC**.

15.5 Miscellaneous GPT12 Registers

The following registers are not assigned to a specific timer block. They control general functions and/or give general information.

Register PISEL selects timer input signal from several sources under software control.

PISEL
Port Input Select Register **(04_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISCAPIN		IST6EUD	IST6IN	IST5EUD	IST5IN	IST4EUD		IST4IN		IST3EUD		IST3IN		IST2EUD	IST2IN
rw		rw	rw	rw	rw	rw		rw		rw		rw		rw	rw

Field	Bits	Typ	Description
IST2IN	0	rw	**Input Select for T2IN** 0_B Signal T2INA is selected 1_B Signal T2INB is selected
IST2EUD	1	rw	**Input Select for T2EUD** 0_B Signal T2EUDA is selected 1_B Signal T2EUDB is selected
IST3IN	[3:2]	rw	**Input Select for T3IN** 00_B Signal T3INA is selected 01_B Signal T3INB is selected 10_B Signal T3INC is selected 11_B Signal T3IND is selected
IST3EUD	[5:4]	rw	**Input Select for T3EUD** 00_B Signal T3EUDA is selected 01_B Signal T3EUDB is selected 10_B Signal T3EUDC is selected 11_B Signal T3EUDD is selected
IST4IN	[7:6]	rw	**Input Select for T4IN** 00_B Signal T4INA is selected 01_B Signal T4INB is selected 10_B Signal T4INC is selected 11_B Signal T4IND is selected
IST4EUD	[9:8]	rw	**Input Select for T4EUD** 00_B Signal T4EUDA is selected 01_B Signal T4EUDB is selected 10_B Signal T4EUDC is selected 11_B Signal T4EUDD is selected
IST5IN	10	rw	**Input Select for T5IN** 0_B Signal T5INA is selected 1_B Reserved

Field	Bits	Typ	Description
IST5EUD	11	rw	**Input Select for T5EUD** 0_B Signal T5EUDA is selected 1_B Signal T5EUDB is selected
IST6IN	12	rw	**Input Select for T6IN** 0_B Signal T6INA is selected 1_B Signal T6INB is selected
IST6EUD	13	rw	**Input Select for T6EUD** 0_B Signal T6EUDA is selected 1_B Signal T6EUDB is selected
ISCAPIN	[15:14]	rw	**Input Select for CAPIN** 00_B Signal CAPINA is selected 01_B Signal CAPINB is selected P0.3 if **MODPISEL1**.GPT12CAPINB = 0 BEMF status if **MODPISEL1**.GPT12CAPINB = 1 10_B Signal CAPINC (Read trigger from T3) is selected 11_B Signal CAPIND (Read trigger from T2 or T3 or T4) is selected

Register ID indicates the module version.

ID

Module Identification Register **(00_H)** **Reset Value: 5804_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOD_TYPE								**MOD_REV**							
r								r							

Field	Bits	Type	Description
MOD_REV	[7:0]	r	**Module Revision Number** MOD_REV defines the revision number. The value of a module revision starts with 01_H (first revision).
MOD_TYPE	[15:8]	r	**Module Identification Number** This bitfield defines the module identification number (58_H = GPT12E).

15.6 Register Map

Table 108 Register Address Space

Module	Base Address	End Address	Note
GPT12E	40010000_H	$40013FFF_H$	

Table 109 shows all registers which are required for programming of the GPT12E module. It summarizes the GPT12E kernel registers and the module external registers and defines their addresses and reset values.

Table 109 GPT12E Module Register Summary

Name	Description	GPT12 Offset Address[1)]	Reset Value
General Purpose Timer Unit (GPT12E), GPT1 Registers, GPT2 Registers, Miscellaneous GPT12 Registers			
ID	GPT12E Module Identification Register	00_H	5804_H
PISEL	GPT12E Port Input Select Register	04_H	0000_H
T2CON	GPT12E Timer T2 Control Register	08_H	0000_H
T3CON	GPT12E Timer T3 Control Register	$0C_H$	0000_H
T4CON	GPT12E Timer T4 Control Register	10_H	0000_H
T5CON	GPT12E Timer T5 Control Register	14_H	0000_H
T6CON	GPT12E Timer T6 Control Register	18_H	0000_H
CAPREL	GPT12E Capture/Reload Register	$1C_H$	0000_H
T2	GPT12E Timer T2 Count Register	20_H	0000_H
T3	GPT12E Timer T3 Count Register	24_H	0000_H
T4	GPT12E Timer T4 Count Register	28_H	0000_H
T5	GPT12E Timer T5 Count Register	$2C_H$	0000_H
T6	GPT12E Timer T6 Count Register	30_H	0000_H

1) The absolute register address is calculated as follows:
Module Base Address + Offset Address (shown in this column).

15.7 Implementation of the GPT12 Module

This chapter describes the implementation of the GPT12 module in the TLE987x device.

15.7.1 Module Connections

Besides the described intra-module connections, the timer unit blocks GPT1 and GPT2 are connected to their environment in two basic ways:

- **Internal connections** interface the timers with on-chip resources such as clock generation unit, interrupt controller, or other timers.
  The GPT module is clocked with the TLE987x system clock, so $f_{GPT} = f_{SYS}$.
- **External connections** interface the timers with external resources via port pins.

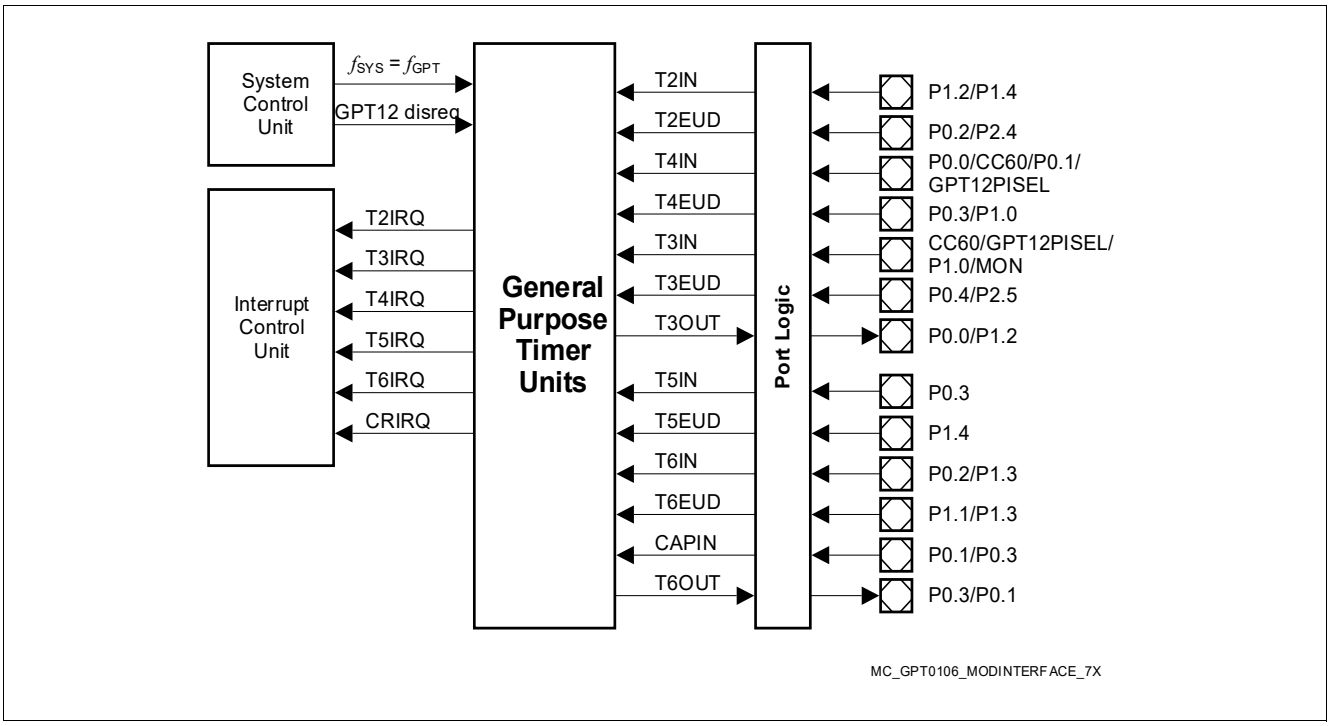


Figure 111 GPT Module Interfaces

The following table **Table 110** (GPT12) shows the digital connections of the GPT12 module with other modules or pins in the TLE987x device.

Table 110 GPT12 Digital Connections in TLE987x

Signal	from/to Module	I/O to GPT	Can be used to/as
T2INA	P1.2	I	count input signals for timer T2
T2INB	P1.4	I	
T2EUDA	P0.2	I	direction input signals for timer T2
T2EUDB	P2.4	I	
T2IRQ	ICU/SCU	O	interrupt request from timer T2

Table 110 GPT12 Digital Connections in TLE987x (cont'd)

Signal	from/to Module	I/O to GPT	Can be used to/as
T3INA	CC60	I	count input signals for timer T3
T3INB	GPT12PISEL	I	
T3INC	P1.0	I	
T3IND	MON	I	
T3EUDA	P0.4	I	direction input signals for timer T3
T3EUDB	P2.5	I	
T3OUT	P0.0	O	count output signal for timer T3
	P1.2	O	
T3IRQ	ICU/SCU	O	interrupt request from timer T3
T4INA	P0.0	I	count input signals for timer T4
T4INB	CC60	I	
T4INC	P0.1	I	
T4IND	GPT12PISEL	I	
T4EUDA	P0.3	I	direction input signals for timer T4
T4EUDB	P1.0	I	
T4IRQ	ICU/SCU	O	interrupt request from timer T4
T5INA	P0.3	I	count input signals for timer T5
T5EUDA	P1.4	I	direction input signals for timer T5
T5IRQ	ICU/SCU	O	interrupt request from timer T5
T6INA	P0.2	I	count input signals for timer T6
T6INB	P1.3	I	
T6EUDA	P1.1	I	direction input signals for timer T6
T6EUDB	P1.3	I	
T6OUT	P0.3	O	count output signal for timer T6
	P0.1	O	
T6IRQ	ICU/SCU	O	interrupt request from timer T6
T6OFL		O	over/under-flow signal from timer T6
CAPINA	P0.1	I	input capture signals
CAPINB	P0.3	I	
CAPINC	read trigger from T3	I	
CAPIND	read trigger from T2 or T3 or T4	I	
CRIRQ	ICU/SCU	O	interrupt request from capture control

Port Control

Port pins to be used for timer input signals must be switched to input (bitfield PC in the respective port control register must be $0xxx_B$) and must be selected via register PISEL.

Port pins to be used for timer output signals must be switched to output and the alternate timer output signal must be selected (bitfield PC in the respective port control register must be $1xxx_B$).

Note: For a description of the port control registers, please refer to chapter "Parallel Ports".

Interrupts

The GPT12 has six interrupt request lines.

Interrupt nodes to be used for timer interrupt requests must be enabled and programmed to a specific interrupt level.

Debug Details

While the module GPT is disabled, its registers can still be read. While disabled the following registers can be written: PISEL, T5CON.

16 Timer2 and Timer21

This chapter describes the Timer2 and Timer21. Each timer is a 16-bit timer which additionally can function as a counter. Each Timer2 module also provides a single channel 16-bit capture.

16.1 Features

- 16-bit auto-reload mode
  - Selectable up- or down-counting
- One-channel 16-bit capture mode

16.2 Introduction

The timer modules are general-purpose 16-bit timers. Timer2 and Timer21 can function as timers or counters in each of their modes. As timers, they count with an input clock of $f_{PCLK}/12$ (if the prescaler is disabled). As a counter, Timer2 counts 1-to-0 transitions on pin T2. In the counter mode, the maximum resolution for counting is $f_{PCLK}/24$ (if the prescaler is disabled).

Note: "Timer2" is generally referred in the following description which is applicable to each of the Timer2 and Timer21.

16.2.1 Timer2 and Timer21 mode overview

Table 111 Timer2 and Timer21 modes

Mode	Description
Auto-reload	**Up/down-count-disabled** • Counting up only. • Counting starts from the 16-bit reload value, overflow at $FFFF_H$. • The reload event can be configured to be triggered only by the overflow condition or by a negative or positive edge at the input pin T2EX as well. • Programmable reload value in register RC2. • Interrupt is generated with reload events.
Auto-reload	**Up/down-count-enabled** • Counting up or down, direction determined by level at input pin T2EX. • No interrupt is generated. • Counting up – Counting starts from the 16-bit reload value, overflow at $FFFF_H$. – Reload event triggered by overflow condition. – Programmable reload value in register RC2. • Counting down – Counting starts from $FFFF_H$, underflow at value defined in register RC2. – Reload event triggered by underflow condition. – Reload value fixed at $FFFF_H$.
Channel capture	• Counting up only. • Counting starts from 0000_H, overflow at $FFFF_H$. • Reload event triggered by overflow condition. • Reload value fixed at 0000_H. • Capture event triggered by falling/rising edge at pin T2EX. • Captured timer value stored in register RC2. • Reload or capture events generate interrupts.

Timer2 can be started by using TR2 bit by hardware or software. Timer2 can be started by setting TR2 bit by software. If bit T2RHEN is set, Timer2 can be started by hardware. Bit T2REGS defines the event on pin T2EX: falling edge or rising edge, that can set the run bit TR2 by hardware. Timer2 can only be stopped by resetting TR2 bit by software.

16.3 Functional Description

16.3.1 Auto-Reload Mode

The auto-reload mode is selected when the bit CP_RL2 in register **T2CON** is zero. In the auto-reload mode, Timer2 counts to an overflow value and then reloads its register contents with a 16-bit start value for a fresh counting sequence. The overflow condition is indicated by setting bit TF2 in the **T2CON** register. This will then generate an interrupt request to the core. The overflow flag TF2 must be cleared by software.

The auto-reload mode is further classified into two categories depending upon the DCEN control bit.

16.3.1.1 Up/Down Count Disabled

If DCEN = 0, the up-down count selection is disabled. The timer, therefore, functions as a pure up counter/timer only. The operational block diagram is shown in **Figure 112**.

In this mode, if EXEN2 = 0, the timer starts to count up to a maximum of $FFFF_H$, once TR2 is set. Upon overflow, bit TF2 is set and the timer register is reloaded with the 16-bit reload value of the RC2 register. This reload value is chosen by software, prior to the occurrence of an overflow condition. A fresh count sequence is started and the timer counts up from this reload value as in the previous count sequence.

If EXEN2 = 1, the timer counts up to a maximum of $FFFF_H$ once TR2 is set. A 16-bit reload of the timer registers from register RC2 is triggered either by an overflow condition or by a negative/positive edge (chosen by T2MOD.EDGESEL) at input pin T2EX. If an overflow caused the reload, the overflow flag TF2 is set. If a negative/positive transition at pin T2EX caused the reload, bit EXF2 is set. In either case, an interrupt is generated to the core and the timer proceeds to its next count sequence. The EXF2 flag, similar to the TF2, must be cleared by software.

If bit T2RHEN is set, Timer2 is started by first falling edge/rising edge at pin T2EX, which is defined by bit T2REGS. If bit EXEN2 is set, bit EXF2 is also set at the same point when Timer2 is started with the same falling edge/rising edge at pin T2EX, which is defined by bit EDGESEL. The reload will happen with the following negative/positive transitions at pin T2EX, which is defined by bit EDGESEL.

Note: In counter mode, if the reload via T2EX and the count clock T2 are detected simultaneously, the reload takes precedence over the count. The counter increments its value with the following T2 count clock.

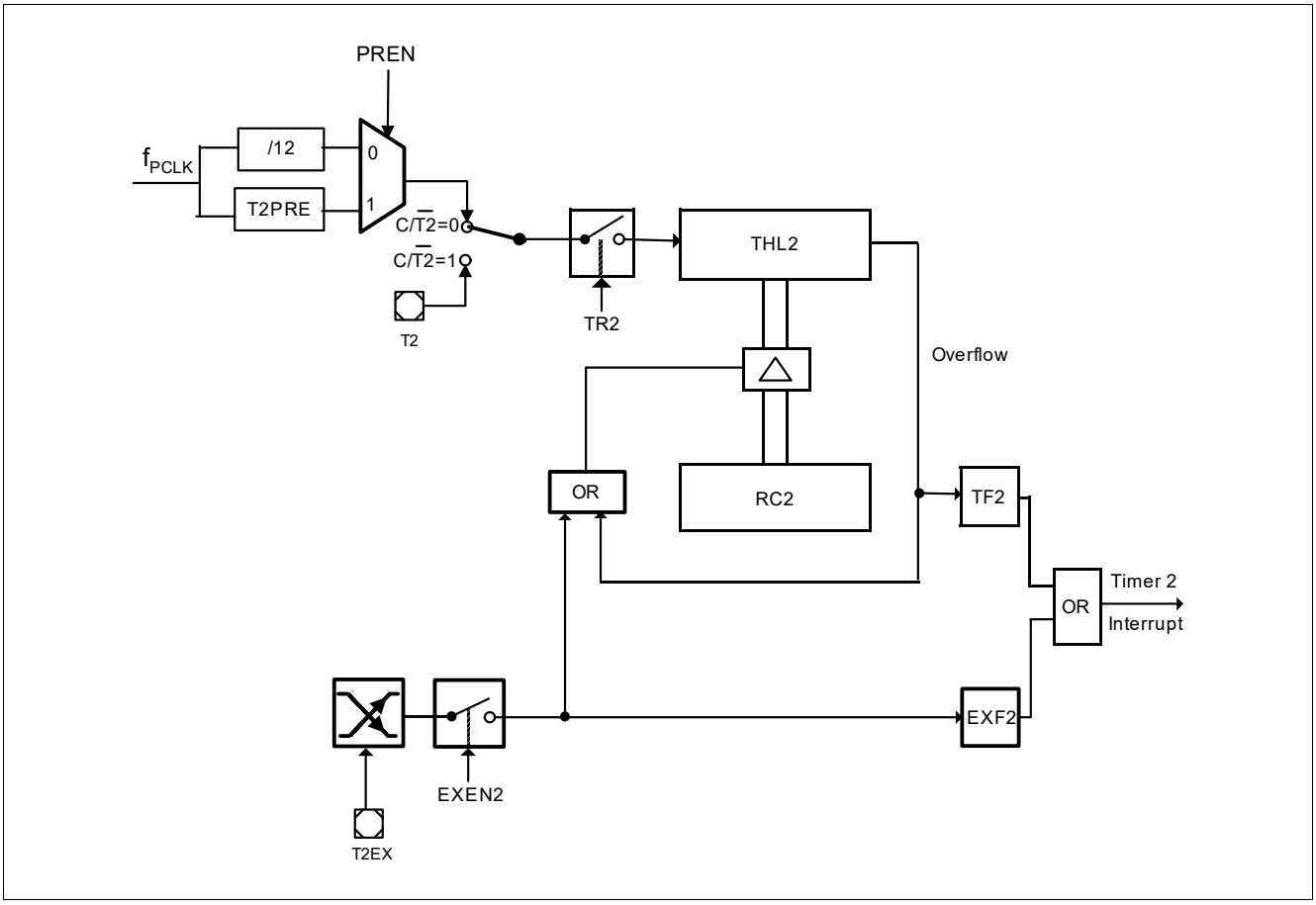


Figure 112 Auto-Reload Mode (DCEN = 0)

16.3.1.2 Up/Down Count Enabled

If DCEN = 1, the up-down count selection is enabled. The direction of count is determined by the level at input pin T2EX. The operational block diagram is shown in **Figure 113**.

A logic 1 at pin T2EX sets the Timer2 to up counting mode. The timer, therefore, counts up to a maximum of $FFFF_H$. Upon overflow, bit TF2 is set and the timer register is reloaded with a 16-bit reload value of the RC2 register. A fresh count sequence is started and the timer counts up from this reload value as in the previous count sequence. This reload value is chosen by software, prior to the occurrence of an overflow condition.

A logic 0 at pin T2EX sets the Timer2 to down counting mode. The timer counts down and underflows when the THL2 value reaches the value stored at register RC2. The underflow condition sets the TF2 flag and causes $FFFF_H$ to be reloaded into the THL2 register. A fresh down counting sequence is started and the timer counts down as in the previous counting sequence.

If bit T2RHEN is set, Timer2 can only be started either by rising edge (T2REGS = 1) at pin T2EX and then do the up counting, or be started by falling edge (T2REGS = 0) at pin T2EX and then do the down counting.

In this mode, bit EXF2 toggles whenever an overflow or an underflow condition is detected. This flag, however, does not generate an interrupt request.

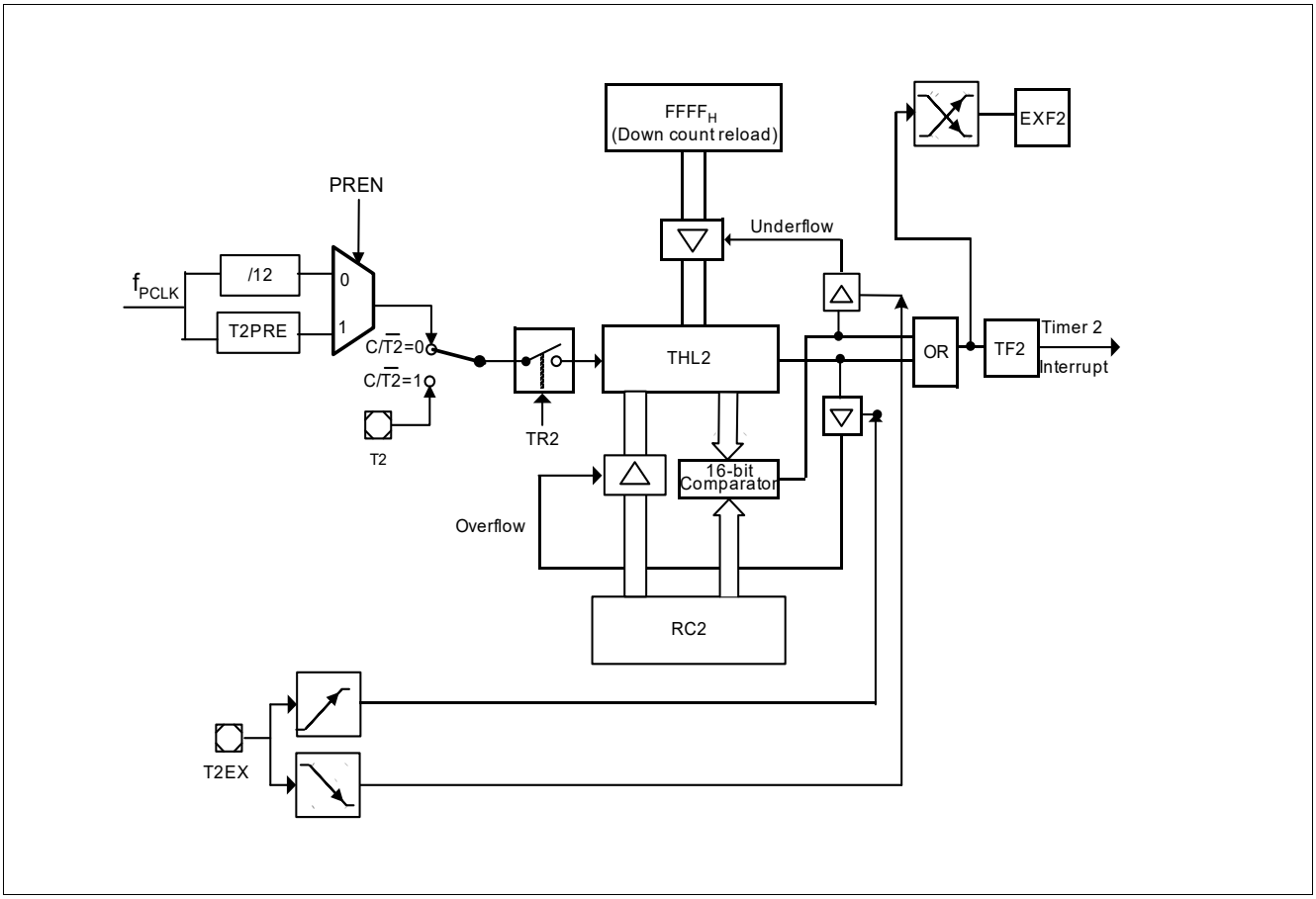


Figure 113 Auto-Reload Mode (DCEN = 1)

16.3.2 Capture Mode

In order to enter the 16-bit capture mode, bits CP_RL2 and EXEN2 in register **T2CON** must be set. In this mode, the down count function must remain disabled. The timer functions as a 16-bit timer or counter and always counts up to $FFFF_H$ and overflows. Upon an overflow condition, bit TF2 is set and the timer reloads its registers with 0000_H. The setting of TF2 generates an interrupt request to the core.

Additionally, with a falling/rising edge on pin T2EX (chosen by T2MOD.EDGESEL) the contents of the timer register (THL2) are captured into the RC2 register. The external input is sampled in every PCLK cycle. When a sampled input shows a low (high) level in one PCLK cycle and a high (low) in the next PCLK cycle, a transition is recognized. If the capture signal is detected while the counter is being incremented, the counter is first incremented before the capture operation is performed. This ensures that the latest value of the timer register is always captured.

If bit T2RHEN is set, Timer2 is started by first falling edge/rising edge at pin T2EX, which is defined by bit T2REGS. If bit EXEN2 is set, bit EXF2 is also set at the same point when Timer2 is started with the same falling edge/rising edge at pin T2EX, which is defined by bit EDGESEL. The capture will happen with the following negative/positive transitions at pin T2EX, which is defined by bit EDGESEL.

When the capture operation is completed, bit EXF2 is set and can be used to generate an interrupt request. **Figure 114** describes the capture function of Timer2.

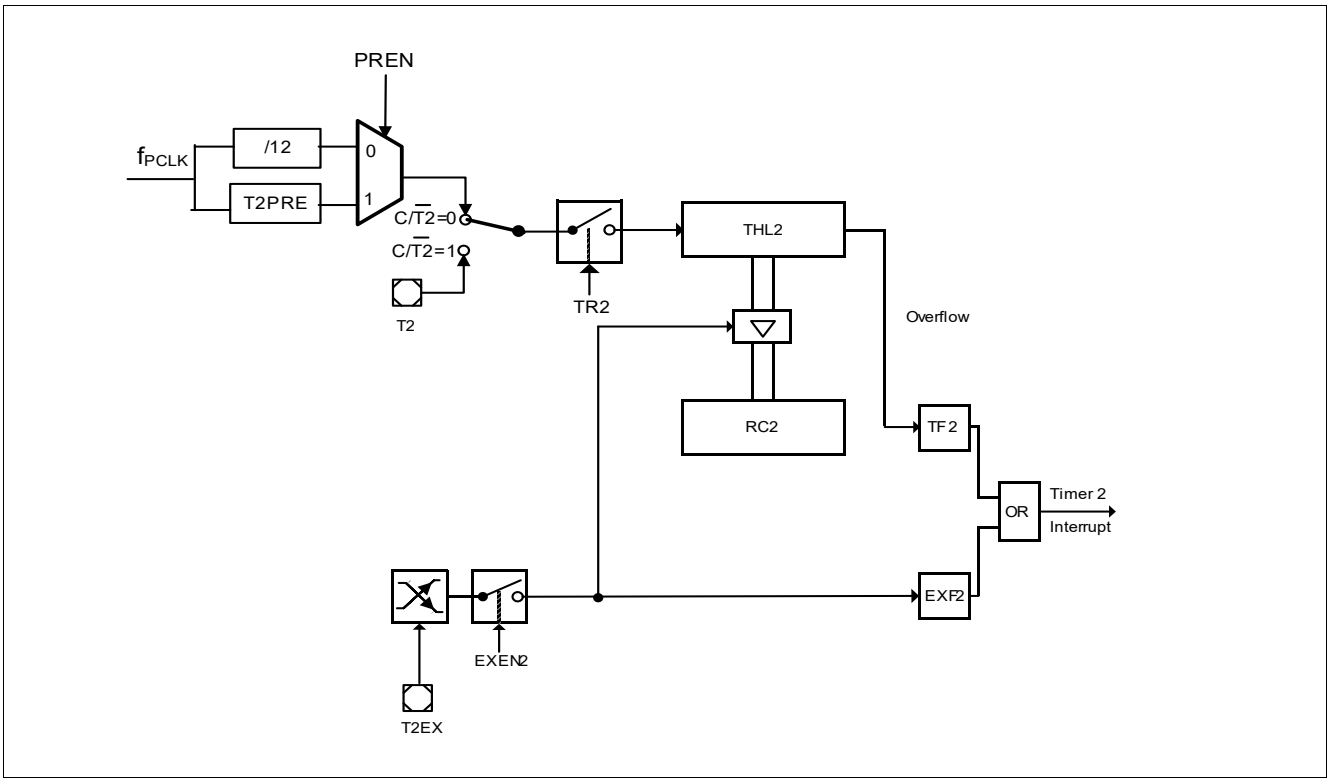


Figure 114 Capture Mode

16.3.3 Count Clock

The count clock for the auto-reload mode is chosen by the bit C_T2 in register **T2CON**. If C_T2 = 0, a count clock of PCLK/12 (if prescaler is disabled) is used for the count operation.

If C_T2 = 1, Timer2 behaves as a counter that counts 1-to-0 transitions of input pin T2. The counter samples pin T2 over 2 PCLK cycles. If a 1 was detected during the first clock and a 0 was detected in the following clock, then the counter increments by one. Therefore, the input levels should be stable for at least 1 clock.

If bit T2RHEN is set, Timer2 can be started by the falling edge/rising edge on pin T2EX, which is defined by bit T2REGS.

Note: If pin T2 is not connected, counting clock function on pin T2 cannot be used.

16.4 Module Interfaces

This section describes:

- the TLE987x module related interfaces such as port connections and interrupt control
- all TLE987x module related registers with their addresses

16.4.1 Interfaces of the Timer2 and Timer21

Overviews of the Timer2 and Timer21 kernel I/O interfaces and interrupt signals are shown in **Figure 115** and **Figure 116**.

Timer2 and Timer21 can be suspended when Debug Mode enters Monitor Mode and has the Debug Suspend signal activated, provided the timer suspend bits, T2SUSP and T21SUSP (in SCU SFR MODSUSP) are set. Refer to SCU chapter.

The interrupt request of the Timer2 and Timer21 is not connected directly to the CPU's Interrupt Controller, but via the System Control Unit (SCU). The General Purpose IO (GPIO) Port provides the interface from the Timer2 and Timer21 to the external world.

The external trigger and counter inputs of the two Timer2 modules can be selected from several different sources. This selection is performed by the SCU via the corresponding input control and select bits in SFR MODPISEL1 and MODPISEL2.

In the TLE987x, Timer2 and Timer21 allow additionally to trigger ADC1 conversions through the t2(1)_adc_trigger signals. These trigger signals are generated while the timer is working in timer mode (C_T2 = 0).

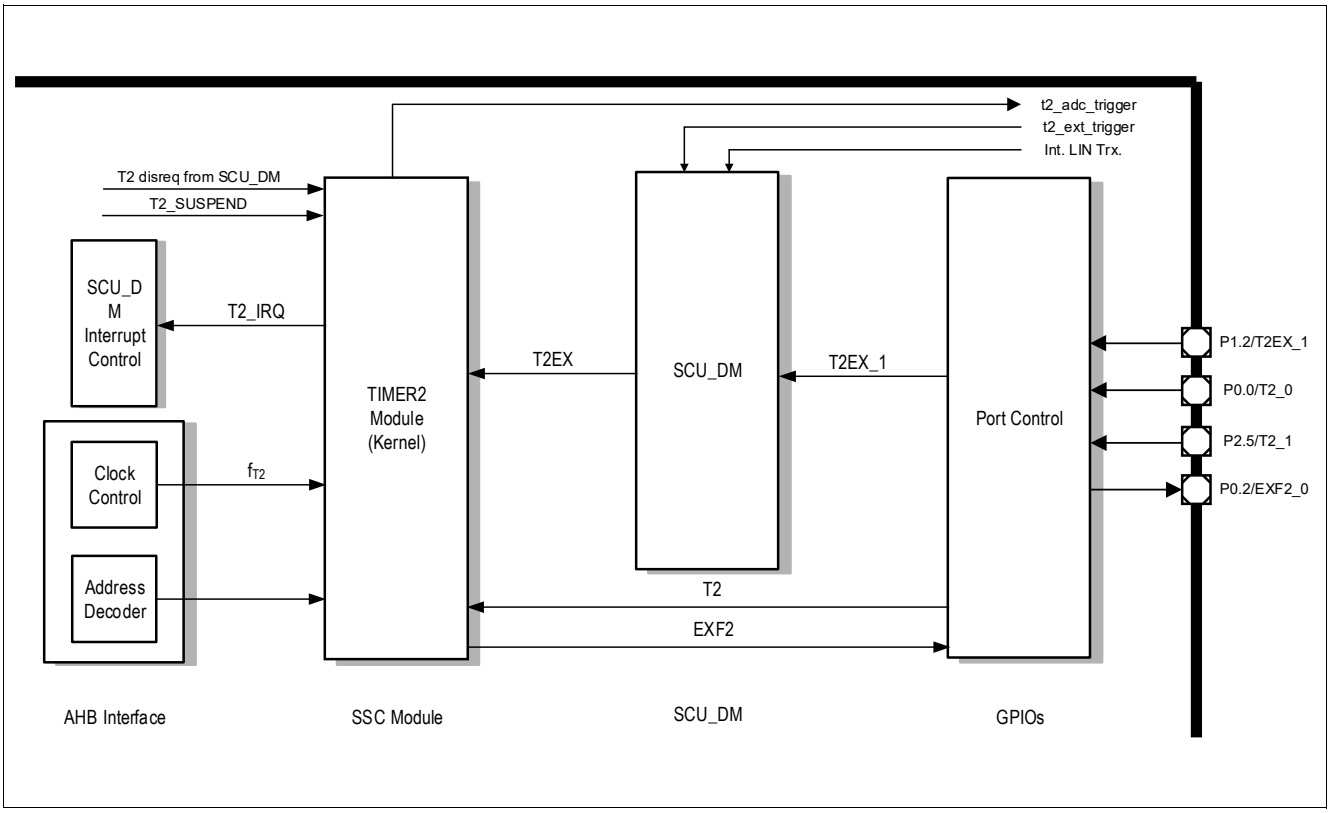


Figure 115 Timer2 Module I/O Interface

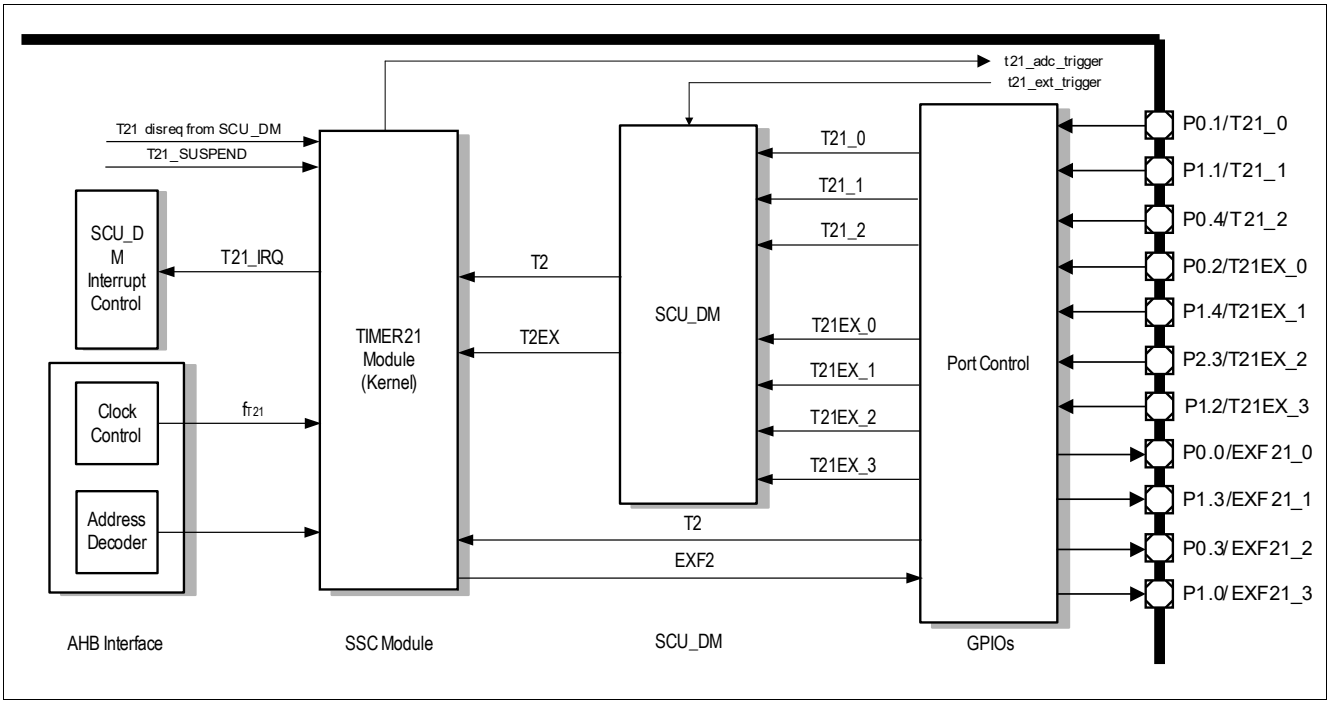


Figure 116 Timer21 Module I/O Interface

16.5 Interrupt Generation

When an interrupt event happened, the corresponding interrupt flag bit EXF2/TF2 is set; an interrupt for the interrupt event EXF2/TF2 will be generated.

*Note: When the timer/counter is stopped and while the module remains enabled, it is possible for an external event at T2EX to generate an interrupt. For this to occur, bit EXEN2 in SFR **T2CON** must be set. In this case, a dummy reload or capture happens depending on the CP_RL2 bit selection. The resulting interrupt could therefore be used in the product as an external falling/rising edge triggered interrupt.*

16.6 Register Definition

16.6.1 Timer2 Registers

All Timer2 and Timer21 register names described in the following sections will be referenced in other chapters with the module name prefix “T2_” and “T21_”, respectively.

16.6.1.1 Mode Register

The T2MOD is used to configure Timer2 for various modes of operation.

T2MOD
Timer 2 Mode Register **(04_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
T2REGS	**T2RHEN**	**EDGESEL**	**PREN**	**T2PRE**			**DCEN**
rw	rw	rw	rw	rw			rw

Field	Bits	Type	Description
DCEN	0	rw	**Up/Down Counter Enable** 0_B Up/Down Counter function is disabled 1_B Up/Down Counter function is enabled and controlled by pin T2EX (Up = 1, Down = 0)
T2PRE	3:1	rw	**Timer2 Prescaler Bit** Selects the input clock for Timer2 which is derived from the peripheral clock. 000_B $f_{T2} = f_{PCLK}$ 001_B $f_{T2} = f_{PCLK} / 2$ 010_B $f_{T2} = f_{PCLK} / 4$ 011_B $f_{T2} = f_{PCLK} / 8$ 100_B $f_{T2} = f_{PCLK} / 16$ 101_B $f_{T2} = f_{PCLK} / 32$ 110_B $f_{T2} = f_{PCLK} / 64$ 111_B $f_{T2} = f_{PCLK} / 128$
PREN	4	rw	**Prescaler Enable** 0_B Prescaler is disabled and the by 12-divider takes effect. 1_B Prescaler is enabled (see T2PRE bit) and the by 12-divider is bypassed.
EDGESEL	5	rw	**Edge Select in Capture Mode/Reload Mode** 0_B The falling edge at Pin T2EX is selected. 1_B The rising edge at Pin T2EX is selected.
T2RHEN	6	rw	**Timer2 External Start Enable** 0_B Timer2 External Start is disabled. 1_B Timer2 External Start is enabled.

Field	Bits	Type	Description
T2REGS	7	rw	**Edge Select for Timer2 External Start** 0_B The falling edge at Pin T2EX is selected. 1_B The rising edge at Pin T2EX is selected.

16.6.1.2 Control Register

Control register is used to control the operating modes and interrupt of Timer2.

T2CON
Timer 2 Control Register **(00_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
TF2	**EXF2**	**RES**		**EXEN2**	**TR2**	**C_T2**	**CP_RL2**
r	r	r		rw	rwh	rw	rw

Field	Bits	Type	Description
CP_RL2	0	rw	**Capture/Reload Select** 0_B Reload upon overflow or upon negative/positive transition at pin T2EX (when EXEN2 = 1). 1_B Capture Timer2 data register contents on the negative/positive transition at pin T2EX, provided EXEN2 = 1.The negative or positive transition at Pin T2EX is selected by bit EDGESEL.
C_T2	1	rw	**Timer or Counter Select** 0_B Timer function selected. 1_B Count upon negative edge at pin T2.
TR2	2	rwh	**Timer2 Start/Stop Control** 0_B Stop Timer2. 1_B Start Timer2.
EXEN2	3	rw	**Timer2 External Enable Control** 0_B External events are disabled. 1_B External events are enabled in Capture/Reload Mode.
RES	5:4	r	**Reserved** Returns 0 if read; should be written with 0.
EXF2	6	r	**Timer2 External Flag** In Capture/Reload Mode, this bit is set by hardware when a negative/positive transition occurs at pin T2EX, if bit EXEN2 = 1. This bit must be cleared by software. *Note: When bit DCEN = 1 in auto-reload mode, no interrupt request to the core is generated.*
TF2	7	r	**Timer2 Overflow/Underflow Flag** Set by a Timer2 overflow/underflow. Must be cleared by software.

T2ICLR
Timer 2 Interrupt Clear Register **($1C_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
TF2CLR	**EXF2CLR**	**RES**					
w	w	r					

Field	Bits	Type	Description
RES	5:0	r	**Reserved** Returns 0 if read; should be written with 0.
EXF2CLR	6	w	**External Interrupt Clear Flag** 0_B External interrupt is not cleared. 1_B External interrupt is cleared.
TF2CLR	7	w	**Overflow/Underflow Interrupt Clear Flag** 0_B Overflow/underflow interrupt is not cleared. 1_B Overflow/underflow interrupt is cleared.

T2CON1

This register is being initialized by the BootROM during the boot-up process to a value of 00_H

T2CON1
Timer 2 Control Register 1 **(18_H)** **Reset Value: 03_H**

7	6	5	4	3	2	1	0
RES						**TF2EN**	**EXF2EN**
r						rw	rw

Field	Bits	Type	Description
EXF2EN	0	rw	**External Interrupt Enable** 0_B External interrupt is disabled. 1_B External interrupt is enabled.
TF2EN	1	rw	**Overflow/Underflow Interrupt Enable** 0_B Overflow/underflow interrupt is disabled. 1_B Overflow/underflow interrupt is enabled.
RES	7:2	r	**Reserved** Returns 0 if read; should be written with 0.

16.6.1.3 Timer2 Reload/Capture Register

The RC2 register is used for a 16-bit reload of the timer count upon overflow or a capture of current timer count depending on the mode selected.

RC2L

Timer 2 Reload/Capture Register, Low Byte (08$_H$) Reset Value: 00$_H$

7	6	5	4	3	2	1	0
				RC2			
				rw			

Field	Bits	Type	Description
RC2	[7:0]	rw	**Reload/Capture Value** Note: Reload/Capture Value can be set by software (highest priority) and is updated by hardware during capture mode. These contents are loaded into the timer register upon an overflow condition, if CP_RL2 = 0. If CP_RL2 = 1, this register is loaded with the current timer count upon a negative/positive transition at pin T2EX when EXEN2 = 1.

RC2H

Timer 2 Reload/Capture Register, High Byte (0C$_H$) Reset Value: 00$_H$

7	6	5	4	3	2	1	0
				RC2			
				rw			

Field	Bits	Type	Description
RC2	[7:0]	rw	**Reload/Capture Value** Note: Reload/Capture Value can be set by software (highest priority) and is updated by hardware during capture mode. These contents are loaded into the timer register upon an overflow condition, if CP_RL2 = 0. If CP_RL2 = 1, this register is loaded with the current timer count upon a negative/positive transition at pin T2EX when EXEN2 = 1.

16.6.1.4 Timer2 Count Register

The T2L/T2H register holds the current 16-bit value of the Timer2 count.

T2L
Timer 2, Low Byte **(10_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
T2L							
rwh							

Field	Bits	Type	Description
T2L	[7:0]	rwh	**Timer2 Value** These bits indicate the current timer value T2[7:0]. Note: Timer2 can be updated of software (highest priority) and is updated by hardware if T2R is set

T2H
Timer 2, High Byte **(14_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
T2H							
rwh							

Field	Bits	Type	Description
T2H	[7:0]	rwh	**Timer2 Value** These bits indicate the current timer value T2[15:8]. Note: Timer2 can be updated of software (highest priority) and is updated by hardware if T2R is set

16.6.2 Register Map

There are two kernels in the TLE987x, namely Timer2 and Timer21. **Table 112** shows the Timer2 and Timer21 module base addresses.

Table 113 lists the addresses of the Timer2 and Timer21 SFRs.

Table 112 Register Address Space

Module	Base Address	End Address	Note
TIMER2	48004000_H	$48004FFF_H$	
TIMER21	48005000_H	$48005FFF_H$	

Table 113 Register Overview Timer2 and Timer21

Register Short Name	Register Long Name	Offset Address	Page Number
T2CON	Timer 2 Control Register	00_H	**498**
T2MOD	Timer 2 Mode Register	04_H	**497**
RC2L	Timer 2 Reload/Capture Register, Low Byte	08_H	**500**
RC2H	Timer 2 Reload/Capture Register, High Byte	$0C_H$	**500**
T2L	Timer 2, Low Byte	10_H	**501**
T2H	Timer 2, High Byte	14_H	**501**
T2CON1	Timer 2 Control Register 1	18_H	**499**
T2ICLR	Timer 2 Interrupt Clear Register	$1C_H$	**499**

17 Timer3

17.1 Features

- 16-bit incremental timer/counter (counting up)
- Counting frequency up to f_{sys}
- Selectable clock prescaler
- Six operating modes
- Interrupt on overflow
- Interrupt on compare

17.2 Introduction

The possible applications for this timer include measuring the time interval between events, counting events, and generating a signal at regular intervals.

Timer3 can function as a timer or a counter. When functioning as a timer, Timer3 is incremented in periods based on the MI_CLK or LP_CLK clocks. When functioning as a counter, Timer3 is incremented in response to a 1-to-0 transition (falling edge) at its configured input. Timer3 can be configured in four different operating modes for a variety of applications (see **Table 114**).

The different operating modes allow the timer to be used for tasks such as:

- Simple measurements of the times between two events.
- Triggering the measuring unit upon PWM/CCU6 unit
- Measurement of the 100 kHz LP_CLK2

17.3 Functional description

Six modes of operation are provided to enable using this timer for various tasks. In every mode, the clocking source can be selected from MI_CLK and LP_CLK. In addition, a prescaler provides the capability to divide the selected clock source by 2, 4, or 8. The clocking source and the prescaler can be set in the register **APCLK_CTRL2**.**T3CLK_DIV** (SCU). The timer counts upwards, starting with the value in the timer count registers, up to the maximum count value, which depends on the selected mode of operation. Timer3 provides two individual interrupts on counter overflow, one for the low-byte and one for the high-byte counter register. The run control of the Timer3 is controlled by software by setting the bits TR3L/TR3H in the register **CTRL**. Some modes of operation are providing in addition to the software a run control triggered by various hardware sources coming from other modules within the device. Furthermore the Timer3 provides a 16-bit compare register, **CMP**. The compare register can be used by some operation modes to issue an overflow interrupt upon matching of the timer counter register to the compare register. The entire Timer3 module will be enabled by resetting bit T3_DIS in the register **PMCON2**.

Timer Overflow

When a timer overflow occurs, the timer overflow flag, **T3L_OVF_STS** or **T3H_OVF_STS**, is set, and an interrupt may be raised if the interrupt enable control bit, in the System Control Unit-Power Management is set. The overflow flag has to be cleared when the interrupt service routine is entered.

Note: When Timer3 operates in Mode 3, the Timer3 control bit **TR3L** *is reserved for TL3 and* **TR3H** *is reserved for TH3. For details see in* ***Section 17.3.6****.*

17.3.1 Timer3 modes overview

The Timer3 provides six modes of operation, which are described in the following chapters. The six modes of operations are divided into four main modes. Some of the four main modes are further separated into sub-modes. The bit field T3M in the register **MODE_CONF** selects one out of the four main modes. The bit field T3_SUBM in the register **MODE_CONF** selects a sub-mode of the selected main mode, if applicable. The following table provides an overview of the timer modes together with the reasonable configuration options in **Table 114**.

When the bit **T3_PD_N** is set the timer is reset and the module clock gating is active (valid for all main operating modes).

Table 114 Timer3 modes

Mode	Submode	Operation
0	**No submode**	**13-bit timer** The timer essentially operates an 8-bit counter with a divide-by-32 prescaler.
1	**a**	**16-bit timer** The timer registers, TL3 and TH3, are concatenated to form a 16-bit counter.
1	**b**	**16-bit timer triggered by an event** The timer registers, TL3 and TH3, are concatenated to form a 16-bit counter, which is triggered by an event to enable a single-shot measurement on a preset channel with the measurement unit.
2	**No submode**	**8-bit timer with auto-reload** The timer register TL3 is reloaded with a user-defined 8-bit value in TH3 on overflow.
3	**a**	**Timer3 operating as two 8-bit timers** The timer registers TL3 and TH3, operate as two separate 8-bit counters.
3	**b**	**Timer3 operating as two 8-bit timers for clock measurement** The timer registers, TL3 and TH3, operate as two separate 8-bit counters. In this mode, the LP_CLK2 low power clock can be measured. TL3 acts as an edge counter for the clock edges and TH3 measures the interval between the edges.

The modes are determined in **MODE_CONF**.

Table 115 Timer3, List of Options

		0	**1a**	**1b**	**2**	**3a**	**3b**
Mode Config	**T3M**	0	1	1	2	3	3
	T3_SUBM	0	0	1	0	0	2
Run Control	**TR3L**	1	1	1	1	1	1
	TR3H	n/a	n/a	n/a	n/a	1	1

Table 115 Timer3, List of Options (cont'd)

		0	1a	1b	2	3a	3b
Trigger Source	**CCU6_CC60**	n/a	n/a	1	n/a	n/a	n/a
	CCU6_CC61	n/a	n/a	1	n/a	n/a	n/a
	CCU6_CC62	n/a	n/a	1	n/a	n/a	n/a
	CCU6_PM	n/a	n/a	1	n/a	n/a	n/a
	CCU6_ZM	n/a	n/a	1	n/a	n/a	n/a
	CCU6_COUT60	n/a	n/a	1	n/a	n/a	n/a
	CCU6_COUT61	n/a	n/a	1	n/a	n/a	n/a
	CCU6_COUT62	n/a	n/a	1	n/a	n/a	n/a
	LP_CLK2	n/a	n/a	n/a	n/a	n/a	1
Interrupt Status	**T3L_OVF_STS**	1	1	1	1	1	1
	T3H_OVF_STS	1	1	n/a	0	1	1
Compare	**Compare**	0	0	1	1	0	1

17.3.2 Mode 0

Mode 0 implements a 13-bit-timer/counter compatible to the 8048 microcontroller. **HI** holds the upper 8 bits of the 13-bit timer value. **LO** holds the lower 5 bits of the 13-bit timer value. The bits **LO** [7:5] are not defined and should not be used in this mode of operation. An overflow will be generated by the transition of the timer value from 0x1FFF to 0x0000.

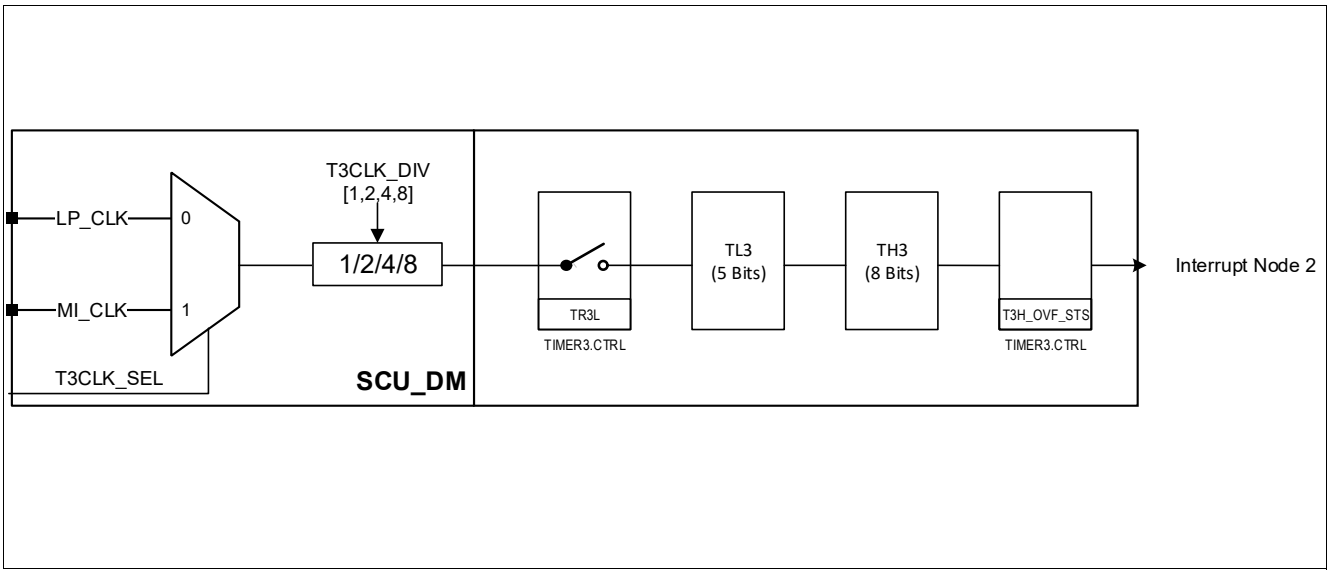


Figure 117 Timer3, Mode 0: 13-Bit Timer

17.3.3 Mode 1a

Mode 1a implements a 16-bit-timer/counter. TIMER3_HI holds the upper 8 bits while TIMER3_LO holds the lower 8 bits of the 16-bit timer value. An overflow will be generated by the transition of the timer value from 0xFFFF to 0x0000.

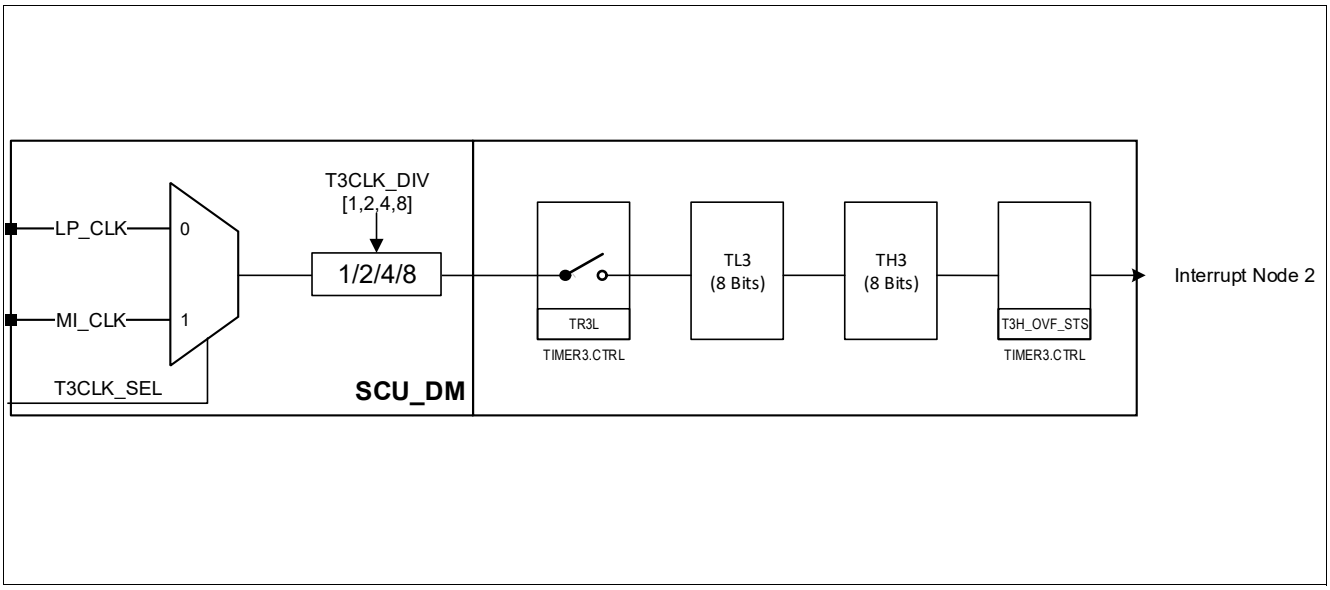


Figure 118 Timer3, Mode 1a: 16-Bit Timer

17.3.4 Mode 1b

Mode 1b is intended to delay certain actions based on a PWM trigger source. The timer operates as a 16-bit-timer. The timer starts counting upwards upon trigger by the selected trigger source. The trigger source has to be selected in the register **T3_TRIGG_CTRL**, bits T3_TRIGG_INP_SEL. To enable a repetitive operation a reset of the timer counter can be defined on the rising edge, falling edge or on both edges of the selected trigger source. The desired selection can be taken in the register **T3_TRIGG_CTRL**, bits T3_RES_CONF. An interrupt will be issued upon overflow, or if the compare value is reached. The compare value may be defined in the registers **CMP**. In this mode the preload of the timer count registers TIMER3_HI and TIMER3_LO has no effect, the counting always starts with 0x0000 until the set compare value.

This mode can be used to trigger the measurement unit with a desired delay based on a PWM edge to perform measurements synchronous to the selected PWM source. The working principle of this mode is shown in **Figure 119**.

Note: If TIMER3_CMP_LO and TIMER3_CMP_HI specify a delay that is greater than the pulse width of the PWM signal, the ccu6_int is not issued.

Note: The TIMER3_CMP value is stored internally (shadowed) on the trigger edge. This means, if the TIMER3_CMP value changes during a current PWM period, the change takes effect with the next PWM period.

Note: Trigger select T12_PM and T12_ZM (in T3_TRIGG_INP_SEL) shall be only operated with rising edge T3_RES_CONF.

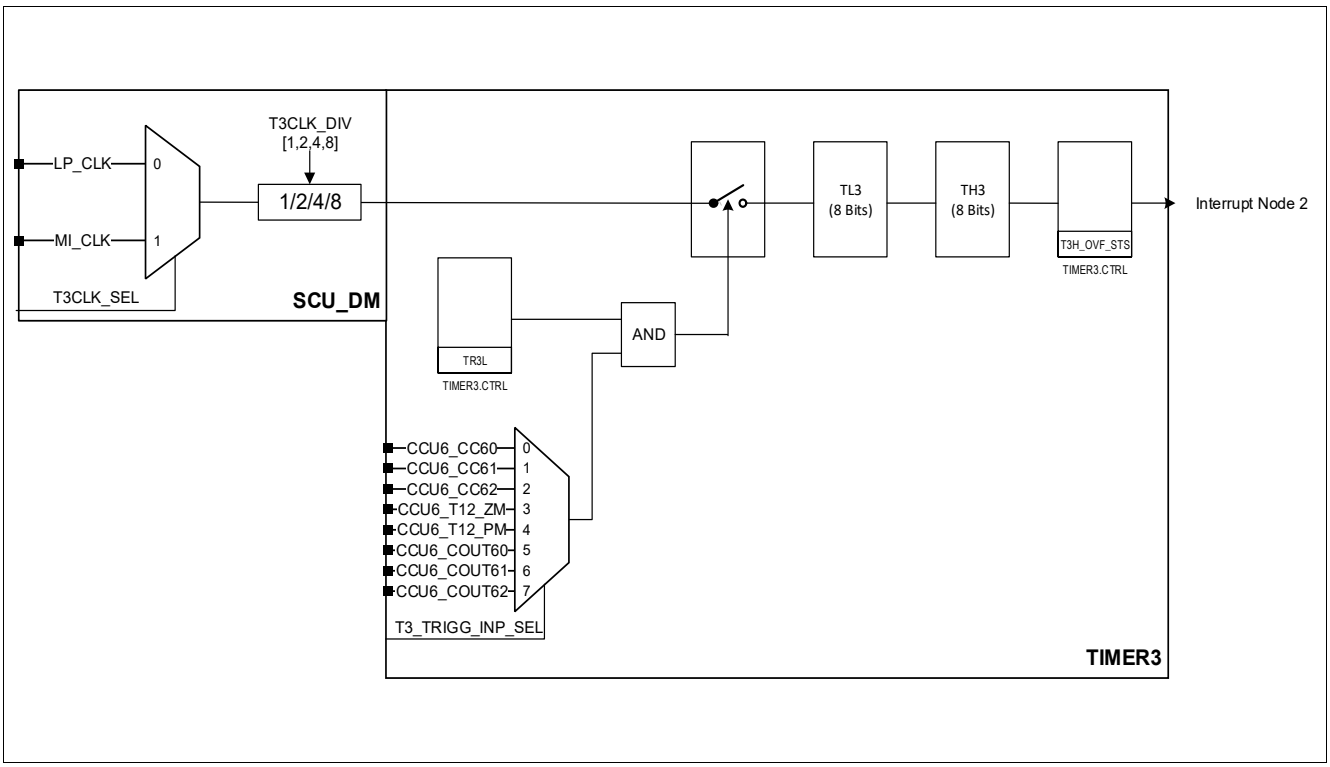


Figure 119 Timer3, Mode 1b: 16-Bit Timer triggered by an event

The retrigger option can be used for CCU6 PM and ZM to be retrigger with the ccu6_int output. T3 is using the inverted ZM/PM signals, and falling edge is recommended as a trigger.

17.3.5 Mode 2

In Mode 2 the timer operates as a 8-bit-timer with reload. The register TIMER3_LOW is the timer value count register, while the register TIMER3_HIGH holds the reload value. Upon an overflow of the TIMER3_LOW register from 0xFF to 0x00 an interrupt gets issued. Simultaneously the TIMER3_LOW register gets loaded with the value in the TIMER3_HIGH register. The working principle is shown in **Figure 120**.

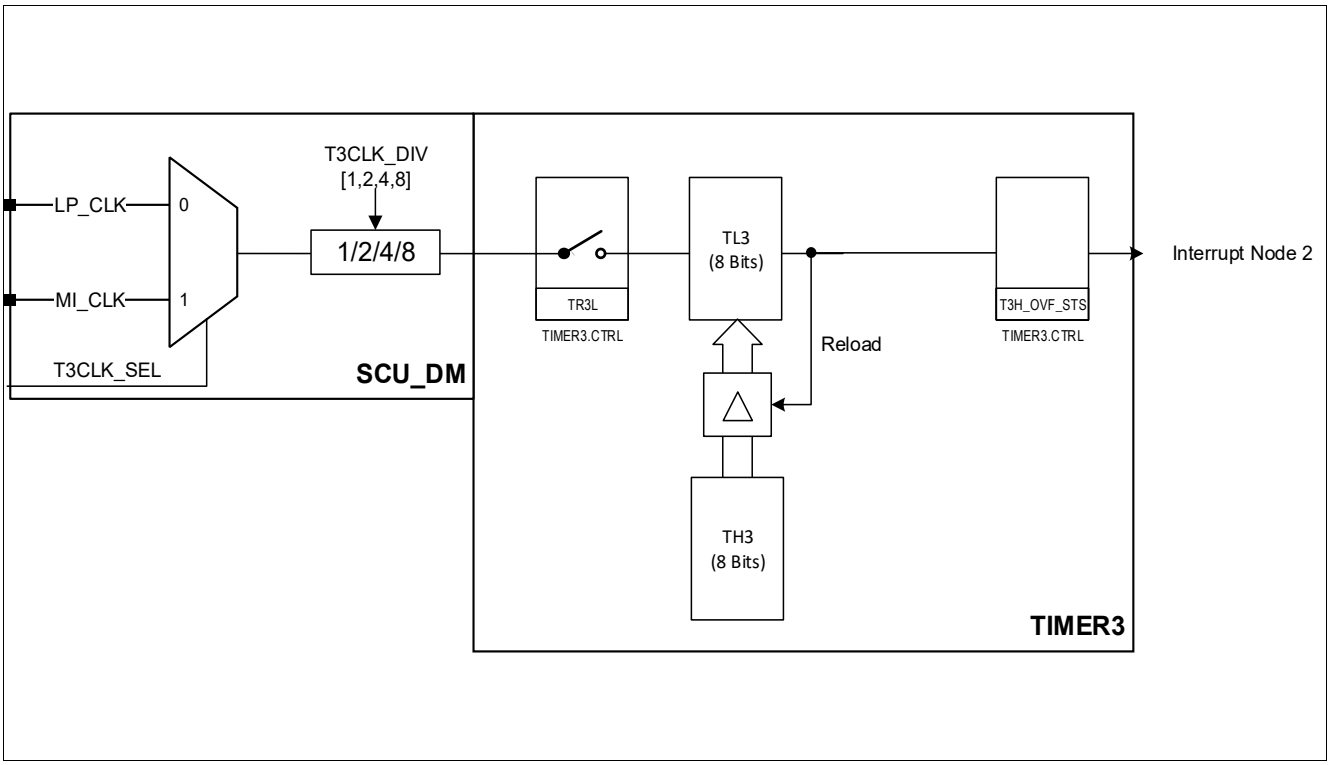


Figure 120 Timer3, Mode 2: 8-Bit Timer with Auto-Reload

17.3.6 Mode 3a

In Mode 3a, the two Timer3 registers, TL3 and TH3, function as two separate 8-bit counters.

The 8-bit counter, TL3, uses the Timer3 control bits TR3 and T3L_OVF_STS, while the other 8-bit counter, TH3, is locked into a timer function (counting machine cycles). Furthermore, TH3 sets the Timer3 flag bit T3H_OVF_STS, upon overflow and generates an interrupt if is set.

Mode 3a is provided for applications requiring an extra 8-bit timer.

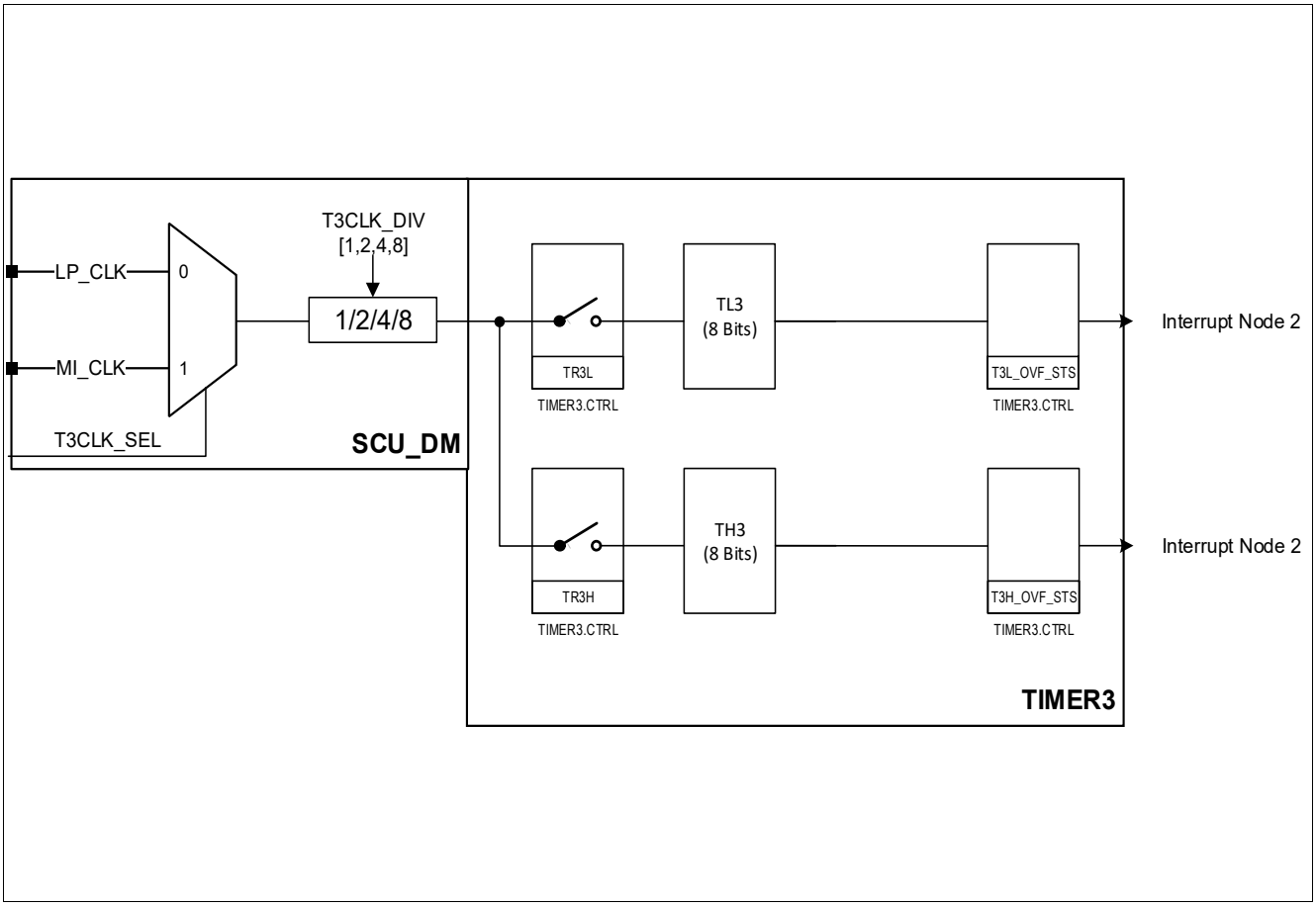


Figure 121 Timer3, Mode 3a: Two 8-Bit Timers

17.3.7 Mode 3b

The Mode 3b is used to measure the period of the LP_CLK2. The Timer3 is split up into an 8-bit counter (TIMER3_LO) and an 8-bit timer (TIMER3_HI). The TIMER3_LO counts falling edges of the LP_CLK2, no trigger source selection is required. The TIMER3_CMP_LO defines how many falling edges shall be counted. The TIMER3_HI runs with the selected clocking source and starts counting with the next falling edge on LP_CLK2. Once TIMER3_LO has counted the desired number of edges of the LP_CLK2, by hitting the TIMER3_CMP_LO value an interrupt will be issued and TIMER3_HI stops.

The recommended sequence in order to obtain correct results in the following:

disable the LP_CLK2 in PMU.**CNF_CYC_SENSE**.OSC_100kHz_EN

configure mode 3b

start the timer3 with the RUN bit

enable the LP_CLK2 in PMU.**CNF_CYC_SENSE**.OSC_100kHz_EN

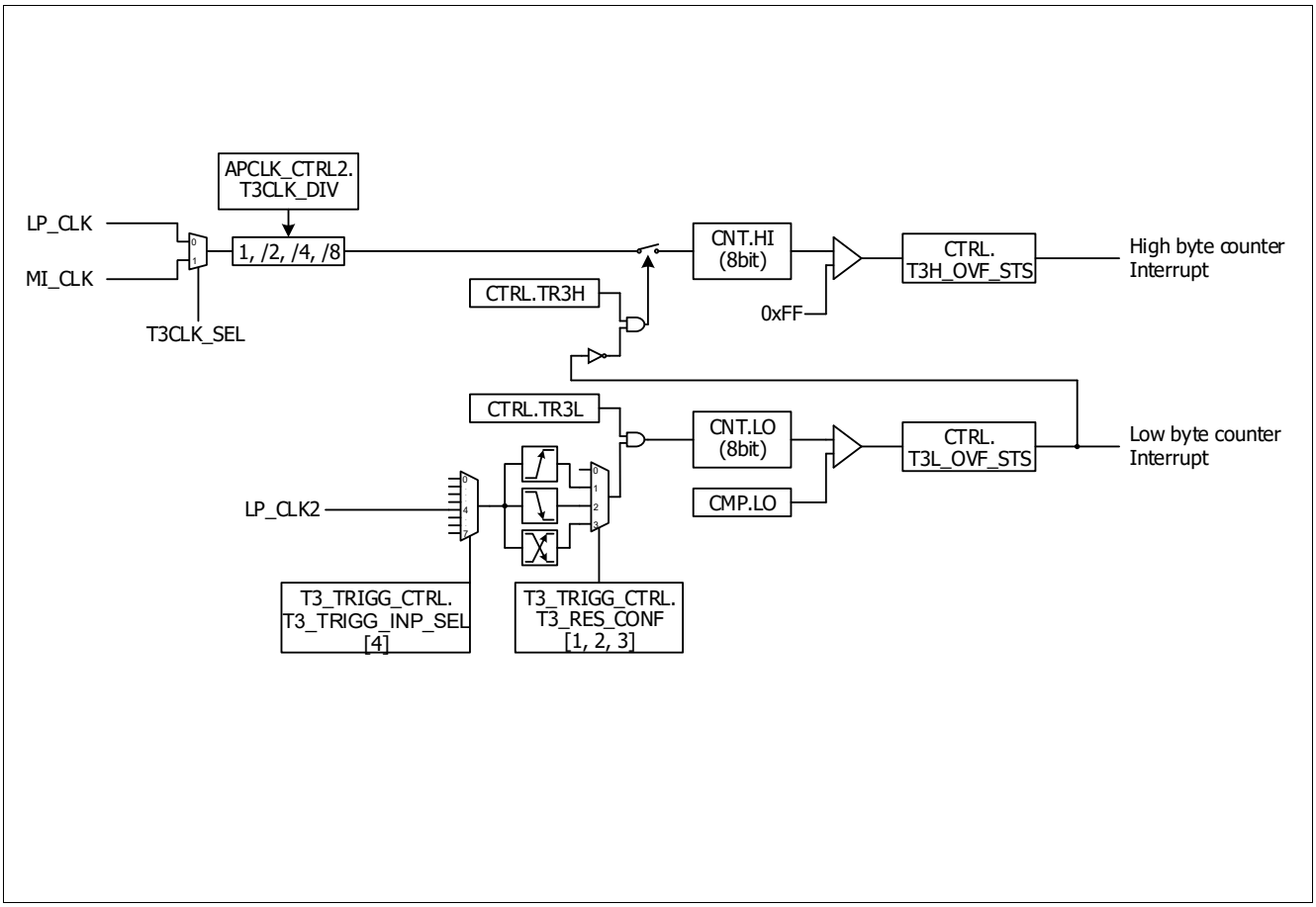


Figure 122 Timer3, Mode 3b: Two 8-Bit Timers for clock measurement

17.4 Interrupts

Figure 123 shows the interrupt generation of Timer3.

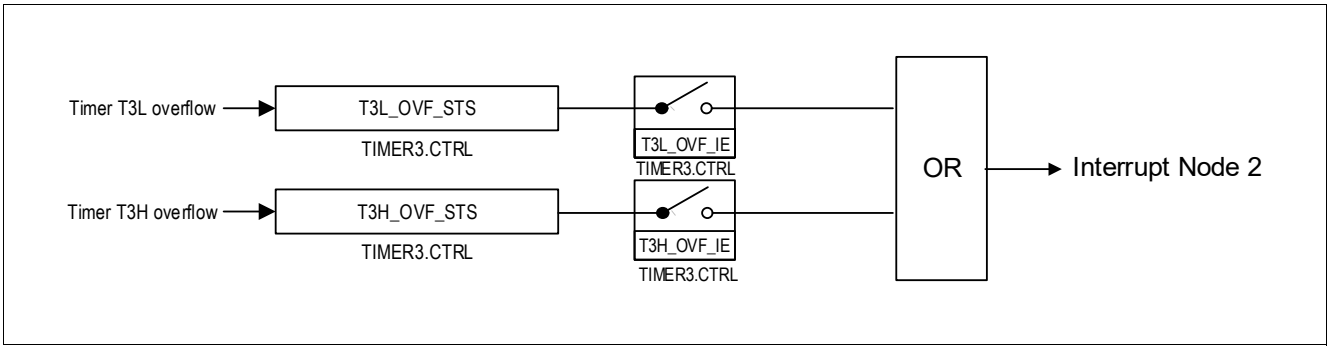


Figure 123 Timer3 Interrupt Generation

The module generates two interrupts. One is assigned to the overflow of lower 8 bit timer, while the other interrupt signals the overflow of the higher 8 bit counter.

17.5 Register Definition

A total of six **SFR** Registers control the operation of Timer3. TL3/TH3 are the low and high timer registers. **CTRL** and **MODE_CONF** are the mode selection register.

The addresses of the kernel SFRs are listed in **Table 117**.

Table 116 shows the module base addresses.

Table 116 Register Address Space

Module	Base Address	End Address	Note
TIMER3	48006000_H	48006020_H	

Table 117 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Register Definition, Timer3 Control Registers			
T3_TRIGG_CTRL	Timer 3 Trigger Control Register	00_H	$0000\ 0000_H$
CMP	Timer 3 Compare Value	04_H	$0000\ 0000_H$
CNT	Timer 3	08_H	$0000\ 0000_H$
CTRL	Timer 3 Control Register	$0C_H$	$0000\ 0001_H$
MODE_CONF	Timer 3 Mode Configuration Register	10_H	$0000\ 0001_H$
ISRCLR	Timer 3 Interrupt Status Clear Register	14_H	$0000\ 0000_H$

The registers are addressed wordwise.

17.5.1 Timer3 Control Registers

After each write access the AHB bus is stalled for at least 2 (timer3_clk) and 3 (hclk) cycles.

Timer 3 Trigger Control Register

The register is reset by RESET_TYPE_3.

T3_TRIGG_CTRL
Timer 3 Trigger Control Register **(00_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								RES	RETRIG	T3_RES_CONF		RES	T3_TRIGG_INP_SEL		
r								r	rw	rw		r	rw		

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
RES	7	r	**Reserved** Always read as 0
RETRIG	6	rw	**Retrigger Condition (in Mode 1b) for CCU6-T12 ZM and CCU6-T12 PM** For CCU6-T12 ZM and PM the ccu6_int output will retrigger the timer in Mode 1b. 0_B **DIS**, Retrigger disabled 1_B **EN**, Retrigger enabled
T3_RES_CONF	5:4	rw	**Timer3 Trigger Reset Selection for Mode 1b** 0_H **No Reset on PWM Edge** , Counter is not reset while PWM Module is running. 1_H **Reset On Rising Edge**, Counter is reset on rising edge input 2_H **Reset On Falling Edge**, Counter is reset on falling edge input. 3_H **Reset on both Edges**, Counter is reset on both edge inputs.
RES	3	r	**Reserved** Always read as 0

Field	Bits	Type	Description
T3_TRIGG_INP_SEL	2:0	rw	**Timer3 Trigger Input Event Selection in Mode 1b** 0_H **CCU6-CC6 0**, Capture Compare Unit Channel 0 (CC60). 1_H **CCU6-CC61**, Capture Compare Unit Channel 1 (CC61). 2_H **CCU6-CC62**, Capture Compare Unit Channel 2 (CC62). 3_H **CCU6-T12 ZM**, Capture Compare Unit T12 Zero Match. 4_H **CCU6-T12 PM**, Capture Compare Unit T12 Period Match. 5_H **CCU6-COUT60**, Capture Compare Unit Channel 0 (COUT60). 6_H **CCU6-COUT61**, Capture Compare Unit Channel 1 (COUT61). 7_H **CCU6-COUT62**, Capture Compare Unit Channel 2 (COUT62). **Timer3 Trigger Input Event Selection in Mode 3b** 4_H **LP_CLK2**, 100kHz low precision clock, enable it here **CNF_CYC_SENSE**.

Timer 3 Compare Value

The register is reset by RESET_TYPE_3.

CMP
Timer 3 Compare Value **(04_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HI								**LO**							
rw								rw							

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
HI	15:8	rw	**Timer3 Compare Value High Byte** 00000000_B**TIMER3_CMP_HI**, holds the compare value of high byte for Measurement Interface Trigger.
LO	7:0	rw	**Timer3 Compare Value Low Byte** 00000000_B**TIMER3_CMP_LO**, holds the compare value of low byte for Measurement Interface Trigger.

Timer 3

The register is reset by RESET_TYPE_3.

CNT
Timer 3 **(08_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HI								LO							
rw								rw							

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
HI	15:8	rw	**Timer3 High Register or Preload Value** 00_B **TIMER3_HI**, holds the higher 8-bit part of the 13-bit timer value. 01_B **TIMER3_HI**, holds the higher 8-bit part of the 16-bit timer value. 10_B **TIMER3_HI**, holds the 8-bit reload value. 11_B **TIMER3_HI**, holds the 8-bit timer value.
LO	7:0	rw	**Timer3 Low Register or Preload Value**[1)] 00_B **TIMER3_LO**, holds the lower 5-bit part of the 13-bit timer value. 01_B **TIMER3_LO**, holds the lower 8-bit part of the 16-bit timer value. 10_B **TIMER3_LO**, holds the 8-bit timer value. 11_B **TIMER3_LO**, holds the 8-bit timer value.

1) TIMER3_LO corresponds to the selected timer main operating mode which can be changed by writing **MODE_CONF**.

Timer 3 Control Register

The register is reset by RESET_TYPE_3.

CTRL
Timer 3 Control Register **($0C_H$)** **Reset Value: $0000\ 0001_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						T3H_OVF_IE	T3L_OVF_IE	T3L_OVF_STS	TR3L	T3H_OVF_STS	TR3H	CNT_RDY	T3_RD_REQ_CONF	T3_RD_REQ	T3_PD_N
r						rw	rw	r	rw	r	rwd	rwh	rw	rwh1	rw

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
T3H_OVF_IE	9	rw	**Timer3 Overflow Interrupt Enable (High Byte Timer)** 0_B **DIS**, Interrupt disabled 1_B **EN**, Interrupt enabled
T3L_OVF_IE	8	rw	**Timer3 Overflow Interrupt Enable (Low Byte Timer)** 0_B **DIS**, Interrupt disabled. 1_B **EN**, Interrupt enabled
T3L_OVF_STS	7	r	**Timer3 Overflow Flag (Low Byte Timer)** 0_B **T3L_OVF_STS**, No overflow occurred. 1_B **T3L_OVF_STS**, Overflow occurred. Set by hardware when Low Byte of Timer3 overflows. Cleared by software.
TR3L	6	rw	**Timer3 Run Control (Low Byte Timer)** 0_B **TR3L**, Timer is halted 1_B **TR3L**, Timer runs
T3H_OVF_STS	5	r	**Timer3 Overflow Flag (High Byte Timer)** 0_B **T3H_OVF_STS**, No Overflow occurred. 1_B **T3H_OVF_STS**, Overflow occurred. Set by hardware when High Byte of Timer3 overflows. Cleared by software.
TR3H	4	rwd	**Timer3 Run Control (High Byte Timer)** 0_B **TR3H**, Timer is halted 1_B **TR3H**, Timer runs
CNT_RDY	3	rwh	**Timer3 Count Ready** 0_B **CNT_RDY**, Timer hasn't finished counting in Mode 1b, 3b 1_B **CNT_RDY**, Timer has finished counting in Mode 1b, 3b

Timer3

Field	Bits	Type	Description
T3_RD_REQ_CONF	2	rw	**Timer3 Read Mode** 0_B **T3_RD_REQ_CONF**, Timer3 Read Request can be triggered by software 1_B **T3_RD_REQ_CONF**, Timer3 Read Request can be triggered by hardware (in Mode 3b) *Note: Hardware read request is ignored in modes other than Mode 3b.* *T3_RD_REQ_CONF and T3_RD_REQ need to updated in two separate write accesses.*
T3_RD_REQ	1	rwh1	**Timer3 Value Read Request** 0_B **T3_RD_REQ**, Timer value is not read from Timer3 1_B **T3_RD_REQ**, Timer value is read from Timer3 *Note: Only possible if T3_RD_REQ_CONF = 0* *This bit T3_RD_REQ remains set until the read request is finished, and only then is cleared to 0*
T3_PD_N	0	rw	**Timer3 Power Down** 0_B **Power Down**, Timer3 is in Power Down 1_B **no Power Down**, Timer3 is not in Power Down *Note: In Power Down Mode, the Timer3 is reset and the module clock gating is active*

Timer 3 Interrupt Status Clear Register

The register is reset by RESET_TYPE_3.

ISRCLR

Timer 3 Interrupt Status Clear Register **(14_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								T3L_OVF_ICLR	RES	T3H_OVF_ICLR	RES				
r								w	r	w	r				

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
T3L_OVF_ICLR	7	w	**Timer3 Overflow Flag (Low Byte Timer) Interrupt Clear** 0_B **T3L_OVF_ ICLR**, Overflow not cleared. 1_B **T3L_OVF_ ICLR**, Overflow cleared. Set by software, cleared by hardware.
RES	6	r	**Reserved** Always read as 0
T3H_OVF_ICLR	5	w	**Timer3 Overflow Flag (High Byte Timer) Interrupt Clear** 0_B **T3H_OVF_ ICLR**, Overflow not cleared. 1_B **T3H_OVF_ ICLR**, Overflow cleared. Set by software, cleared by hardware
RES	4:0	r	**Reserved** Always read as 0

Timer 3 Mode Configuration Register

The register is reset by RESET_TYPE_3.

MODE_CONF
Timer 3 Mode Configuration Register **(10_H)** **Reset Value: $0000\ 0001_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								T3_SUBM		RES				T3M	
r								rw		r				rw	

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
T3_SUBM	7:6	rw	**Sub-Mode Select Bits** 00_B **No Sub-Mode**, no Sub-Mode enabled 01_B **Mode 1b**, enables 16-bit Timer triggered by an event. This mode has only an effect with Mode 1 (16-Bit Mode) 10_B **Mode 3b**, enables two 8-Bit Timers for clock measurement. This Mode has only an effect with Mode 3. 11_B **RES**, Reserved.
RES	5:2	r	**Reserved** Always read as 0
T3M	1:0	rw	**Mode Select Bits** 00_B **T3M**, 13-bit timer 01_B **T3M**, 16-bit timer 10_B **T3M**, 8-bit auto-reload timer 11_B **T3M**, Timer3 is split into two halves. TL3 is an 8-bit timer controlled by the standard Timer3 low byte control bits, and TH3 is the other 8-bit timer controlled by the standard Timer3 high byte control bits.

18 Capture/compare unit 6 (CCU6)

The CCU6 is a high-resolution 16-bit capture and compare unit with application specific modes, mainly for AC drive control. Special operating modes support the control of brushless DC-motors using Hall sensors or Back-EMF detection. Furthermore, block commutation and control mechanisms for multi-phase machines are supported.

It also supports inputs to start several timers synchronously, an important feature in devices with several CCU6 modules.

This chapter is structured as follows:

- Functional description of the CCU6 kernel (see **Section 18.2**)
  - Introduction (see **Section 18.2**)
  - Operating T12 (see **Section 18.3**)
  - Operating T13 (see **Section 18.4**)
  - Trap handling (see **Section 18.5**)
  - Multi-Channel mode (see **Section 18.6**)
  - Hall sensor mode (see **Section 18.7**)
  - Interrupt handling (see **Section 18.10**)
  - General module operation (see **Section 18.8**)
- CCU6 kernel registers description (see **Section 18.11.1**)
- TLE987x implementation specific details (see **Section 18.9**)

18.1 Feature set overview

This section gives an overview over the different building blocks and their main features.

Timer12 block features

- Three capture/compare channels. Each channel can be used either as capture or as compare channel.
- Supports three-phase PWM (six outputs with separate signals for high-side and low-side switches).
- 16-bit resolution, maximum count frequency = peripheral clock.
- Dead-time control for each channel to avoid short-circuits in the power stage.
- Concurrent update of the T12 registers.
- Center-aligned and edge-aligned PWM can be generated.
- Single-shot mode is supported.
- Start can be controlled by external events.
- External events can be counted.
- Multiple interrupt request sources.
- Hysteresis-like control mode.

Timer13 block features

- One independent compare channel with one output.
- 16-bit resolution, maximum count frequency = peripheral clock.
- Concurrent update of T13 registers.
- Can be synchronized to T12.

- Interrupt generation at period-match and compare-match.
- Single-shot mode is supported.
- Start can be controlled by external events.
- Capability of counting external events.

Additional specific functions

- Block commutation for brushless DC-drives implemented.
- Position detection via hall-sensor pattern.
- Noise filter for position input signals supported.
- Automatic rotational speed measurement and commutation control for block commutation.
- Integrated error handling.
- Fast emergency stop without CPU load via external signal ($\overline{\text{CTRAP}}$).
- Control modes for multi-channel AC drives.
- Output levels can be selected and adapted to the power stage.

18.2 Introduction

The CCU6 unit is made up of the T12 timer block with three capture/compare channels and the T13 timer block with one compare channel. The T12 channels can independently generate PWM signals or accept capture triggers, or they can jointly generate control signal patterns to drive DC motors or inverters.

A rich set of status bits, synchronized updating of parameter values via shadow registers, and flexible generation of interrupt request signals provide efficient software control.

Note: The capture/compare module itself is referred to as CCU6 (capture/compare unit 6). A capture/compare channel inside this module is referred to as CC6x.

The timer T12 can work in capture and/or compare mode for its three channels. The modes can also be combined (e.g., a channel works in compare mode, whereas another channel works in capture mode). The timer T13 can work only in compare mode. The multi-channel control unit generates output patterns which can be modulated by T12 and/or T13. The modulation sources can be selected and combined for modulating signals.

18.2.1 Block diagram

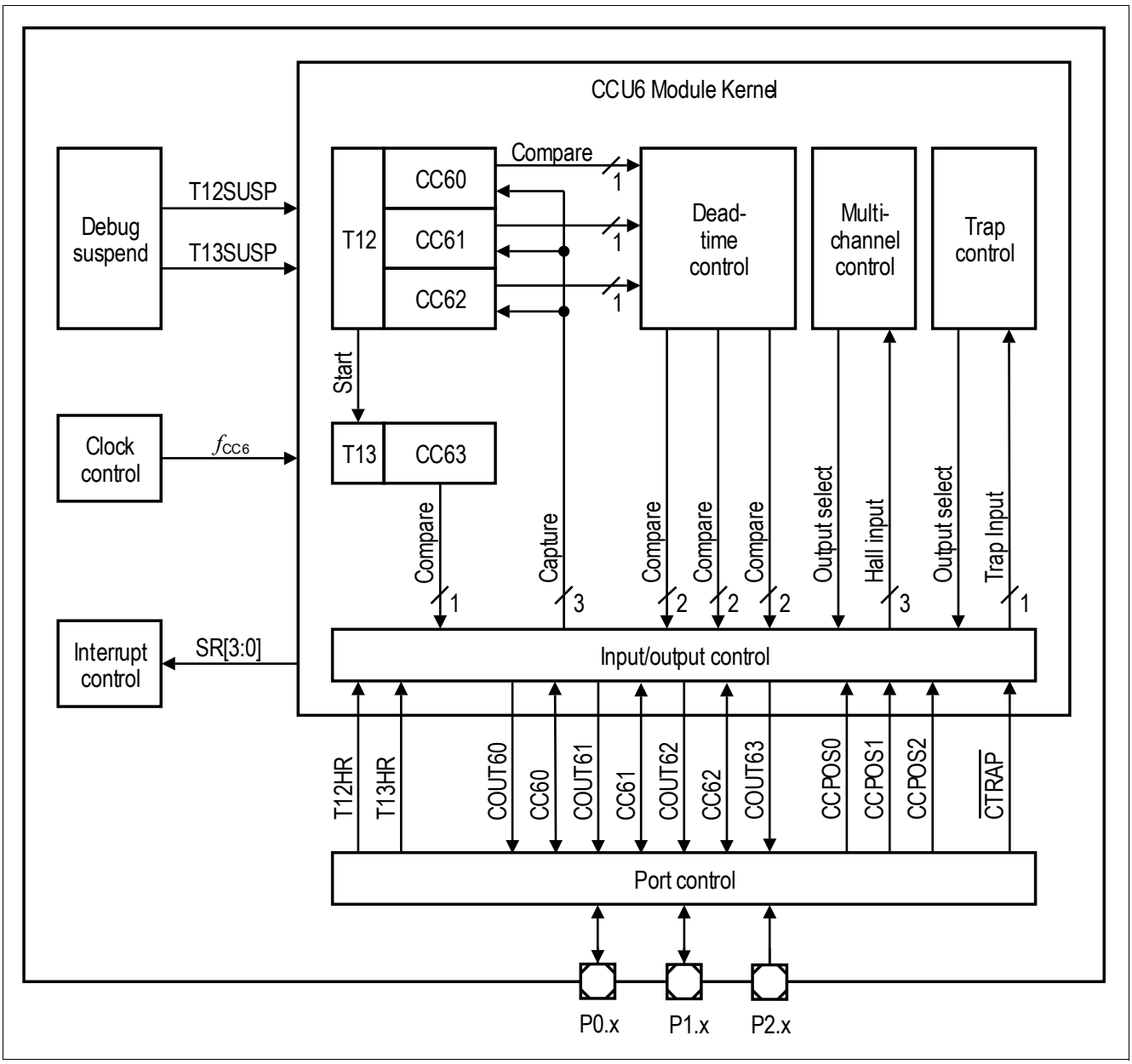


Figure 124 CCU6 block diagram

18.3 Operating Timer T12

The timer T12 block is the main unit to generate the 3-phase PWM signals. A 16-bit counter is connected to 3 channel registers via comparators, that generate a signal when the counter contents match one of the channel register contents. A variety of control functions facilitate the adaptation of the T12 structure to different application needs.

Besides the 3-phase PWM generation, the T12 block offers options for individual compare and capture functions, as well as dead-time control and hysteresis-like compare mode.

This section provides information about:

- T12 overview (see **Section 18.3.1**)
- Counting scheme (see **Section 18.3.2**)
- Compare modes (see **Section 18.3.3**)
- Compare mode output path (see **Section 18.3.4**)
- Capture modes (see **Section 18.3.5**)
- Shadow transfer (see **Section 18.3.6**)
- T12 operating mode selection (see **Section 18.3.7**)

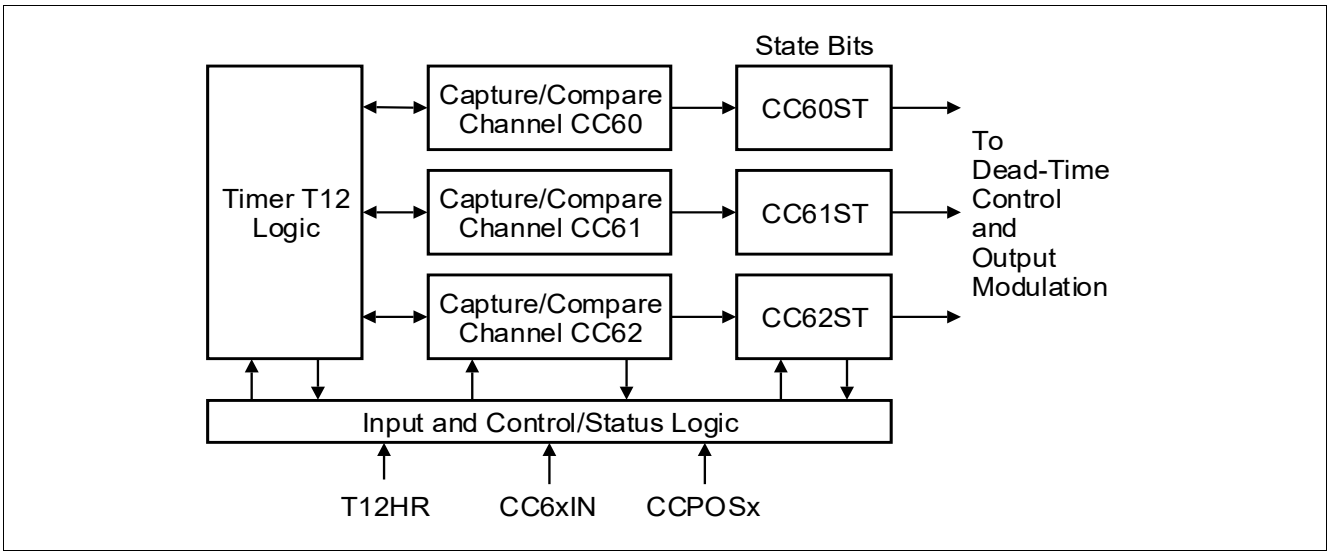


Figure 125 Overview Diagram of the Timer T12 Block

18.3.1 T12 Overview

Figure 126 shows a detailed block diagram of Timer T12. The functions of the timer T12 block are controlled by bits in registers TCTR0, TCTR2, and PISEL0.

Timer T12 receives its input clock (f_{T12}) from the module clock f_{CC6} via a programmable prescaler and an optional 1/256 divider or from an input signal T12HR. These options are controlled via bit fields T12CLK and T12PRE (see **Table 118**). T12 can count up or down, depending on the selected operation mode. A direction flag, CDIR, indicates the current counting direction.

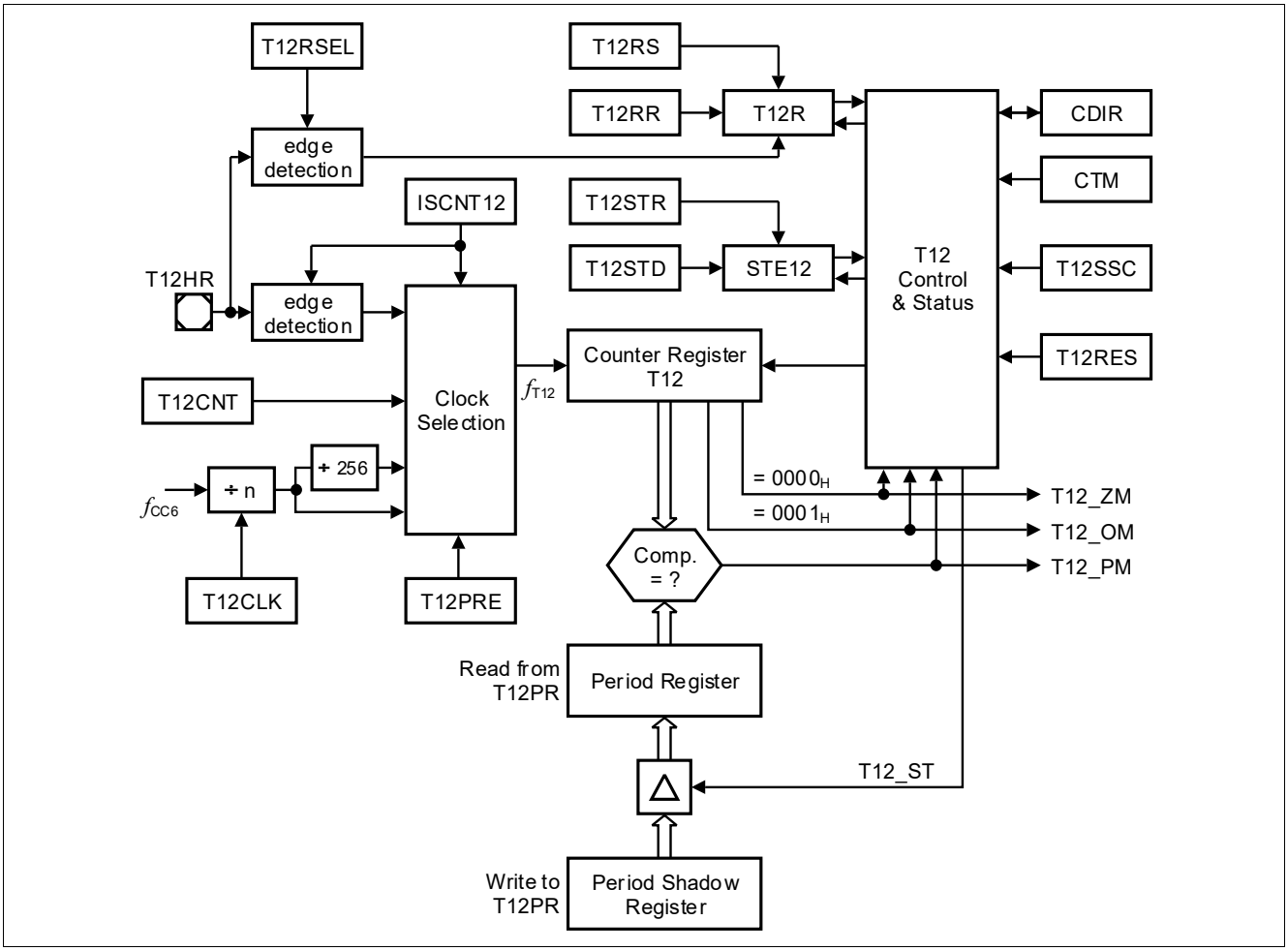


Figure 126 Timer T12 Logic and Period Comparators

Via a comparator, the T12 counter register T12 is connected to a Period Register T12PR. This register determines the maximum count value for T12.

In Edge-Aligned mode, T12 is cleared to 0000_H after it has reached the period value defined by T12PR. In Center-Aligned mode, the count direction of T12 is set from 'up' to 'down' after it has reached the period value (please note that in this mode, T12 exceeds the period value by one before counting down). In both cases, signal T12_PM (T12 Period Match) is generated. The Period Register receives a new period value from its Shadow Period Register.

A read access to T12PR delivers the current period value at the comparator, whereas a write access targets the Shadow Period Register to prepare another period value. The transfer of a new period value from the Shadow Period Register into the Period Register (see **Section 18.3.6**) is controlled via the 'T12 Shadow Transfer' control signal, T12_ST. The generation of this signal depends on the operating mode and on the shadow

transfer enable bit STE12. Providing a shadow register for the period value as well as for other values related to the generation of the PWM signal allows a concurrent update by software for all relevant parameters.

Two further signals indicate whether the counter contents are equal to 0000_H (T12_ZM = zero match) or 0001_H (T12_OM = one match). These signals control the counting and switching behavior of T12.

The basic operating mode of T12, either Edge-Aligned mode (**Figure 127**) or Center-Aligned mode (**Figure 128**), is selected via bit CTM. A Single-Shot control bit, T12SSC, enables an automatic stop of the timer when the current counting period is finished (see **Figure 129** and **Figure 130**).

The start or stop of T12 is controlled by the Run bit T12R that can be modified by bits in register TCTR4. The run bit can be set/cleared by software via the associated set/clear bits T12RS or T12RR, it can be set by a selectable edge of the input signal T12HR (TCTR2.T12RSEL), or it is cleared by hardware according to preselected conditions.

The timer T12 run bit T12R must not be set while the applied T12 period value is zero. Timer T12 can be cleared via control bit T12RES. Setting this write-only bit does only clear the timer contents, but has no further effects, for example, it does not stop the timer.

The generation of the T12 shadow transfer control signal, T12_ST, is enabled via bit STE12. This bit can be set or reset by software indirectly through its associated set/clear control bits T12STR and T12STD.

While Timer T12 is running, write accesses to the count register T12 are not taken into account. If T12 is stopped and the Dead-Time counters are 0, write actions to register T12 are immediately taken into account.

18.3.2 T12 Counting Scheme

This section describes the clocking and counting capabilities of T12.

18.3.2.1 Clock Selection

In **Timer Mode** (PISEL2.ISCNT12 = 00_B), the input clock f_{T12} of Timer T12 is derived from the internal module clock f_{CC6} through a programmable prescaler and an optional 1/256 divider. The resulting prescaler factors are listed in **Table 118**. The prescaler of T12 is cleared while T12 is not running (TCTR0.T12R = 0) to ensure reproducible timings and delays.

Table 118 Timer T12 Input Frequency Options

T12CLK	Resulting Input Clock f_{T12} Prescaler Off (T12PRE = 0)	Resulting Input Clock f_{T12} Prescaler On (T12PRE = 1)
000_B	f_{CC6}	f_{CC6} / 256
001_B	f_{CC6} / 2	f_{CC6} / 512
010_B	f_{CC6} / 4	f_{CC6} / 1024
011_B	f_{CC6} / 8	f_{CC6} / 2048
100_B	f_{CC6} / 16	f_{CC6} / 4096
101_B	f_{CC6} / 32	f_{CC6} / 8192
110_B	f_{CC6} / 64	f_{CC6} / 16384
111_B	f_{CC6} / 128	f_{CC6} / 32768

In **Counter Mode**, timer T12 counts one step:

- If a 1 is written to TCTR4.T12CNT and PISEL2.ISCNT12 = 01_B
- If a rising edge of input signal T12HR is detected and PISEL2.ISCNT12 = 10_B
- If a falling edge of input signal T12HR is detected and PISEL2.ISCNT12 = 11_B

18.3.2.2 Edge-Aligned / Center-Aligned Mode

In **Edge-Aligned Mode** (CTM = 0), timer T12 is always counting upwards (CDIR = 0). When reaching the value given by the period register (period-match T12_PM), the value of T12 is cleared with the next counting step (saw tooth shape).

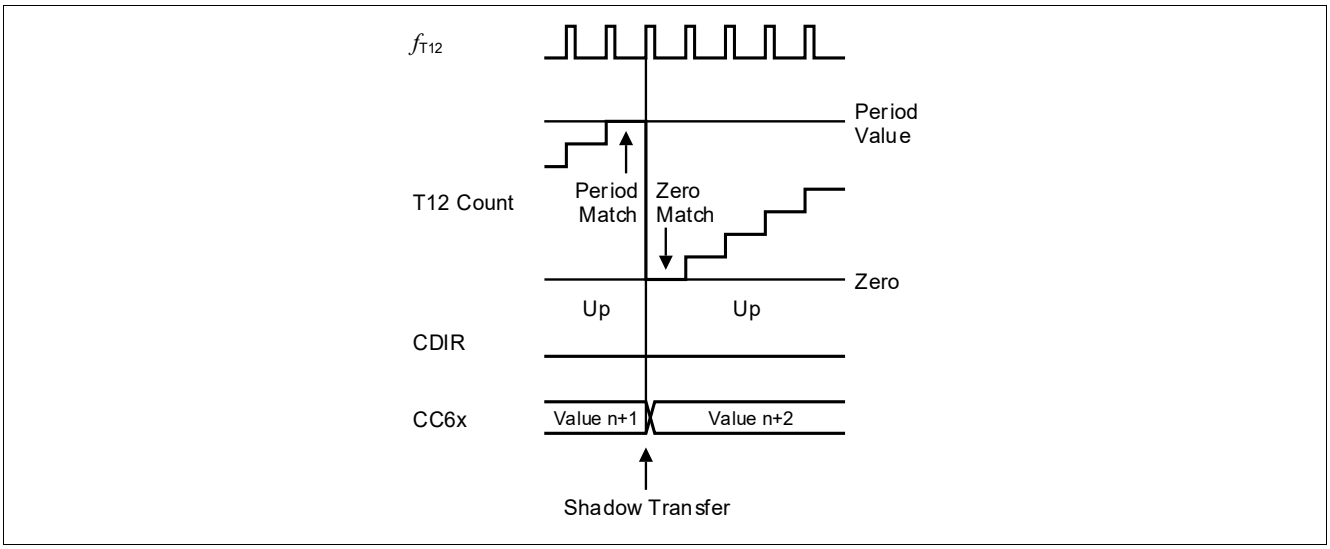


Figure 127 T12 Operation in Edge-Aligned Mode

As a result, in Edge-Aligned mode, the timer period is given by:

$T12_{PER}$ = <Period-Value> + 1; in T12 clocks (f_{T12}) (18.1)

In **Center-Aligned Mode** (CTM = 1), timer T12 is counting upwards or downwards (triangular shape). When reaching the value given by the period register (period-match T12_PM) while counting upwards (CDIR = 0), the counting direction control bit CDIR is changed to downwards (CDIR = 1) with the next counting step.

When reaching the value 0001_H (one-match T12_OM) while counting downwards, the counting direction control bit CDIR is changed to upwards with the next counting step.

As a result, in Center-Aligned mode, the timer period is given by:

$T12_{PER}$ = (<Period-Value> + 1) x2; in T12 clocks (f_{T12}) (18.2)

- With the next clock event of f_{T12} the count direction is set to counting up (CDIR = 0) when the counter reaches 0001_H while counting down.
- With the next clock event of f_{T12} the count direction is set to counting down (CDIR = 1) when the Period-Match is detected while counting up.
- With the next clock event of f_{T12} the counter counts up while CDIR = 0 and it counts down while CDIR = 1.

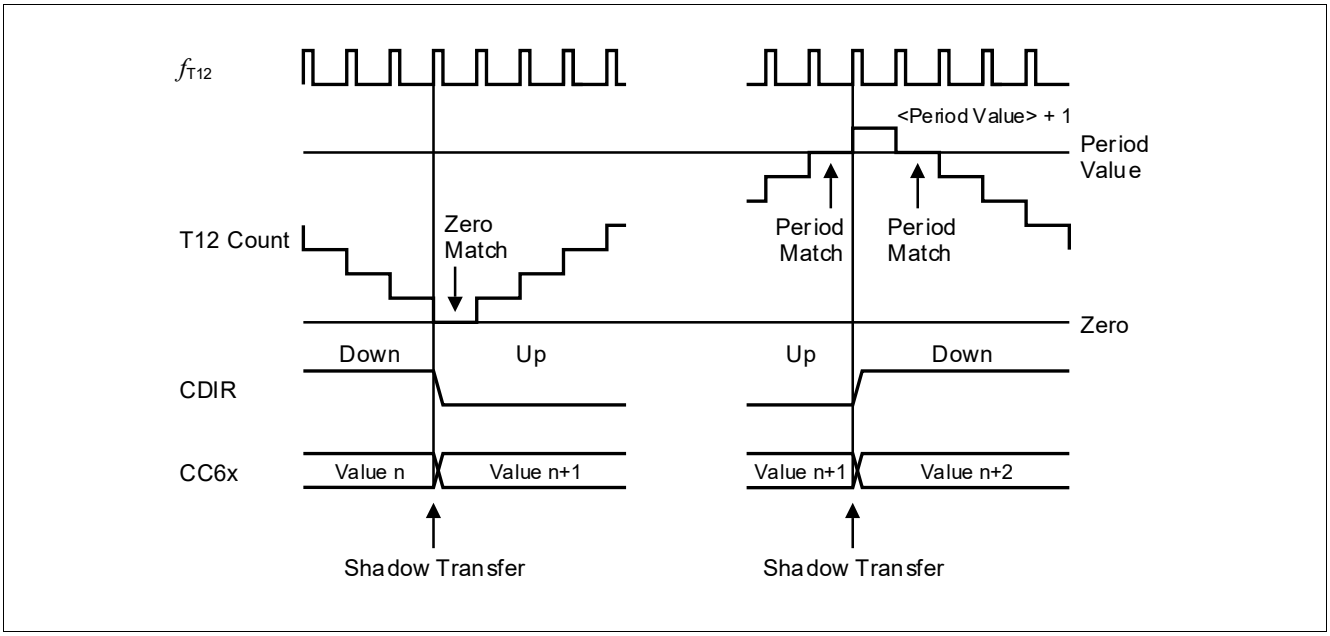


Figure 128 T12 Operation in Center-Aligned Mode

*Note: Bit CDIR changes with the next timer clock event after the one-match or the period-match. Therefore, the timer continues counting in the previous direction for one cycle before actually changing its direction (see **Figure 128**).*

18.3.2.3 Single-Shot Mode

In Single-Shot Mode, the timer run bit T12R is cleared by hardware. If bit T12SSC = 1, the timer T12 will stop when the current timer period is finished.

In Edge-Aligned mode, T12R is cleared when the timer becomes zero after having reached the period value (see **Figure 129**).

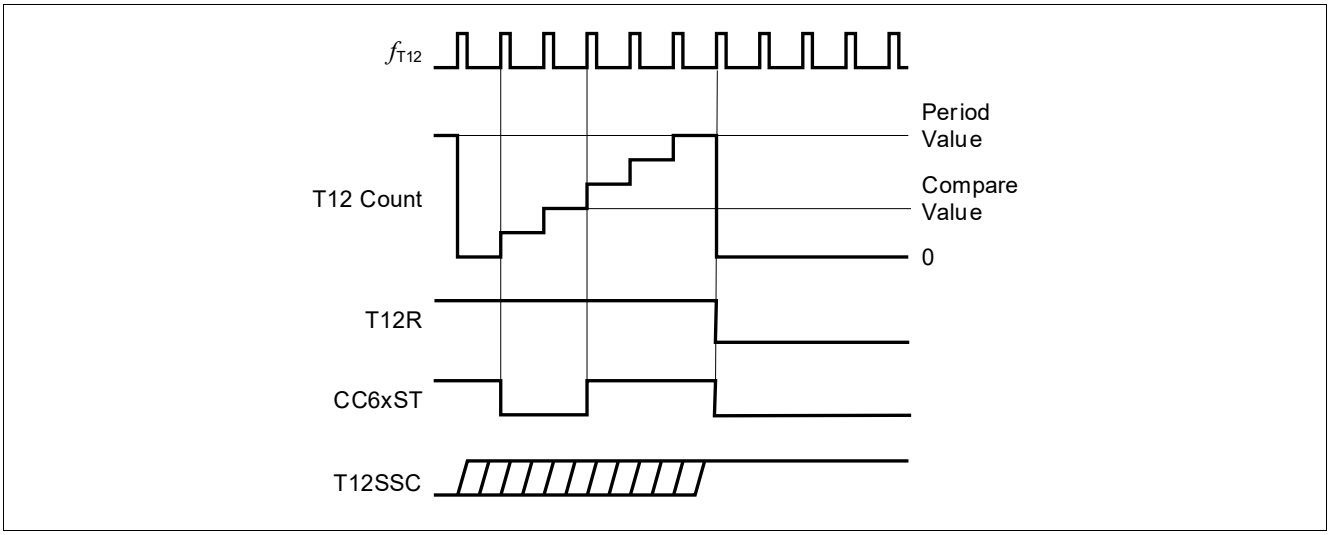


Figure 129 Single-Shot Operation in Edge-Aligned Mode

In Center-Aligned mode, the period is finished when the timer has counted down to zero (one clock cycle after the one-match while counting down, see **Figure 130**).

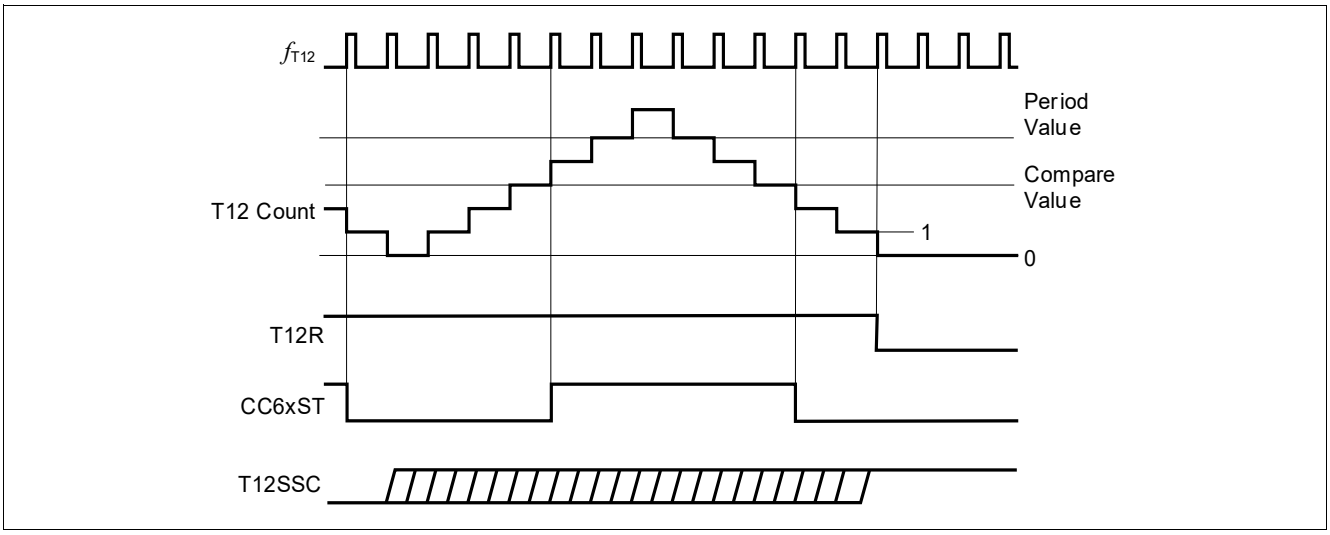


Figure 130 Single-Shot Operation in Center-Aligned Mode

18.3.3 T12 Compare Mode

Associated with Timer T12 are three individual capture/compare channels, that can perform compare or capture operations with regard to the contents of the T12 counter. The capture functions are explained in **Section 18.3.5**.

18.3.3.1 Compare Channels

In Compare Mode (see **Figure 131**), the three individual compare channels CC60 CC61, and CC62 can generate a three-phase PWM pattern.

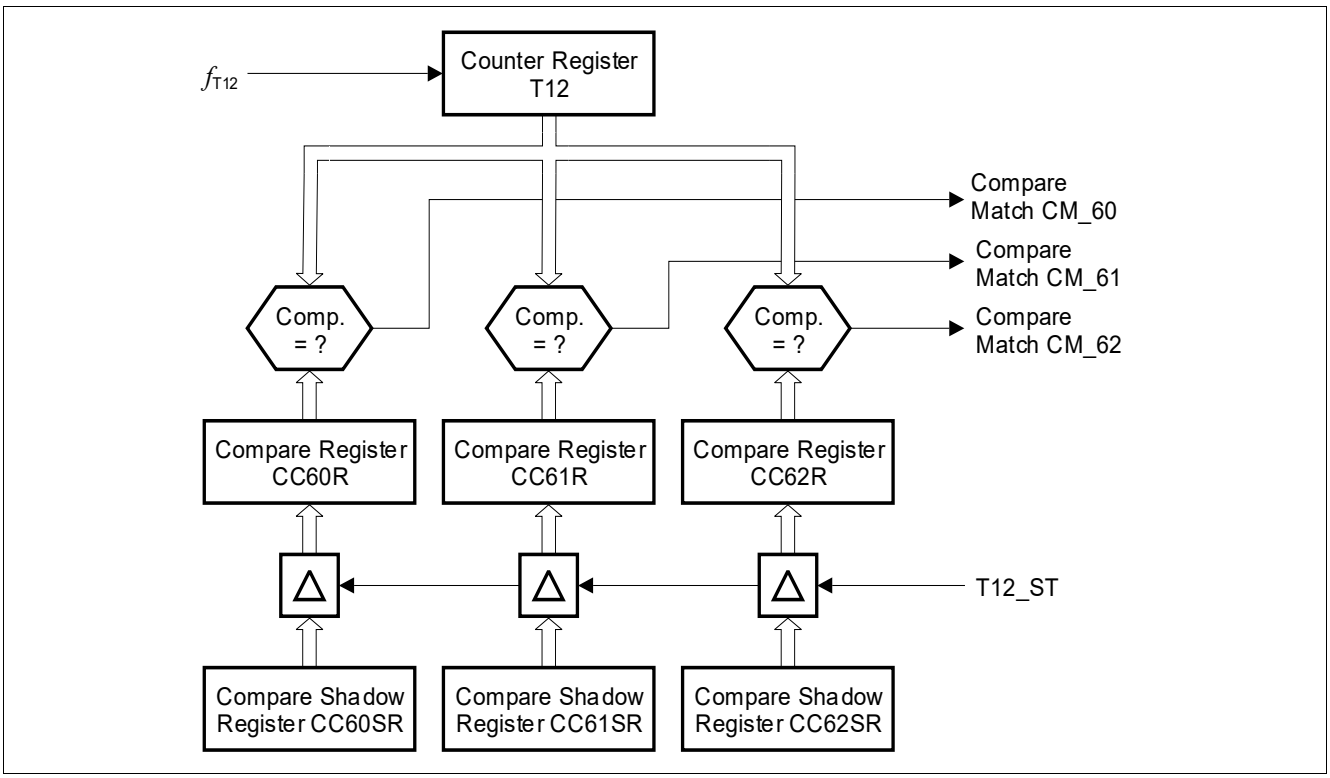


Figure 131 T12 Channel Comparators

Each compare channel is connected to the T12 counter register via its individual equal-to comparator, generating a match signal when the contents of the counter matches the contents of the associated compare register. Each channel consists of the comparator and a double register structure - the actual compare register CC6xR, feeding the comparator, and an associated shadow register CC6xSR, that is preloaded by software and transferred into the compare register when signal T12 shadow transfer, T12_ST, gets active. Providing a shadow register for the compare value as well as for other values related to the generation of the PWM signal facilitates a concurrent update by software for all relevant parameters of a three-phase PWM.

18.3.3.2 Channel State Bits

Associated with each (compare) channel is a State Bit, CMPSTAT.CC6xST, holding the status of the compare (or capture) operation (see **Figure 132**). In compare mode, the State Bits are modified according to a set of switching rules, depending on the current status of timer T12.

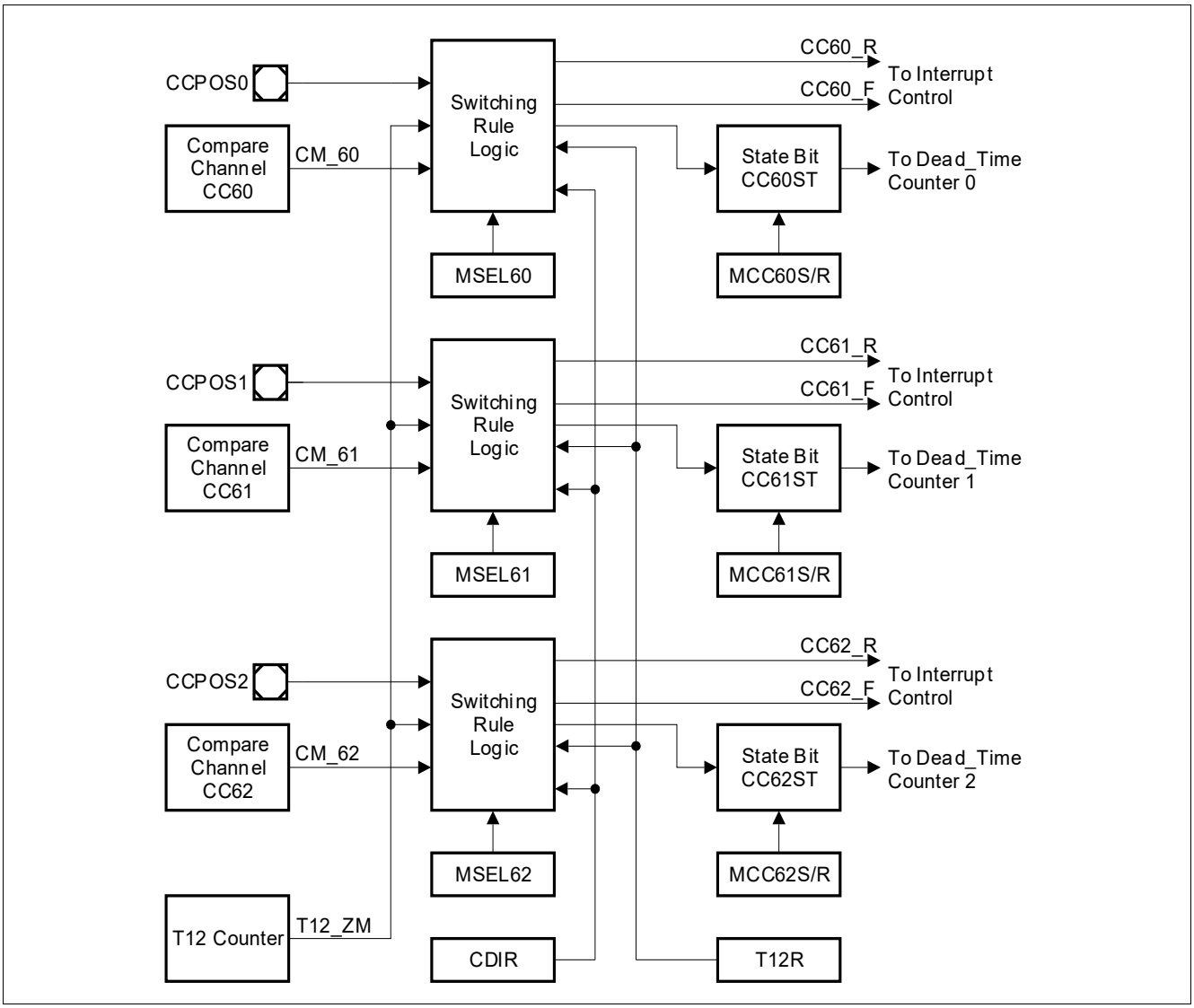


Figure 132 Compare State Bits for Compare Mode

The inputs to the switching rule logic for the CC6xST bits are the timer direction (CDIR), the timer run bit (T12R), the timer T12 zero-match signal (T12_ZM), and the actual individual compare-match signals CM_6x as well as the mode control bits, T12MSEL.MSEL6x.

In addition, each state bit can be set or cleared by software via the appropriate set and reset bits in register CMPMODIF, MCC6xS and MCC6xR. The input signals CCPOSx are used in hysteresis-like compare mode, whereas in normal compare mode, these inputs are ignored.

Note: In Hall Sensor, single shot or capture modes, additional/different rules are taken into account (see related sections).

A compare interrupt event CC6x_R is signaled when a compare match is detected while counting upwards, whereas the compare interrupt event CC6x_F is signaled when a compare match is detected while counting down. The actual setting of a State Bit has no influence on the interrupt generation in compare mode.

A modification of a State Bit CC6xST by the switching rule logic due to a compare action is only possible while Timer T12 is running (T12R = 1). If this is the case, the following switching rules apply for setting and clearing the State Bits in Compare Mode (illustrated in **Figure 133** and **Figure 134**):

A State Bit **CC6xST is set** to 1:

- with the next T12 clock (f_{T12}) after a compare-match when T12 is counting up (i.e., when the counter is incremented above the compare value);
- with the next T12 clock (f_{T12}) after a zero-match AND a parallel compare-match when T12 is counting up.

A State Bit **CC6xST is cleared** to 0:

- with the next T12 clock (f_{T12}) after a compare-match when T12 is counting down (i.e., when the counter is decremented below the compare value in center-aligned mode);
- with the next T12 clock (f_{T12}) after a zero-match AND NO parallel compare-match when T12 is counting up.

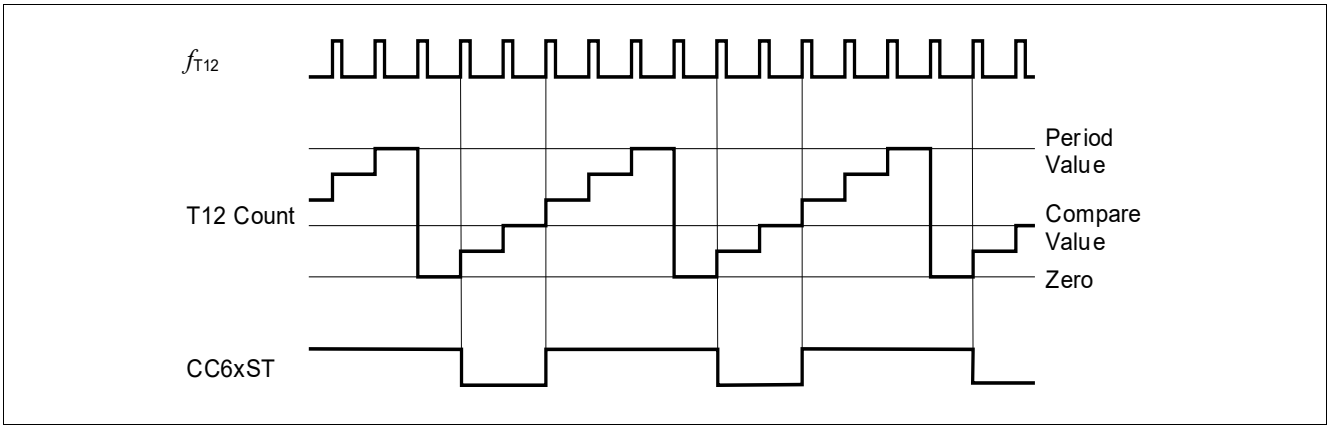


Figure 133 Compare Operation, Edge-Aligned Mode

Figure 135 illustrates some more examples for compare waveforms. It is important to note that in these examples, it is assumed that some of the compare values are changed while the timer is running. This change is performed via a software preload of the Shadow Register, CC6xSR. The value is transferred to the actual Compare Register CC6xR with the T12 Shadow Transfer signal, T12_ST, that is assumed to be enabled.

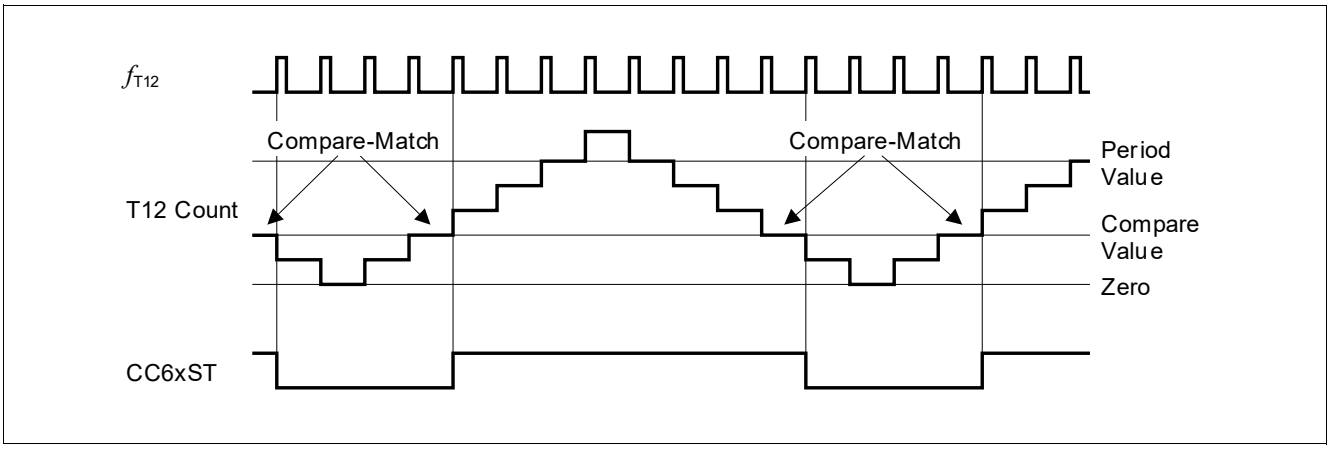


Figure 134 Compare Operation, Center-Aligned Mode

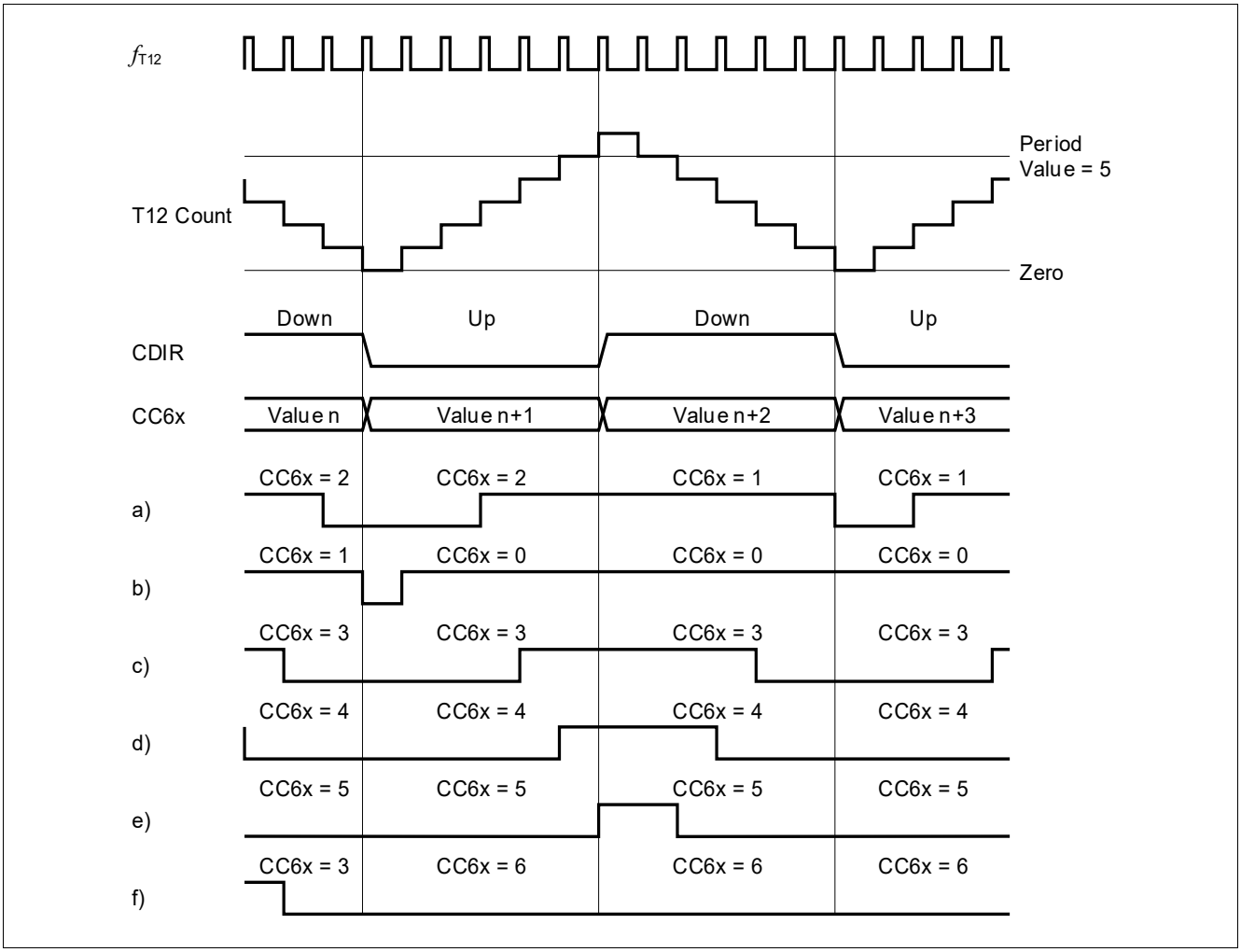


Figure 135 Compare Waveform Examples

Example b) illustrates the transition to a duty cycle of 100%. First, a compare value of 0001_H is used, then changed to 0000_H. Please note that a low pulse with the length of one T12 clock is still produced in the cycle where the new value 0000_H is in effect; this pulse originates from the previous value 0001_H. In the following timer cycles, the State Bit CC6xST remains at 1, producing a 100% duty cycle signal. In this case, the compare rule ‘zero-match AND compare-match’ is in effect.

Example f) shows the transition to a duty cycle of 0%. The new compare value is set to <Period-Value> + 1, and the State Bit CC6ST remains cleared.

Figure 136 illustrates an example for the waveforms of all three channels. With the appropriate dead-time control and output modulation, a very efficient 3-phase PWM signal can be generated.

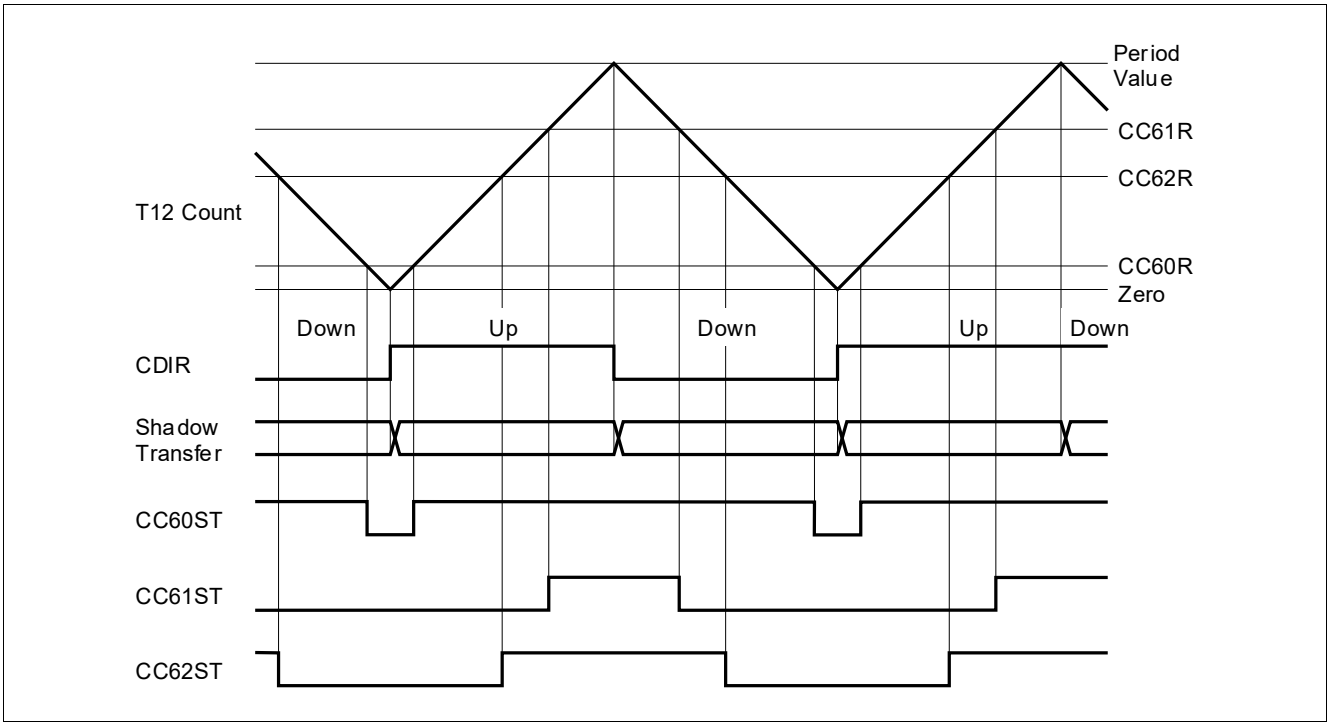


Figure 136 Three-Channel Compare Waveforms

18.3.3.3 Hysteresis-Like Control Mode

The hysteresis-like control mode (T12MSEL.MSEL6x = 1001_B) offers the possibility to switch off the PWM output if the input CCPOSx becomes 0 by clearing the State Bit CC6xST. This can be used as a simple motor control feature by using a comparator indicating, e.g., overcurrent. While CCPOSx = 0, the PWM outputs of the corresponding channel are driving their passive levels, because the setting of bit CC6xST is only possible while CCPOSx = 1.

As long as input CCPOSx is 0, the corresponding State Bit is held 0. When CCPOSx is at high level, the outputs can be in active state and are determined by bit CC6xST (see **Figure 132** for the state bit logic and **Figure 137** for the output paths). The CCPOSx inputs are evaluated with f_{CC6}.

This mode can be used to introduce a timing-related behavior to a hysteresis controller. A standard hysteresis controller detects if a value exceeds a limit and switches its output according to the compare result. Depending on the operating conditions, the switching frequency and the duty cycle are not fixed, but change permanently.

If (outer) time-related control loops based on a hysteresis controller in an inner loop should be implemented, the outer loops show a better behavior if they are synchronized to the inner loops. Therefore, the hysteresis-like mode can be used, that combines timer-related switching with a hysteresis controller behavior. For example, in this mode, an output can be switched on according to a fixed time base, but it is switched off as soon as a falling edge is detected at input CCPOSx.

This mode can also be used for standard PWM with overcurrent protection. As long as there is no low level signal at pin CCPOSx, the output signals are generated in the normal manner as described in the previous sections. Only if input CCPOSx shows a low level, e.g. due to the detection of overcurrent, the outputs are shut off to avoid harmful stress to the system.

18.3.4 Compare Mode Output Path

Figure 137 gives an overview on the signal path from a channel State Bit to its output pin in its simplest form. As illustrated, a user has a variety of controls to determine the desired output signal switching behavior in relation to the current state of the State Bit, CC6xST. Please refer to **Section 18.3.4.3** for details on the output modulation.

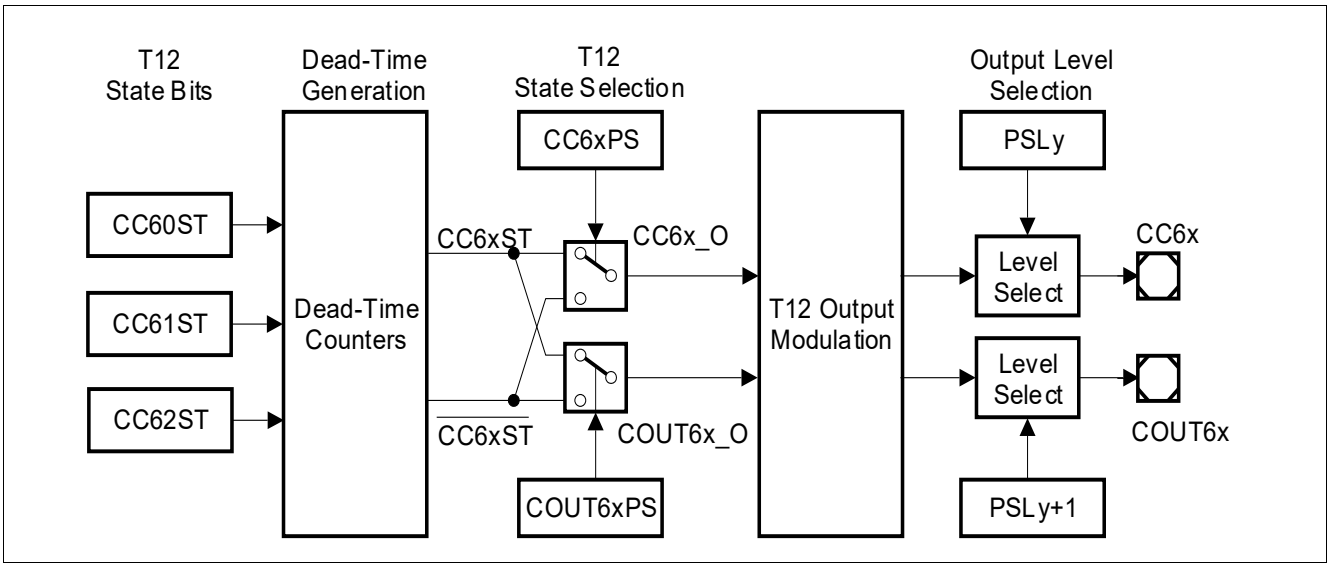


Figure 137 Compare Mode Simplified Output Path Diagram

The output path is based on signals that are defined as active or passive. The terms active and passive are not related to output levels, but to internal actions. This mainly applies for the modulation, where T12 and T13 signals are combined with the multi-channel signals and the trap function. The Output level Selection allows the user to define the output level at the output pin for the passive state (inverted level for the active state). It is recommended to configure this block in a way that an external power switch is switched off while the CCU6 delivers an output signal in the passive state.

18.3.4.1 Dead-Time Generation

The generation of (complementary) signals for the high-side and the low-side switches of one power inverter phase is based on the same compare channel. For example, if the high-side switch should be active while the T12 counter value is above the compare value (State Bit = 1), then the low-side switch should be active while the counter value is below the compare value (State Bit = 0).

In most cases, the switching behavior of the connected power switches is not symmetrical concerning the switch-on and switch-off times. A general problem arises if the time for switch-on is smaller than the time for switch-off of the power device. In this case, a short-circuit can occur in the inverter bridge leg, which may damage the complete system. In order to solve this problem by HW, this capture/compare unit contains a programmable Dead-Time Generation Block, that delays the passive to active edge of the switching signals by a programmable time (the active to passive edge is not delayed).

The Dead-Time Generation Block, illustrated in **Figure 138**, is built in a similar way for all three channels of T12. It is controlled by bits in register T12DTC. Any change of a CC6xST State Bit activates the corresponding Dead-Time Counter, that is clocked with the same input clock as T12 (f_{T12}). The length of the dead-time can be programmed by bit field DTM. This value is identical for all three channels. Writing TCTR4.DTRES = 1 sets all dead-times to passive.

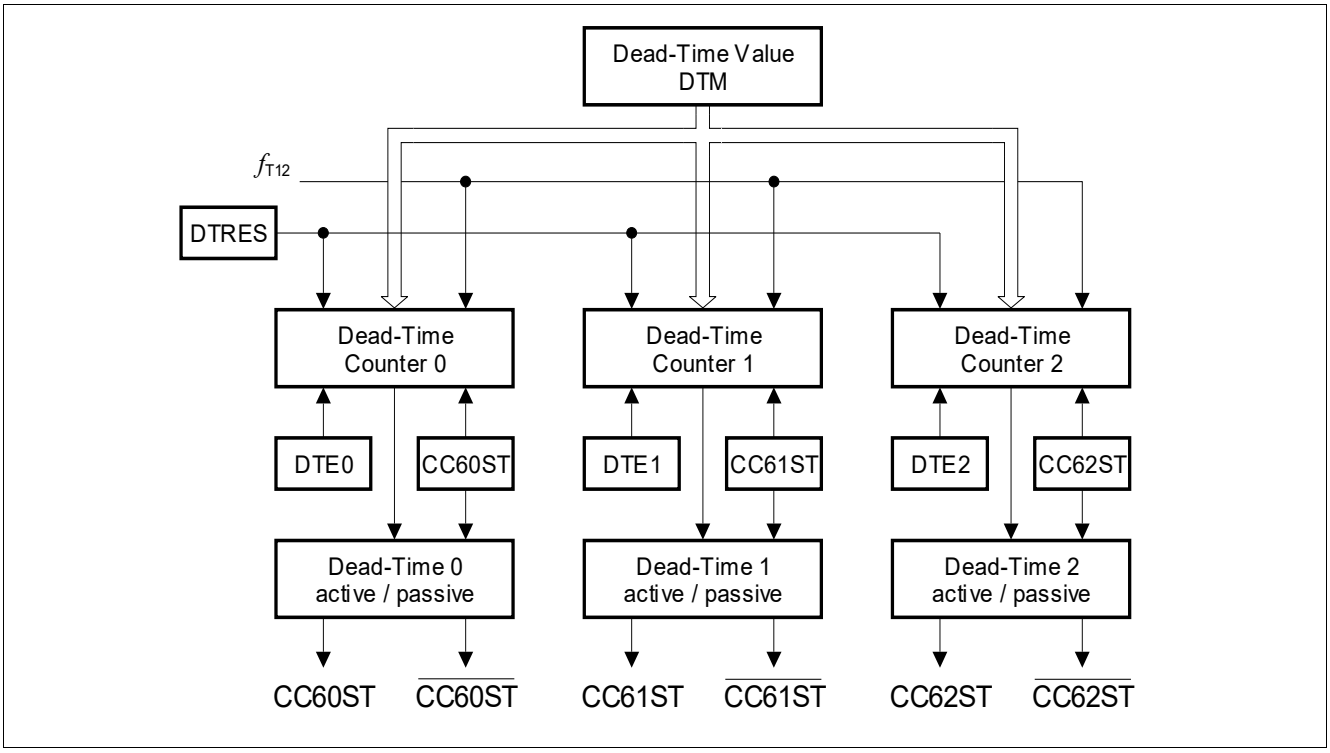


Figure 138 Dead-Time Generation Block Diagram

Each of the three dead-time counters has its individual dead-time enable bit, DTEx. An enabled dead-time counter generates a dead-time delaying the passive-to-active edge of the channel output signal. The change in a State Bit CC6xST is not taken into account while the dead-time generation of this channel is currently in progress (active). This avoids an unintentional additional dead-time if a State Bit CC6xST changes too early.

A disabled dead-time counter is always considered as passive and does not delay any edge of CC6xST.

Based on the State Bits CC6xST, the Dead-Time Generation Block outputs a direct signal CC6xST and an inverted signal $\overline{CC6xST}$ for each compare channel, each masked with the effect of the related Dead-Time Counters (waveforms illustrated in **Figure 139**).

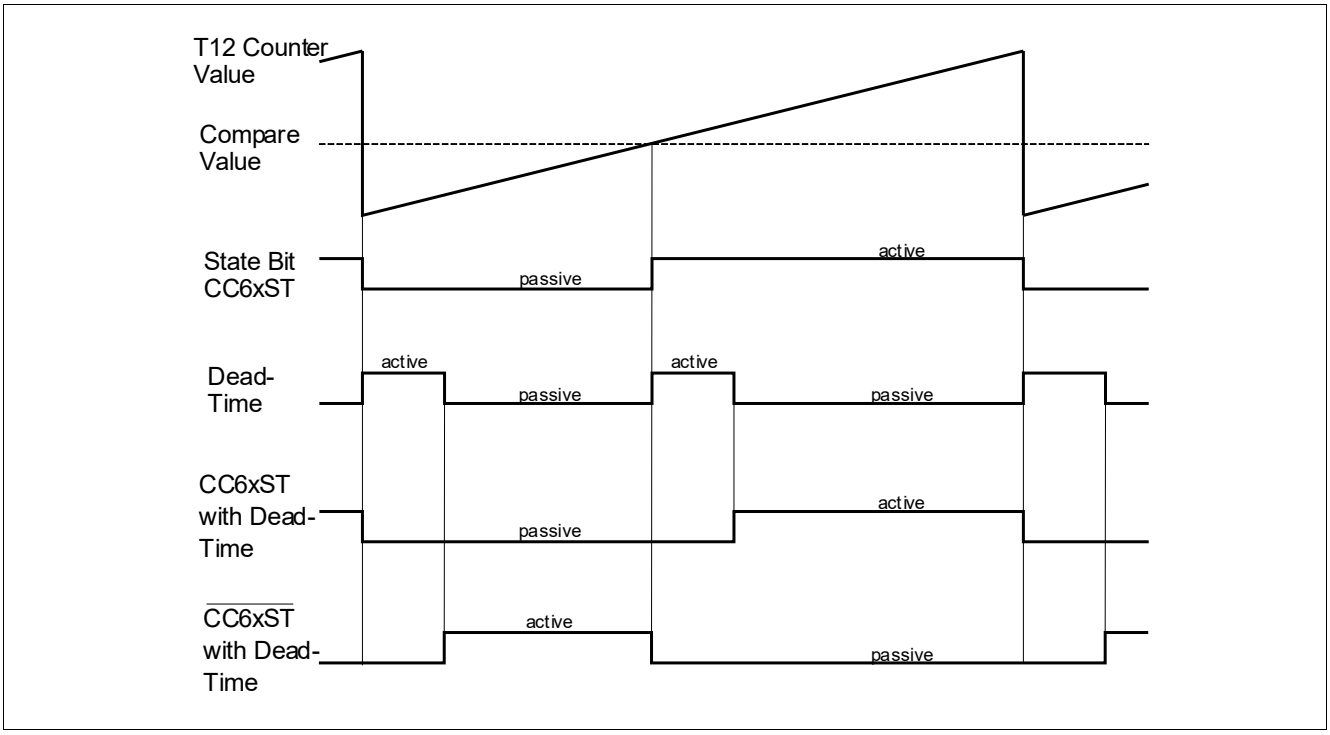


Figure 139 Dead-Time Generation Waveforms

18.3.4.2 State Selection

To support a wide range of power switches and drivers, the state selection offers the flexibility to define when a an output can be active and can be modulated, especially useful for **complementary or multi-phase PWM** signals.

The state selection is based on the signals CC6xST and $\overline{\text{CC6xST}}$ delivered by the dead-time generator (see **Figure 137**). Both signals are never active at the same time, but can be passive at the same time. This happens during the dead-time of each compare channel after a change of the corresponding State Bit CC6xST.

The user can select independently for each output signal CC6xO and COUT6xO if it should be active before or after the compare value has been reached (see register CMPSTAT). With this selection, the active (conducting) phases of complementary power switches in a power inverter bridge leg can be positioned with respect to the compare value (e.g. signal CC6xO can be active before, whereas COUT6xO can be active after the compare value is reached). Like this, the output modulation, the trap logic and the output level selection can be programmed independently for each output signal, although two output signals are referring to the same compare channel.

18.3.4.3 Output Modulation and Level Selection

The last block of the data path is the Output Modulation block. Here, all the modulation sources and the trap functionality are combined and control the actual level of the output pins (controlled by the modulation enable bits T1xMODENy and MCMEN in register MODCTR). The following signal sources can be combined here **for each T12 output signal** (see **Figure 140** for compare channel CC60):

- A **T12 related compare signal** CC6x_O (for outputs CC6x) or COUT6x_O (for outputs COUT6x) delivered by the T12 block (state selection with dead-time) with an individual enable bit T12MODENy per output signal (y = 0, 2, 4 for outputs CC6x and y = 1, 3, 5 for outputs COUT6x)
- The **T13 related compare signal** CC63_O delivered by the T13 state selection with an individual enable bit T13MODENy per output signal (y = 0, 2, 4 for outputs CC6x and y = 1, 3, 5 for outputs COUT6x)
- A **multi-channel output signal** MCMPy (y = 0, 2, 4 for outputs CC6x and y = 1, 3, 5 for outputs COUT6x) with a common enable bit MCMEN
- The **trap state** TRPS with an individual enable bit TRPENy per output signal (y = 0, 2, 4 for outputs CC6x and y = 1, 3, 5 for outputs COUT6x)

If one of the modulation input signals CC6x_O/COUT6x_O, CC63_O, or MCMPy of an output modulation block is enabled and is at passive state, the modulated is also in passive state, regardless of the state of the other signals that are enabled. Only if all enabled signals are in active state the modulated output shows an active state. If no modulation input is enabled, the output is in passive state.

If the Trap State is active (TRPS = 1), then the outputs that are enabled for the trap signal (by TRPENy = 1) are set to the passive state.

The output of each of the modulation control blocks is connected to a level select block that is configured by register PSLR. It offers the option to determine the actual output level of a pin, depending on the state of the output line (decoupling of active/passive state and output polarity) as specified by the Passive State Select bit PSLy. If the modulated output signal is in the passive state, the level specified directly by PSLy is output. If it is in the active state, the inverted level of PSLy is output. This allows the user to adapt the polarity of an active output signal to the connected circuitry.

The PSLy bits have shadow registers to allow for updates without undesired pulses on the output lines. The bits related to CC6x and COUT6x (x = 0, 1, 2) are updated with the T12 shadow transfer signal (T12_ST). A read action returns the actually used values, whereas a write action targets the shadow bits. Providing a shadow register for the PSL value as well as for other values related to the generation of the PWM signal facilitates a concurrent update by software for all relevant parameters.

Figure 140 shows the output modulation structure for compare channel CC60 (output signals CC60 and COUT60). A similar structure is implemented for the other two compare channels CC61 and CC62.

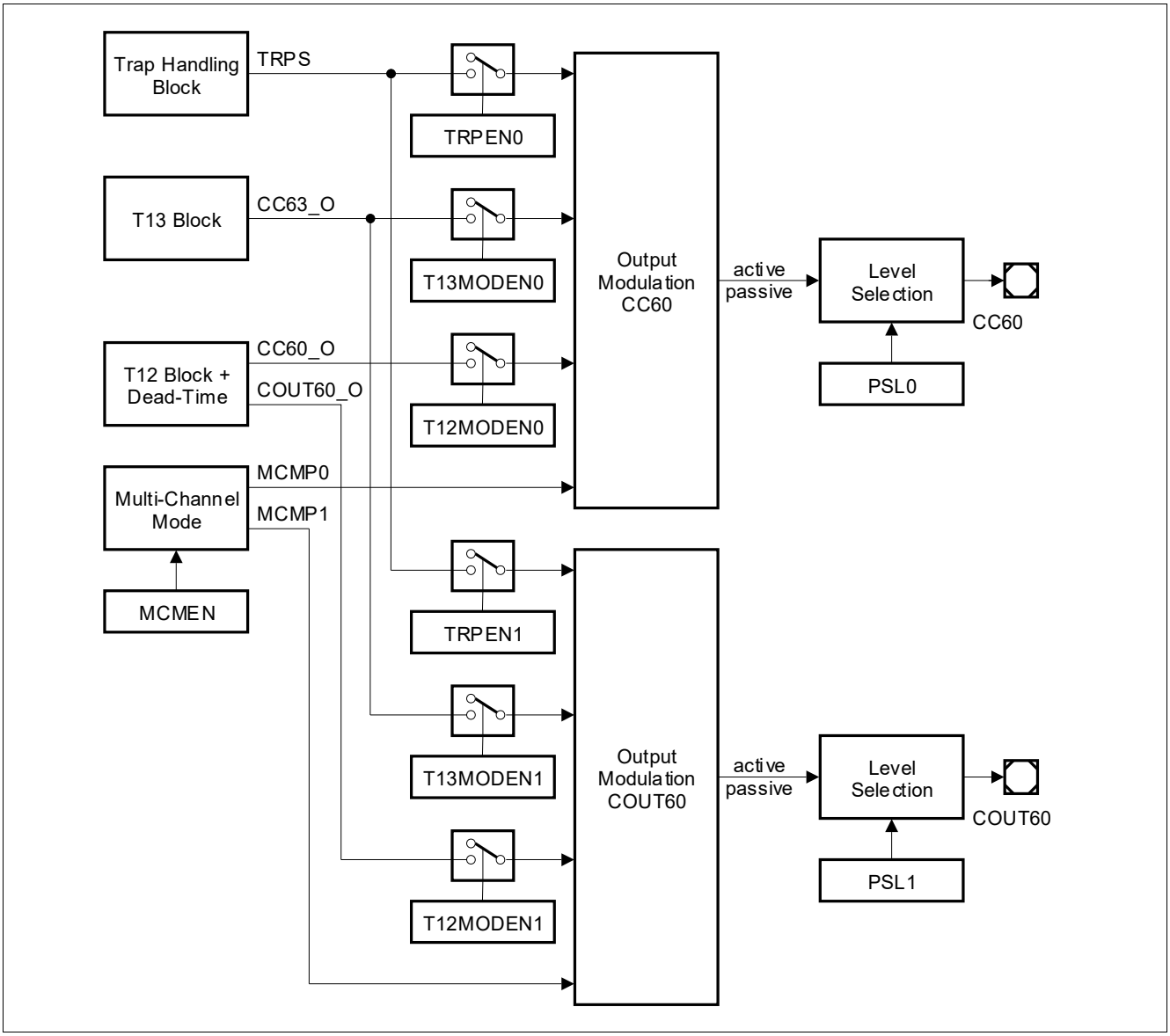


Figure 140 Output Modulation for Compare Channel CC60

18.3.5 T12 Capture Modes

Each of the three channels of the T12 Block can also be used to capture T12 time information in response to an external signal CC6xIN.

In capture mode, the interrupt event CC6x_R is detected when a rising edge is detected at the input CC6xIN, whereas the interrupt event CC6x_F is detected when a falling edge is detected.

There are a number of different modes for capture operation. In all modes, both of the registers of a channel are used. The selection of the capture modes is done via the T12MSEL.MSEL6x bit fields and can be selected individually for each of the channels.

Table 119 Capture Modes Overview

MSEL6x	Mode	Signal	Active Edge	CC6nSR Stored in	T12 Stored in
0100_B	1	CC6xIN	Rising	–	CC6xR
		CC6xIN	Falling	–	CC6xSR
0101_B	2	CC6xIN	Rising	CC6xR	CC6xSR
0110_B	3	CC6xIN	Falling	CC6xR	CC6xSR
0111_B	4	CC6xIN	Any	CC6xR	CC6xSR

Figure 141 illustrates **Capture Mode 1**. When a rising edge (0-to-1 transition) is detected at the corresponding input signal CC6xIN, the current contents of Timer T12 are captured into register CC6xR. When a falling edge (1-to-0 transition) is detected at the input signal CC6xIN, the contents of Timer T12 are captured into register CC6xSR.

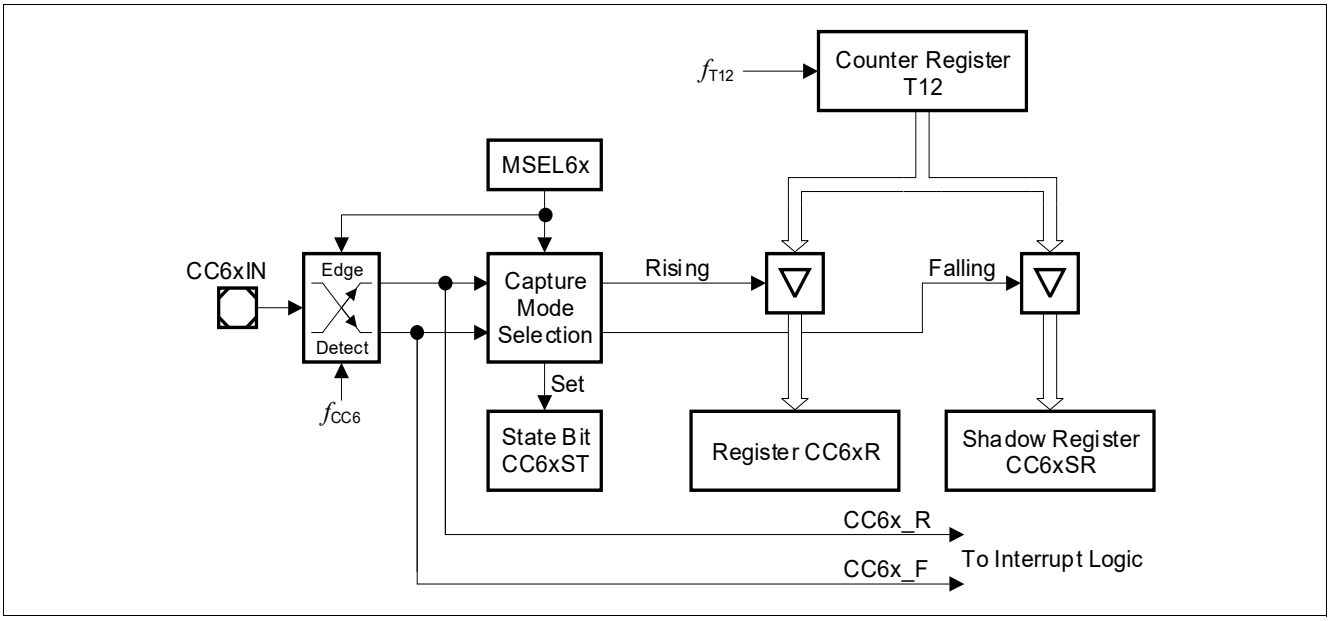


Figure 141 Capture Mode 1 Block Diagram

Capture Modes 2, 3 and 4 are shown in **Figure 142**. They differ only in the active edge causing the capture operation. In each of the three modes, when the selected edge is detected at the corresponding input signal CC6xIN, the current contents of the shadow register CC6xSR are transferred into register CC6xR, and the current Timer T12 contents are captured in register CC6xSR (simultaneous transfer). The active edge is a rising edge of CC6xIN for Capture Mode 2, a falling edge for Mode 3, and both, a rising or a falling edge for Capture Mode 4, as shown in **Table 119**. These capture modes are very useful in cases where there is little time between two consecutive edges of the input signal.

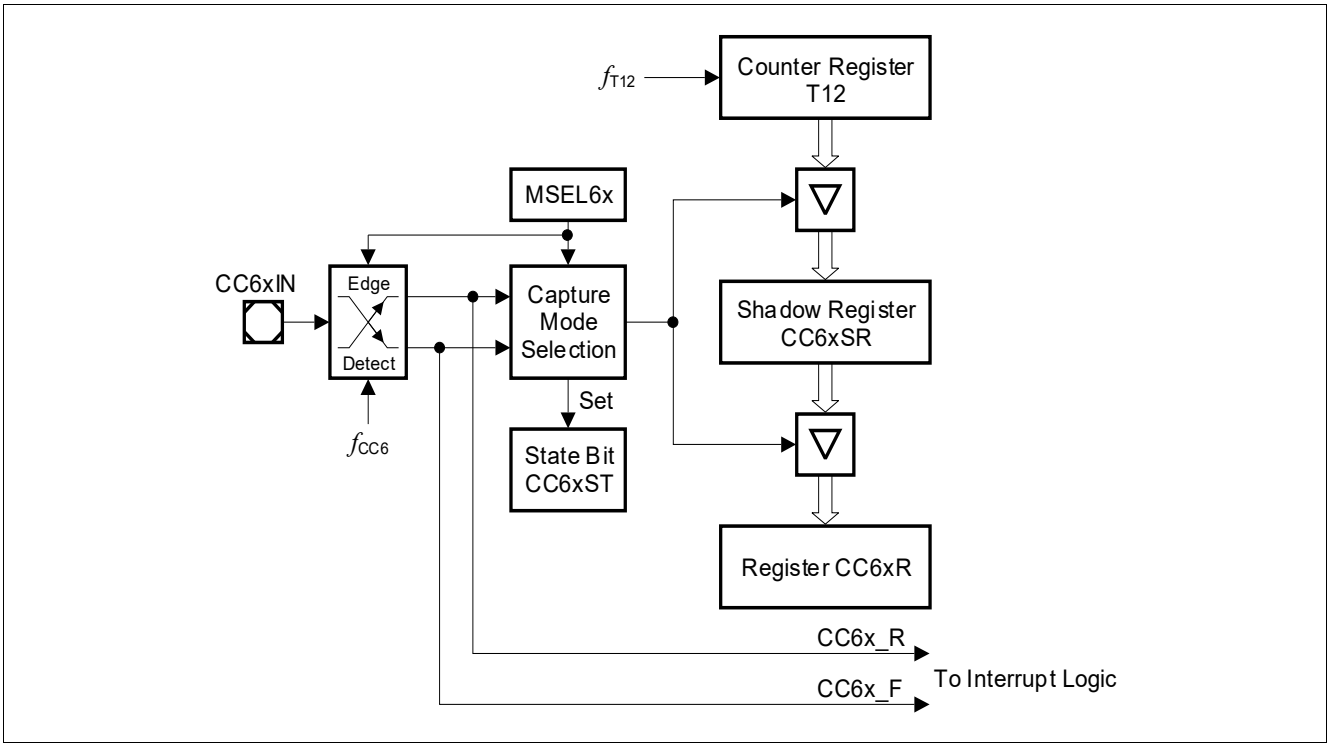


Figure 142 Capture Modes 2, 3 and 4 Block Diagram

Five further capture modes are called **Multi-Input Capture Modes**, as they use two different external inputs, signal CC6xIN and signal CCPOSx.

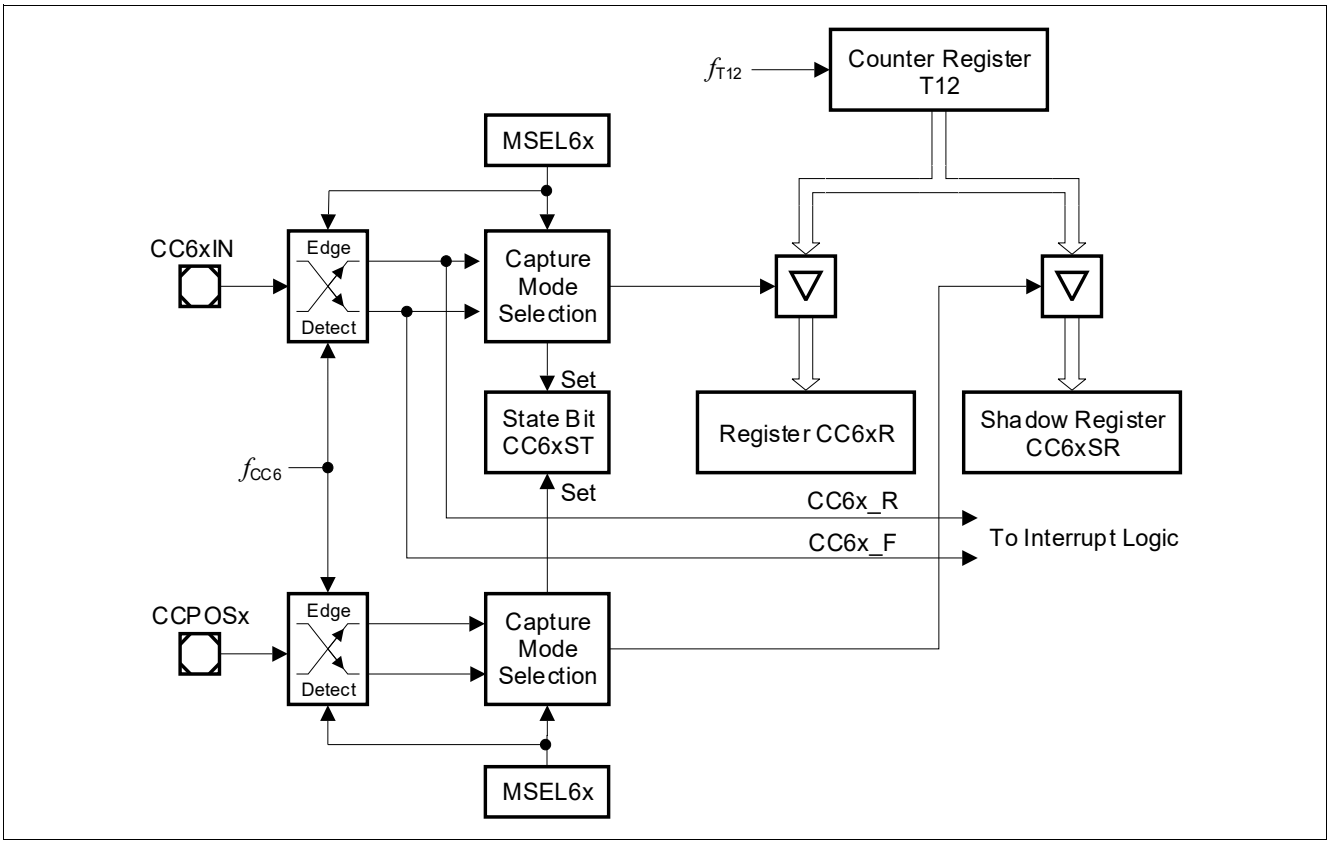


Figure 143 Multi-Input Capture Modes Block Diagram

In each of these modes, the current T12 contents are captured in register CC6xR in response to a selected event at signal CC6xIN, and in register CC6xSR in response to a selected event at signal CCPOSx. The possible events can be opposite input transitions, or the same transitions, or any transition at the two inputs. The different options are detailed in **Table 120**.

In each of the various capture modes, the Channel State Bit, CC6xST, is set to 1 when the selected capture trigger event at signal CC6xIN or CCPOSx has occurred. The State Bit is not cleared by hardware, but can be cleared by software.

In addition, appropriate signal lines to the interrupt logic are activated, that can generate an interrupt request to the CPU. Regardless of the selected active edge, all edges detected at signal CC6xIN can lead to the activation of the appropriate interrupt request line (see also **Section 18.10**).

Table 120 Multi-Input Capture Modes Overview

MSEL6x	Mode	Signal	Active Edge	T12 Stored in
1010_B	5	CC6xIN	Rising	CC6xR
		CCPOSx	Falling	CC6xSR
1011_B	6	CC6xIN	Falling	CC6xR
		CCPOSx	Rising	CC6xSR
1100_B	7	CC6xIN	Rising	CC6xR
		CCPOSx	Rising	CC6xSR
1101_B	8	CC6xIN	Falling	CC6xR
		CCPOSx	Falling	CC6xSR
1110_B	9	CC6xIN	Any	CC6xR
		CCPOSx	Any	CC6xSR
1111_B	–	reserved (no capture or compare action)		

18.3.6 T12 Shadow Register Transfer

A special shadow transfer signal (T12_ST) can be generated to facilitate updating the period and compare values of the compare channels CC60, CC61, and CC62 synchronously to the operation of T12. Providing a shadow register for values defining one PWM period facilitates a concurrent update by software for all relevant parameters. The next PWM period can run with a new set of parameters. The generation of this signal is requested by software via bit TCTR0.STE12 (set by writing 1 to the write-only bit TCTR4.T12STR, cleared by writing 1 to the write-only bit TCTR4.T12STD).

Figure 144 shows the shadow register structure and the shadow transfer signals, as well as on the read/write accessibility of the various registers.

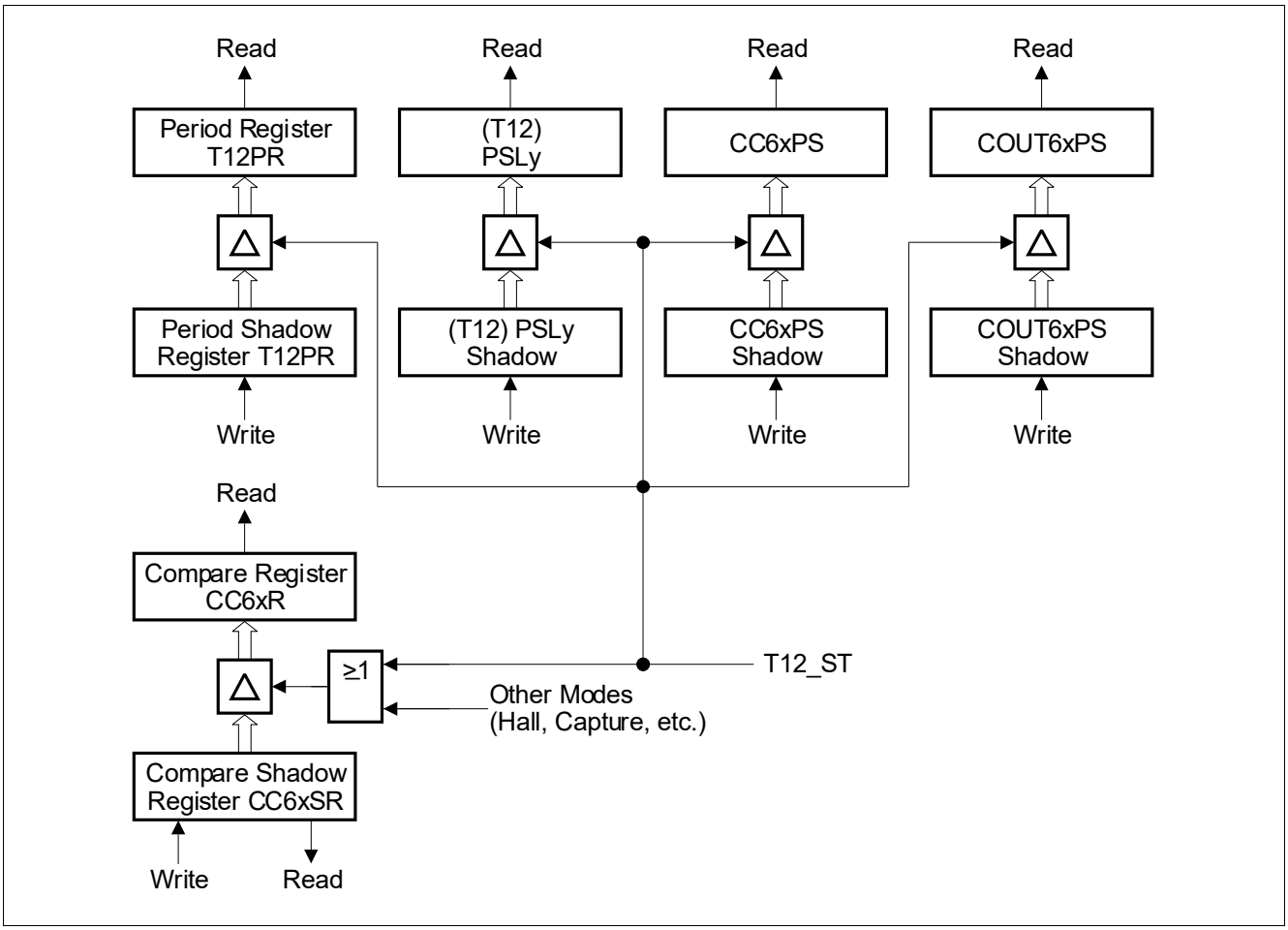


Figure 144 T12 Shadow Register Overview

A T12 shadow register transfer takes place (T12_ST active):

- STE12 = 1 and a Period-Match is detected while counting up, or
- STE12 = 1 and a One-Match is detected while counting down

When signal T12_ST is active, a shadow register transfer is triggered with the next cycle of the T12 clock. Bit STE12 is automatically cleared with the shadow register transfer.

18.3.7 Timer T12 Operating Mode Selection

The operating mode for the T12 channels are defined by the bit fields T12MSEL.MSEL6x.

Table 121 T12 Capture/Compare Modes Overview

MSEL6x	Selected Operating Mode
0000_B, 1111_B	Capture/Compare modes switched off
0001_B, 0010_B, 0011_B	Compare mode, see **Section 18.3.3** same behavior for all three codings
$01XX_B$	Double-Register Capture modes, see **Section 18.3.5**
1000_B	Hall Sensor Mode, see **Section 18.7** In order to properly enable this mode, all three MSEL6x fields have to be programmed to Hall Sensor mode.
1001_B	Hysteresis-like compare mode, see **Section 18.3.3.3**
1010_B, 1011_B, 1100_B, 1101_B, 1110_B	Multi-Input Capture modes, see **Section 18.3.5**

The clocking and counting scheme of the timers are controlled by the timer control registers TCTR0 and TCTR2. Specific actions are triggered by write operations to register TCTR4.

18.4 Operating Timer T13

Timer T13 is implemented similarly to Timer T12, but only with one channel in compare mode. A 16-bit up-counter is connected to a channel register via a comparator, that generates a signal when the counter contents match the contents of the channel register. A variety of control functions facilitate the adaptation of the T13 structure to different application needs. In addition, T13 can be started synchronously to timer T12 events.

This section provides information about:

- T13 overview (see **Section 18.4.1**)
- Counting scheme (see **Section 18.4.2**)
- Compare mode (see **Section 18.4.3**)
- Compare output path (see **Section 18.4.4**)
- Shadow register transfer (see **Section 18.4.5**)

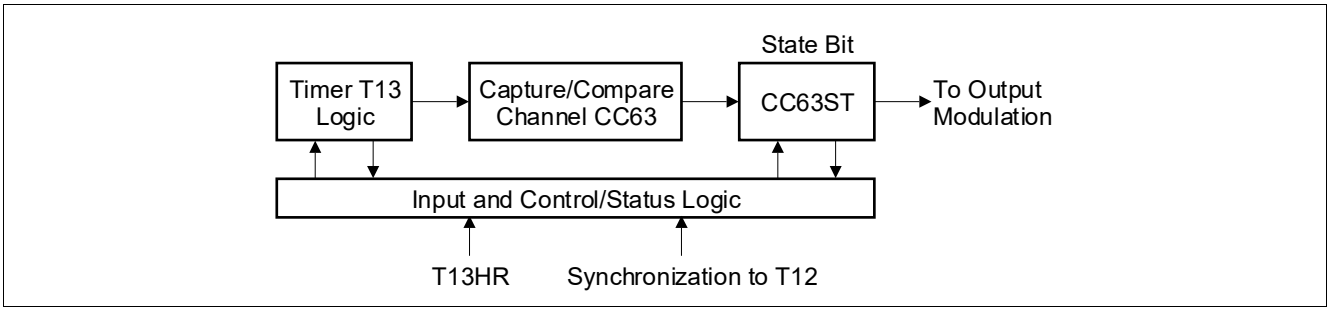


Figure 145 Overview Diagram of the Timer T13 Block

18.4.1 T13 Overview

Figure 146 shows a detailed block diagram of Timer T13. The functions of the timer T13 block are controlled by bits in registers TCTR0, TCTR2, and PISEL2.

Timer T13 receives its input clock, f_{T13}, from the module clock f_{CC6} via a programmable prescaler and an optional 1/256 divider or from an input signal T13HR. T13 can only count up (similar to the Edge-Aligned mode of T13).

Via a comparator, the timer T13 Counter Register T13 is connected to the Period Register T13PR. This register determines the maximum count value for T13. When T13 reaches the period value, signal T13_PM (T13 Period Match) is generated and T13 is cleared to 0000_H with the next T13 clock edge. The Period Register receives a new period value from its Shadow Period Register, T13PS, that is loaded via software. The transfer of a new period value from the shadow register into T13PR is controlled via the 'T13 Shadow Transfer' control signal, T13_ST. The generation of this signal depends on the associated control bit STE13. Providing a shadow register for the period value as well as for other values related to the generation of the PWM signal facilitates a concurrent update by software for all relevant parameters (refer to **Table 18.4.5**). Another signal indicates whether the counter contents are equal to 0000_H (T13_ZM).

A Single-Shot control bit, T13SSC, enables an automatic stop of the timer when the current counting period is finished (see **Figure 148**).

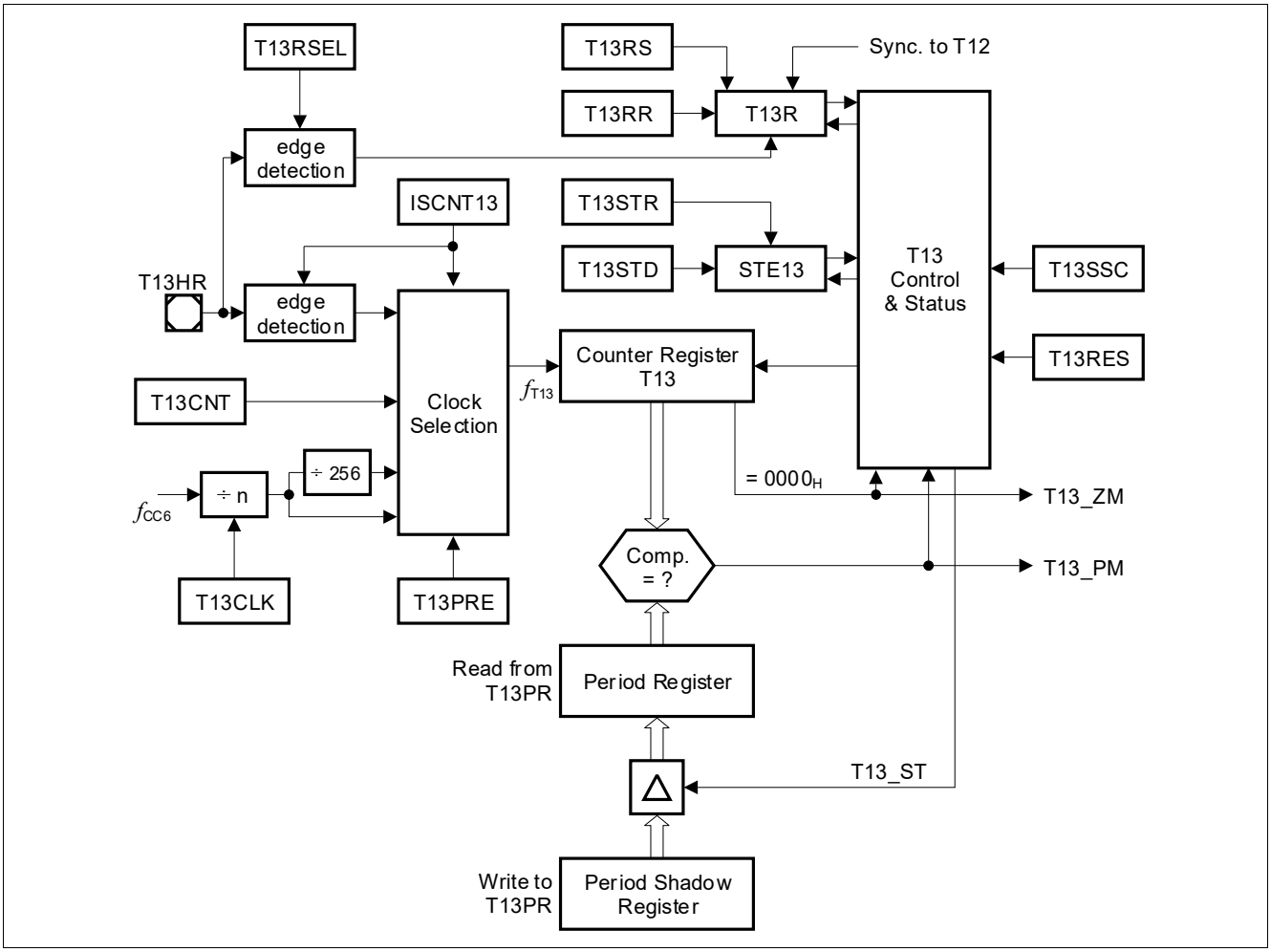


Figure 146 T13 Counter Logic and Period Comparators

The start or stop of T13 is controlled by the Run bit, T13R. This control bit can be set by software via the associated set/clear bits T13RS or T13RR in register TCTR4, or it is cleared by hardware according to preselected conditions (single-shot mode).

The timer T13 run bit T13R must not be set while the applied T13 period value is zero. Bit T13R can be set automatically if an event of T12 is detected to synchronize T13 timings to T12 events, e.g. to generate a programmable delay via T13 after an edge of a T12 compare channel before triggering an AD conversion (T13 can trigger ADC conversions).

Timer T13 can be cleared to 0000_H via control bit T13RES. Setting this write-only bit only clears the timer contents, but has no further effects, e.g., it does not stop the timer.

The generation of the T13 shadow transfer control signal, T13_ST, is enabled via bit STE13. This bit can be set or cleared by software indirectly through its associated set/reset control bits T13STR and T13STD.

Two bit fields, T13TEC and T13TED, control the synchronization of T13 to Timer T12 events. T13TEC selects the trigger event, while T13TED determines for which T12 count direction the trigger should be active.

While Timer T13 is running, write accesses to the count register T13 are not taken into account. If T13 is stopped, write actions to register T13 are immediately taken into account.

Note: The T13 Period Register and its associated shadow register are located at the same physical address. A write access to this address targets the Shadow Register, while a read access reads from the actual period register.

18.4.2 T13 Counting Scheme

This section describes the clocking and the counting capabilities of T13.

18.4.2.1 Clock Selection

In **Timer Mode** (PISEL2. ISCNT13 = 00_B), the input clock f_{T13} of Timer T13 is derived from the internal module clock f_{CC6} through a programmable prescaler and an optional 1/256 divider. The resulting prescaler factors are listed in **Table 122**. The prescaler of T13 is cleared while T13 is not running (TCTR0.T13R = 0) to ensure reproducible timings and delays.

Table 122 Timer T13 Input Clock Options

T13CLK	Resulting Input Clock f_{T13} Prescaler Off (T13PRE = 0)	Resulting Input Clock f_{T13} Prescaler On (T13PRE = 1)
000_B	f_{CC6}	f_{CC6} / 256
001_B	f_{CC6} / 2	f_{CC6} / 512
010_B	f_{CC6} / 4	f_{CC6} / 1024
011_B	f_{CC6} / 8	f_{CC6} / 2048
100_B	f_{CC6} / 16	f_{CC6} / 4096
101_B	f_{CC6} / 32	f_{CC6} / 8192
110_B	f_{CC6} / 64	f_{CC6} / 16384
111_B	f_{CC6} / 128	f_{CC6} / 32768

In **Counter Mode**, timer T13 counts one step:

- If a 1 is written to TCTR4.T13CNT and PISEL2.ISCNT13 = 01_B
- If a rising edge of input signal T13HR is detected and PISEL2.ISCNT13 = 10_B
- If a falling edge of input signal T13HR is detected and PISEL2.ISCNT13 = 11_B

18.4.2.2 T13 Counting

The period of the timer is determined by the value in the period Register T13PR according to the following formula:

$T13_{PER}$ = <Period-Value> + 1; in T13 clocks (f_{T13}) (18.3)

Timer T13 can only count up, comparable to the Edge-Aligned mode of T12. This leads to very simple 'counting rule' for the T13 counter:

- The counter is cleared with the next T13 clock edge if a Period-Match is detected. The counting direction is always upwards.

The behavior of T13 is illustrated in **Figure 147**.

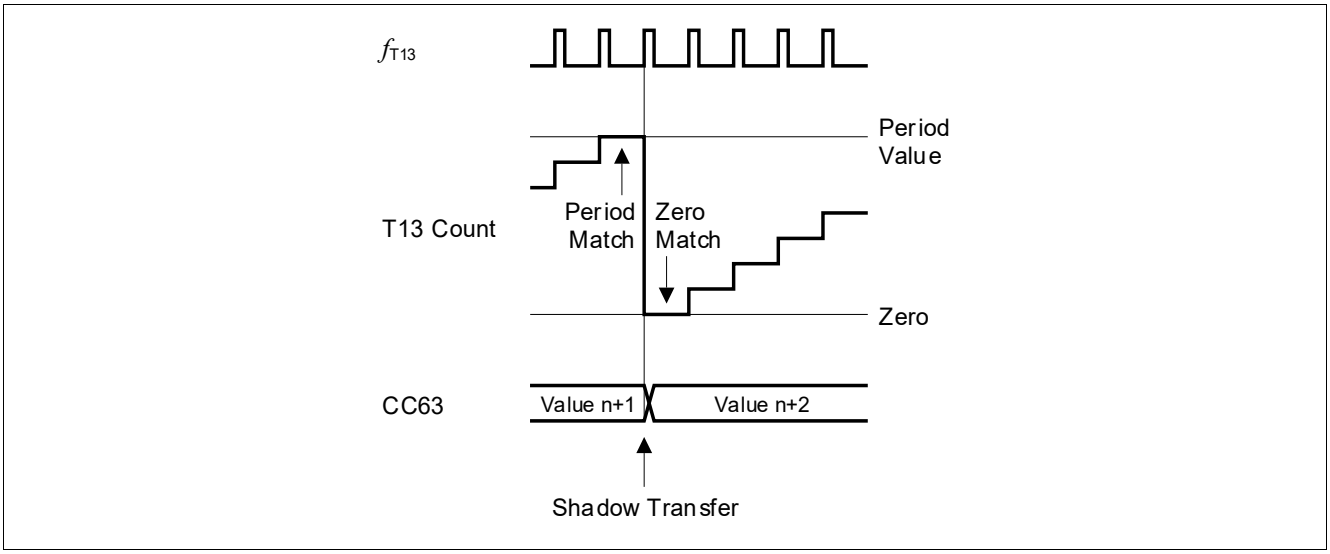


Figure 147 T13 Counting Sequence

18.4.2.3 Single-Shot Mode

In Single-Shot Mode, the timer run bit T13R is cleared by hardware. If bit T13SSC = 1, the timer T13 will stop when the current timer period is finished.

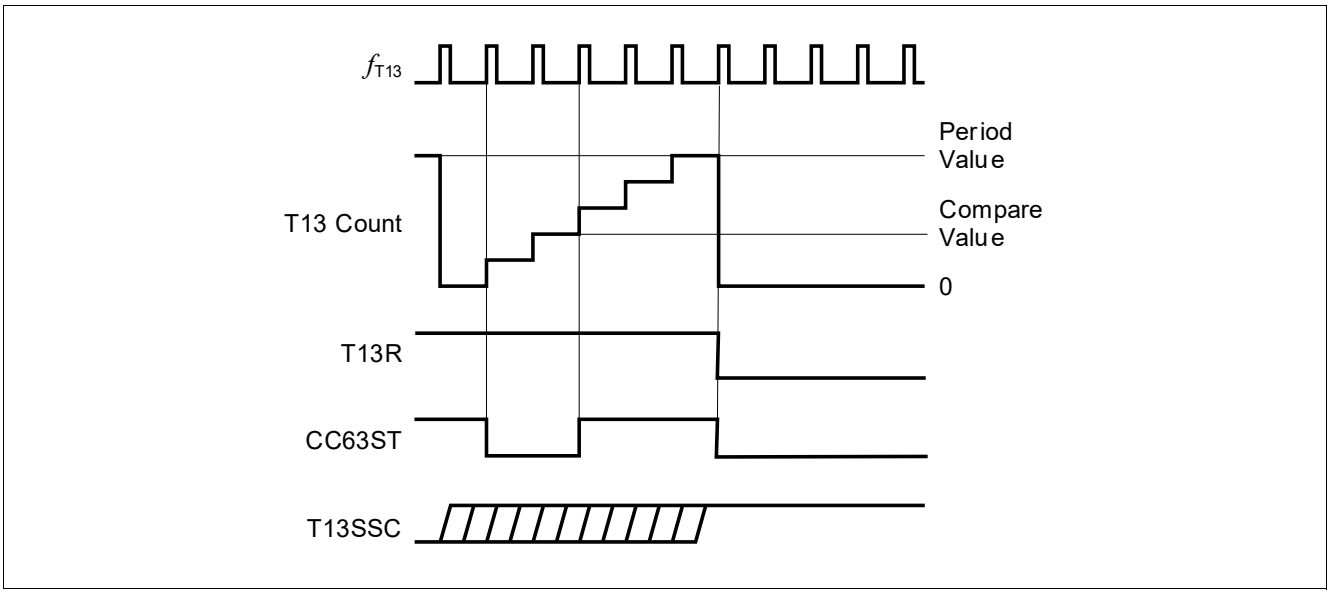


Figure 148 Single-Shot Operation of Timer T13

18.4.2.4 Synchronization to T12

Timer T13 can be synchronized to a T12 event. Bit fields T13TEC and T13TED select the event that is used to start Timer T13. The selected event sets bit T13R via HW, and T13 starts counting. Combined with the Single-Shot mode, this feature can be used to generate a programmable delay after a T12 event.

Figure 149 shows an example for the synchronization of T13 to a T12 event. Here, the selected event is a compare-match (compare value = 2) while counting up. The clocks of T12 and T13 can be different (other prescaler factor); the figure shows an example in which T13 is clocked with half the frequency of T12.

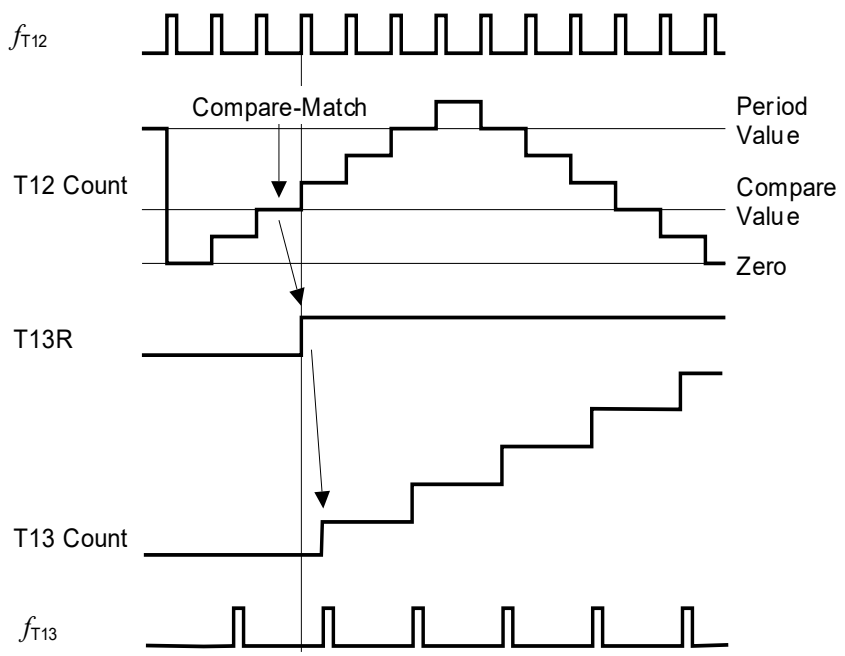


Figure 149 Synchronization of T13 to T12 Compare Match

Bit field T13TEC selects the trigger event to start T13 (automatic set of T13R for synchronization to T12 compare signals) according to the combinations shown in **Table 123**. Bit field T13TED additionally specifies for which count direction of T12 the selected trigger event should be regarded (see **Table 124**).

Table 123 T12 Trigger Event Selection

T13TEC	Selected Event
000_B	None
001_B	T12 Compare Event on Channel 0 (CM_CC60)
010_B	T12 Compare Event on Channel 1 (CM_CC61)
011_B	T12 Compare Event on Channel 2 (CM_CC62)
100_B	T12 Compare Event on any Channel (0, 1, 2)
101_B	T12 Period-Match (T12_PM)
110_B	T12 Zero-Match while counting up (T12_ZM and CDIR = 0)
111_B	Any Hall State Change

Table 124 T12 Trigger Event Additional Specifier

T13TED	Selected Event Specifier
00_B	Reserved, no action
01_B	Selected event is active while T12 is counting up (CDIR = 0)
10_B	Selected event is active while T12 is counting down (CDIR = 1)
11_B	Selected event is active independently of the count direction of T12

18.4.3 T13 Compare Mode

Associated with Timer T13 is one compare channel, that can perform compare operations with regard to the contents of the T13 counter.

Figure 145 gives an overview on the T13 channel in Compare Mode. The channel is connected to the T13 counter register via an equal-to comparator, generating a compare match signal when the contents of the counter matches the contents of the compare register.

The channel consists of the comparator and a double register structure - the actual compare register, CC63R, feeding the comparator, and an associated shadow register, CC63SR, that is preloaded by software and transferred into the compare register when signal T13 shadow transfer, T13_ST, gets active. Providing a shadow register for the compare value as well as for other values related to the generation of the PWM signal facilitates a concurrent update by software for all relevant parameters.

Associated with the channel is a State Bit, CMPSTAT.CC63ST, holding the status of the compare operation. **Figure 150** gives an overview on the logic for the State Bit.

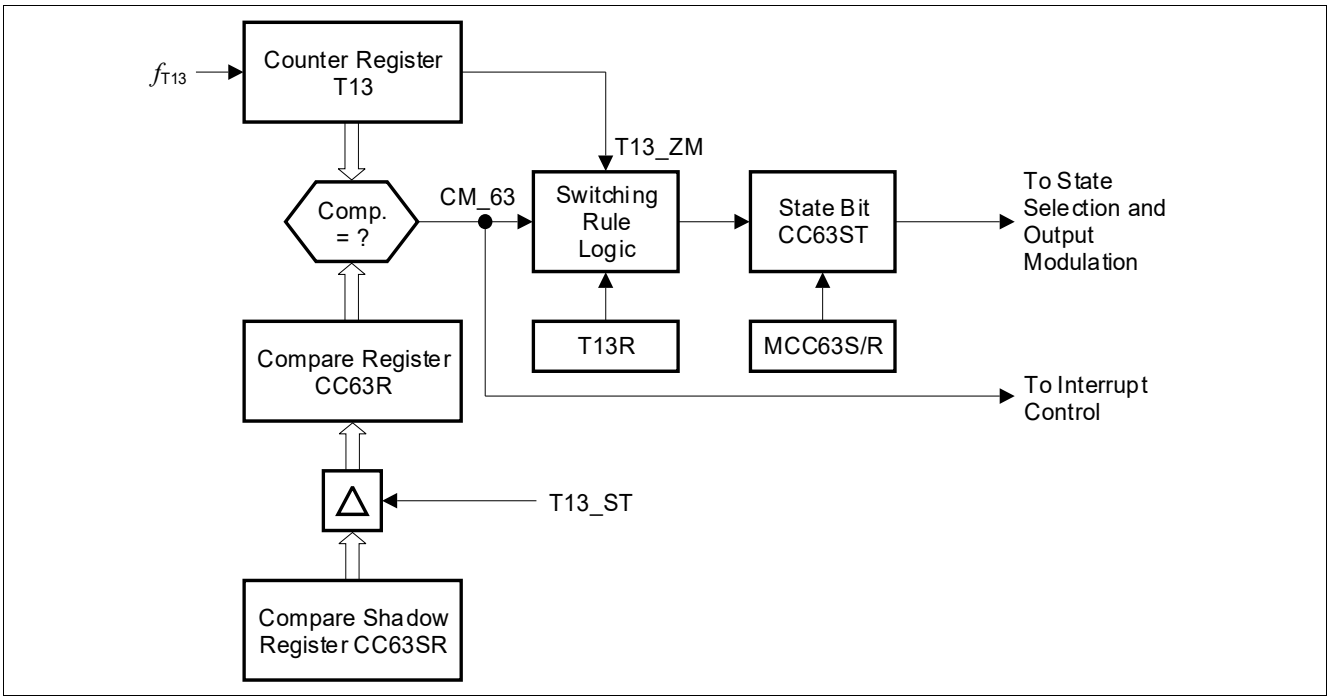


Figure 150 T13 State Bit Block Diagram

A compare interrupt event CM_63 is signaled when a compare match is detected. The actual setting of a State Bit has no influence on the interrupt generation.

The inputs to the switching rule logic for the CC63ST bit are the timer run bit (T13R), the timer zero-match signal (T13_ZM), and the actual individual compare-match signal CM_63. In addition, the state bit can be set or cleared by software via bits MCC63S and MCC63R in register CMPMODIF.

A modification of the State Bit CC63ST by hardware is only possible while Timer T13 is running (T13R = 1). If this is the case, the following switching rules apply for setting and resetting the State Bit in Compare Mode:

State Bit **CC63ST is set** to 1

- with the next T13 clock (f_{T13}) after a compare-match (T13 is always counting up) (i.e., when the counter is incremented above the compare value);
- with the next T13 clock (f_{T13}) after a zero-match AND a parallel compare-match.

State Bit **CC63ST is cleared** to 0

- with the next T13 clock (f_{T13}) after a zero-match AND NO parallel compare-match.

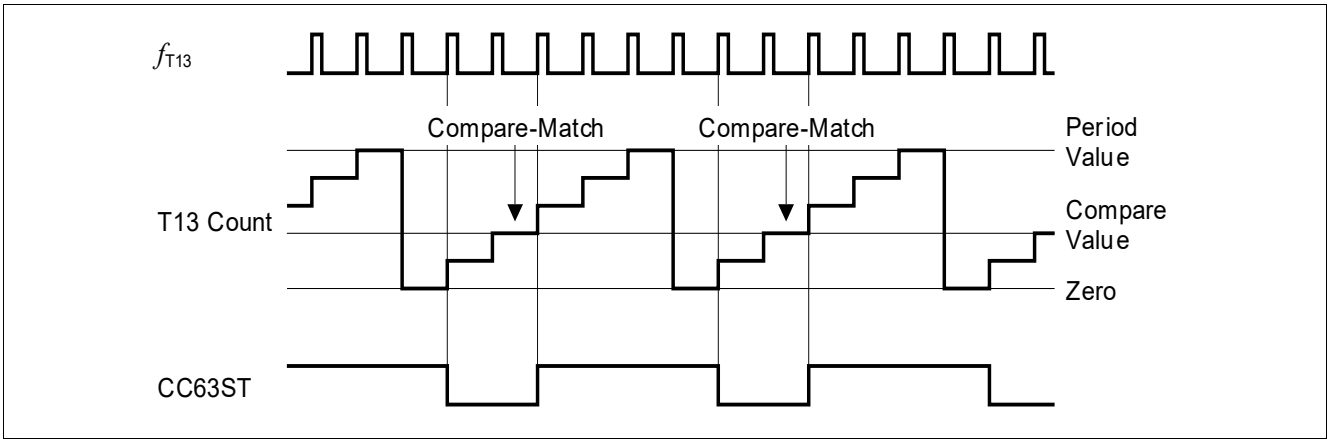


Figure 151 T13 Compare Operation

18.4.4 Compare Mode Output Path

Figure 152 gives an overview on the signal path from the channel State Bit CC63ST to its output pin COUT63. As illustrated, a user can determine the desired output behavior in relation to the current state of CC63ST. Please refer to **Section 18.3.4.3** for detailed information on the output modulation for T12 signals.

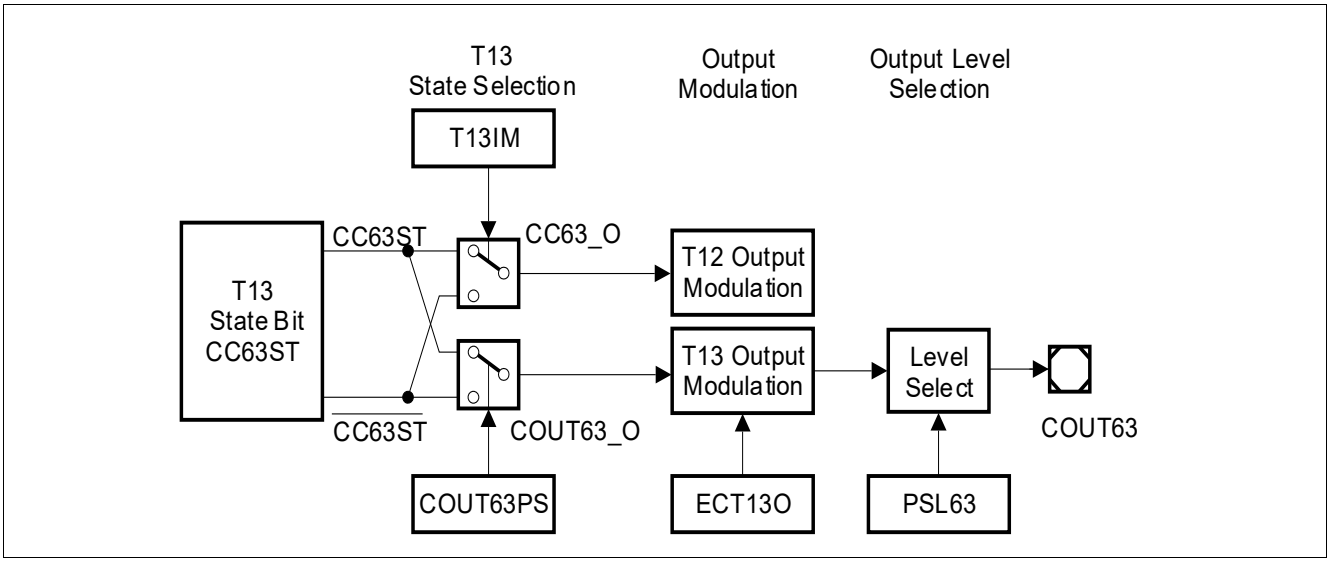


Figure 152 Channel 63 Output Path

The output line COUT63_O can generate a T13 PWM at the output pin COUT63. The signal CC63_O can be used to modulate the T12-related output signals with a T13 PWM. In order to decouple COUT63 from the internal

modulation, the compare state leading to an active signal can be selected independently by bits T13IM and COUT63PS.

The last block of the data path is the Output Modulation block. Here, the modulation source T13 and the trap functionality are combined and control the actual level of the output pin COUT63 (see **Figure 153**):

- The **T13 related compare signal** COUT63_O delivered by the T13 state selection with the enable bit MODCTR.ECT13O
- The **trap state** TRPS with an individual enable bit TRPCTR.TRPEN13

If the modulation input signal COUT63_O is enabled (ECT13O = 1) and is at passive state, the modulated is also in passive state. If the modulation input is not enabled, the output is in passive state.

If the Trap State is active (TRPS = 1), then the output enabled for the trap signal (by TRPEN13 = 1) is set to the passive state.

The output of the modulation control block is connected to a level select block. It offers the option to determine the actual output level of a pin, depending on the state of the output line (decoupling of active/passive state and output polarity) as specified by the Passive State Select bit PSLR.PSL63. If the modulated output signal is in the passive state, the level specified directly by PSL63 is output. If it is in the active state, the inverted level of PSL63 is output. This allows the user to adapt the polarity of an active output signal to the connected circuitry.

The PSL63 bit has a shadow register to allow for updates with the T13 shadow transfer signal (T13_ST) without undesired pulses on the output lines. A read action returns the actually used value, whereas a write action targets the shadow bit. Providing a shadow register for the PSL value as well as for other values related to the generation of the PWM signal facilitates a concurrent update by software for all relevant parameters.

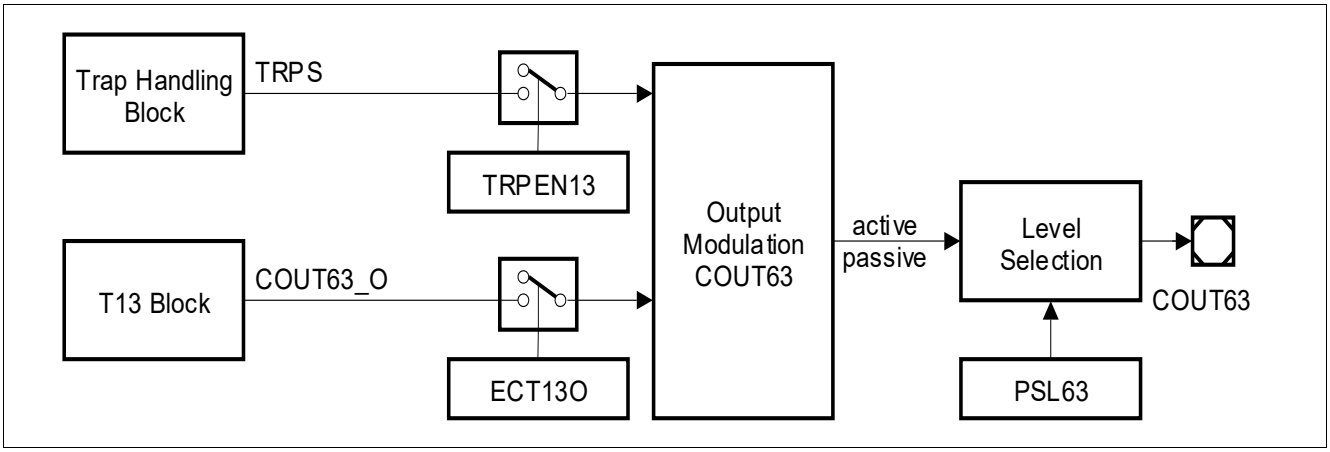


Figure 153 T13 Output Modulation

18.4.5 T13 Shadow Register Transfer

A special shadow transfer signal (T13_ST) can be generated to facilitate updating the period and compare values of the compare channel CC63 synchronously to the operation of T13. Providing a shadow register for values defining one PWM period facilitates a concurrent update by software for all relevant parameters. The next PWM period can run with a new set of parameters. The generation of this signal is requested by software via bit TCTR0.STE13 (set by writing 1 to the write-only bit TCTR4.T13STR, cleared by writing 1 to the write-only bit TCTR4.T13STD).

When signal T13_ST is active, a shadow register transfer is triggered with the next cycle of the T13 clock. Bit STE13 is automatically cleared with the shadow register transfer. A T13 shadow register transfer takes place (T13_ST active):

- while timer T13 is not running (T13R = 0), or
- STE13 = 1 and a Period-Match is detected while T13R = 1

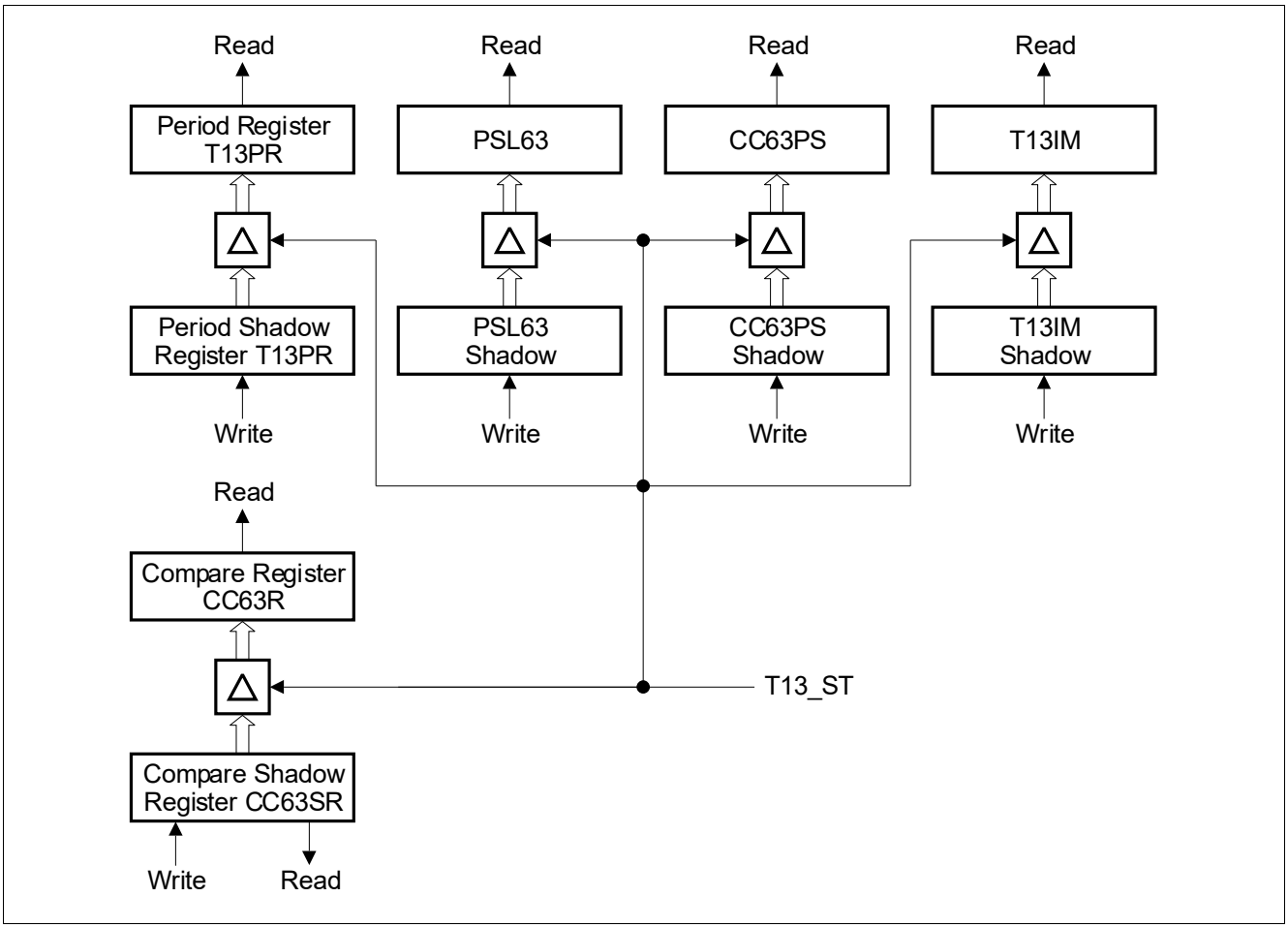


Figure 154 T13 Shadow Register Overview

18.5 Trap Handling

The trap functionality permits the PWM outputs to react on the state of the input signal $\overline{\text{CTRAP}}$. This functionality can be used to switch off the power devices if the trap input becomes active (e.g. to perform an emergency stop). The trap handling and the effect on the output modulation are controlled by the bits in the trap control register TRPCTR. The trap flags TRPF and TRPS are located in register IS and can be set/cleared by SW by writing to registers ISS and ISR.

Figure 155 gives an overview on the trap function.

The Trap Flag TRPF monitors the trap input and initiates the entry into the Trap State. The Trap State Bit TRPS determines the effect on the outputs and controls the exit of the Trap State.

When a trap condition is detected ($\overline{\text{CTRAP}}$ = 0) and the input is enabled (TRPPEN = 1), both, the Trap Flag TRPF and the Trap State Bit TRPS, are set to 1 (trap state active). The output of the Trap State Bit TRPS leads to the Output Modulation Blocks (for T12 and for T13) and can there deactivate the outputs (set them to the passive state). Individual enable control bits for each of the six T12-related outputs and the T13-related output facilitate a flexible adaptation to the application needs.

There are a number of different ways to exit the Trap State. This offers SW the option to select the best operation for the application. Exiting the Trap State can be done either immediately when the trap condition is removed ($\overline{\text{CTRAP}}$ = 1 or TRPPEN = 0), or under software control, or synchronously to the PWM generated by either Timer T12 or Timer T13.

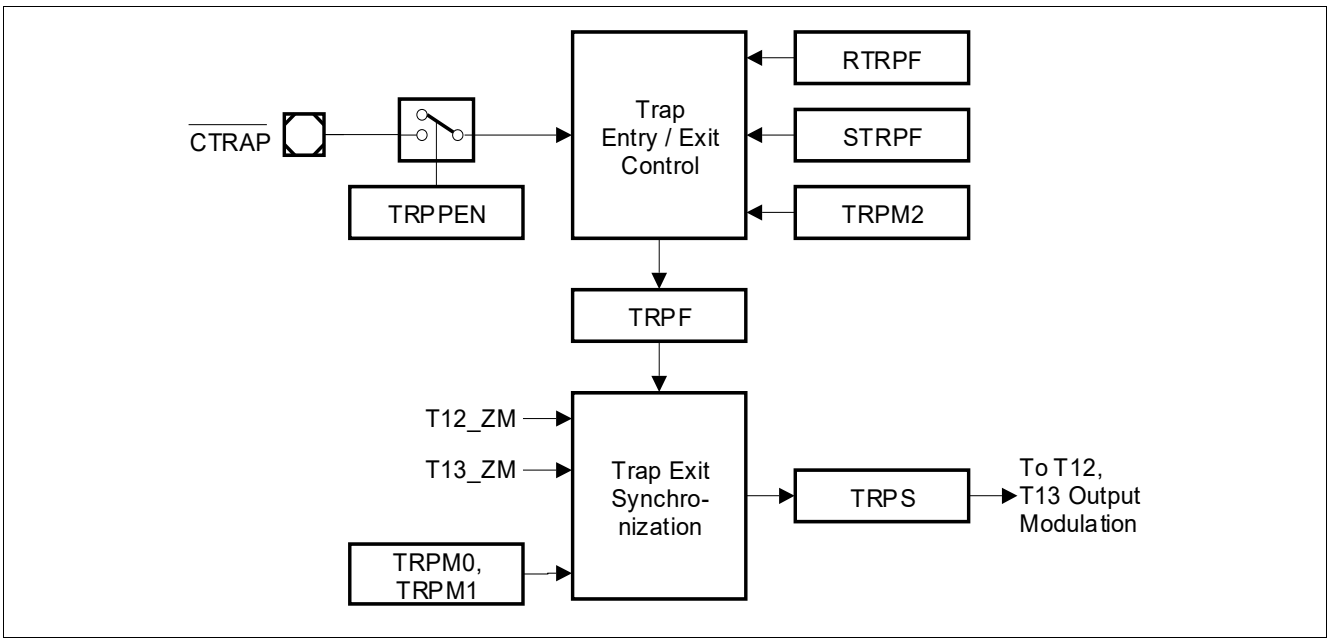


Figure 155 Trap Logic Block Diagram

Clearing of TRPF is controlled by the mode control bit TRPM2. If TRPM2 = 0, TRPF is automatically cleared by HW when $\overline{\text{CTRAP}}$ returns to the inactive level ($\overline{\text{CTRAP}}$ = 1) or if the trap input is disabled (TRPPEN = 0). When TRPM2 = 1, TRPF must be reset by SW after $\overline{\text{CTRAP}}$ has become inactive.

Clearing of TRPS is controlled by the mode control bits TRPM1 and TRPM0 (located in the Trap Control Register TRPCTR). A reset of TRPS terminates the Trap State and returns to normal operation. There are three options selected by TRPM1 and TRPM0. One is that the Trap State is left immediately when the Trap Flag TRPF is cleared, without any synchronization to timers T12 or T13. The other two options facilitate the synchronization of the termination of the Trap State to the count periods of either Timer T12 or Timer T13. **Figure 156** gives an overview on the associated operation.

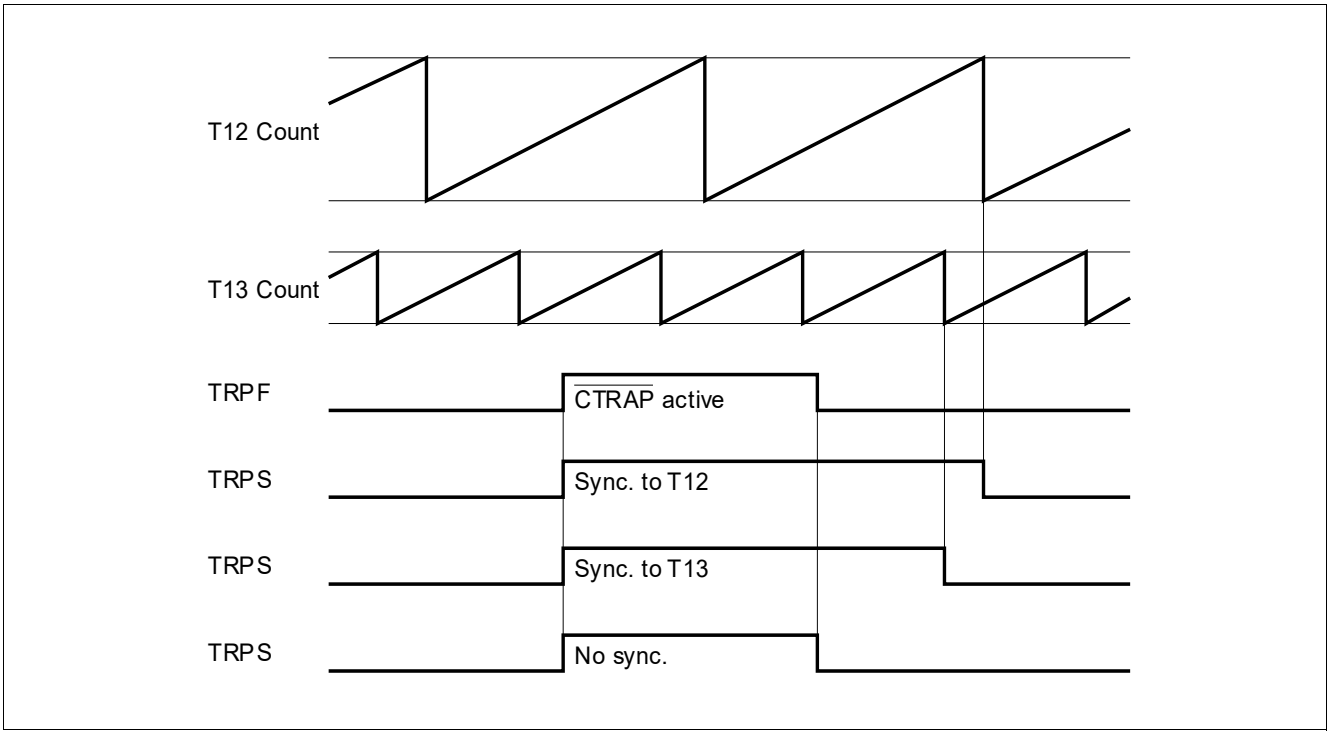


Figure 156 Trap State Synchronization (with TRPM2 = 0)

18.6 Multi-Channel Mode

The Multi-Channel mode offers the possibility to modulate all six T12-related output signals with one instruction. The bits in bit field MCMOUT.MCMP are used to specify the outputs that may become active. If Multi-Channel mode is enabled (bit MODCTR.MCMEN = 1), only those outputs may become active, that have a 1 at the corresponding bit position in bit field MCMP.

This bit field has its own shadow bit field MCMOUTS.MCMPS, that can be written by software. The transfer of the new value in MCMPS to the bit field MCMP can be triggered by, and synchronized to, T12 or T13 events. This structure permits the software to write the new value, that is then taken into account by the hardware at a well-defined moment and synchronized to a PWM signal. This avoids unintended pulses due to unsynchronized modulation sources.

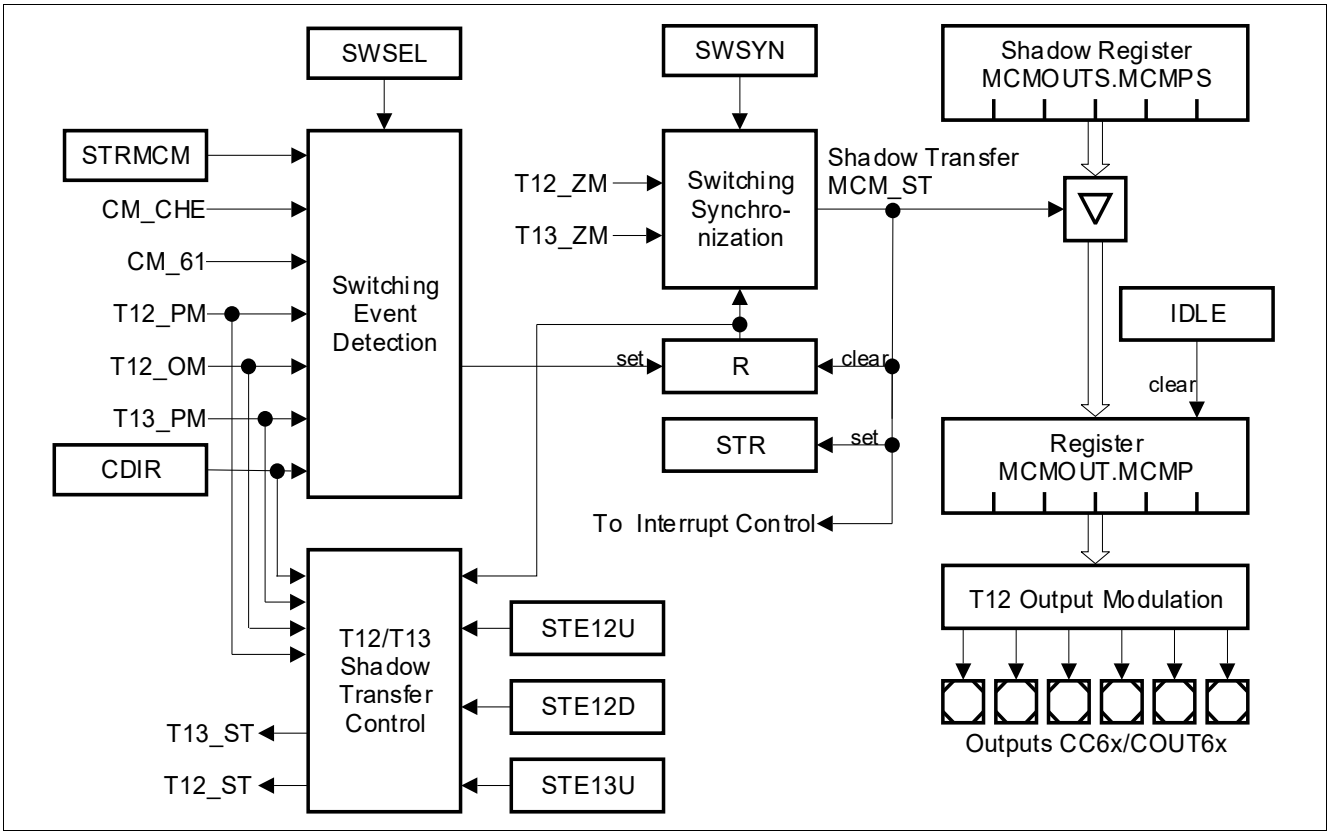


Figure 157 Multi-Channel Mode Block Diagram

Figure 157 shows the functional blocks for the Multi-Channel operation, controlled by bit fields in register MCMCTR. The event that triggers the update of bit field MCMP is chosen by SWSEL. In order to synchronize the update of MCMP to a PWM generated by T12 or T13, bit field SWSYN allows the selection of the synchronization event leading to the transfer from MCMPS to MCMP. Due to this structure, an update takes place with a new PWM period. A reminder flag R is set when the selected switching event occurs (the event is not necessarily synchronous to the modulating PWM), and is cleared when the transfer takes place. This flag can be monitored by software to check for the status of this logic block. If the shadow transfer from MCMPS to MCMP takes place, bit IS.STR becomes set and an interrupt can be generated.

In addition to the Multi-Channel shadow transfer event MCM_ST, the shadow transfers for T12 (T12_ST) and T13 (T13_ST) can be generated to allow concurrent updates of applied duty cycles for T12 and/or T13 modulation and Multi-Channel patterns.

If it is explicitly desired, the update takes place immediately with the occurrence of the selected event when the direct synchronization mode is selected. The update can also be requested by software by writing to bit

field MCMPS with the shadow transfer request bit STRMCM = 1. The option to trigger an update by SW is possible for all settings of SWSEL.

By using the direct mode and bit STRMCM = 1, the update takes place completely under software control.

Table 125 Multi-Channel Mode Switching Event Selection

SWSEL	Selected Event (see register MCMCTR)
000_B	No automatic event detection
001_B	Correct Hall Event (CM_CHE) detected at input signals CCPOSx without additional delay
010_B	T13 Period-Match (T13_PM)
011_B	T12 One-Match while counting down (T12_OM and CDIR = 1)
100_B	T12 Compare Channel 1 Event while counting up (CM_61 and CDIR = 0) to support the phase delay function by CC61 for block commutation mode.
101_B	T12 Period-Match while counting up (T12_PM and CDIR = 0)
110_B, 111_B	Reserved, no action

Table 126 Multi-Channel Mode Switching Synchronization

SWSYN	Synchronization Event (see register MCMCTR)
00_B	Direct Mode: the trigger event directly causes the shadow transfer
01_B	T13 Zero-Match (T13_ZM), the MCM shadow transfer is synchronized to a T13 PWM
10_B	T12 Zero-Match (T12_ZM), the MCM shadow transfer is synchronized to a T12 PWM
11_B	Reserved, no action

18.7 Hall Sensor Mode

For Brushless DC-Motors in block commutation mode, the Multi-Channel Mode has been introduced to provide efficient means for switching pattern generation. These patterns need to be output in relation to the angular position of the motor. For this, usually Hall sensors or Back-EMF sensing are used to determine the angular rotor position. The CCU6 provides three inputs, CCPOS0, CCPOS1, and CCPOS2, that can be used as inputs for the Hall sensors or the Back-EMF detection signals.

There is a strong correlation between the motor position and the output modulation pattern. When a certain position of the motor has been reached, indicated by the sampled Hall sensor inputs (the Hall pattern), the next, pre-determined Multi-Channel Modulation pattern has to be output. Because of different machine types, the modulation pattern for driving the motor can vary. Therefore, it is wishful to have a wide flexibility in defining the correlation between the Hall pattern and the corresponding Modulation pattern. Furthermore, a hardware mechanism significantly reduces the CPU for block-commutation.

The CCU6 offers the flexibility by having a register containing the currently assumed Hall pattern (CURH), the next expected Hall pattern (EXPH) and the corresponding output pattern (MCMP). A new Modulation pattern is output when the sampled Hall inputs match the expected ones (EXPH). To detect the next rotation phase (segment for block commutation), the CCU6 monitors the Hall inputs for changes. When the next expected Hall pattern is detected, the next corresponding Modulation pattern is output.

To increase for noise immunity (to a certain extend), the CCU6 offers the possibility to introduce a sampling delay for the Hall inputs. Some changes of the Hall inputs are not leading to the expected Hall pattern, because they are only short spikes due to noise. The Hall pattern compare logic compares the Hall inputs to the next expected pattern and also to the currently assumed pattern to filter out spikes.

For the Hall and Modulation output patterns, a double-register structure is implemented. While register MCMOUT holds the actually used values, its shadow register MCMOUTS can be loaded by software from a pre-defined table, holding the appropriate Hall and Modulation patterns for the given motor control.

A transfer from the shadow register into register MCMOUT can take place when a correct Hall pattern change is detected. Software can then load the next values into register MCMOUTS. It is also possible by software to force a transfer from MCMOUTS into MCMOUT.

Note: The Hall input signals CCPOSx and the CURH and EXPH bit fields are arranged in the following order:
CCPOS0 corresponds to CURH.0 (LSB) and EXPH.0 (LSB)
CCPOS1 corresponds to CURH.1 and EXPH.1
CCPOS2 corresponds to CURH.2 (MSB) and EXPH.2 (MSB)

18.7.1 Hall Pattern Evaluation

The Hall sensor inputs CCPOSx can be permanently monitored via an edge detection block (with the module clock f_{CC6}). In order to suppress spikes on the Hall inputs due to noise in rugged inverter environment, two optional noise filtering methods are supported by the Hall logic (both methods can be combined).

- Noise filtering with delay:
  For this function, the mode control bit fields MSEL6x for all T12 compare channels must be programmed to 1000_B and DBYP = 0. The selected event triggers Dead-Time Counter 0 to generate a programmable delay (defined by bit field DTM). When the delay has elapsed, the evaluation signal HCRDY becomes activated. Output modulation with T12 PWM signals is not possible in this mode.
- Noise filtering by synchronization to PWM:
  The Hall inputs are not permanently monitored by the edge detection block, but samples are taken only at defined points in time during a PWM period. This can be used to sample the Hall inputs when the switching noise (due to PWM) does not disturb the Hall input signals.

If neither the delay function of Dead-Time Counter 0 is not used for the Hall pattern evaluation nor the Hall mode for Brushless DC-Drive control is enabled, the timer T12 block is available for PWM generation and output modulation.

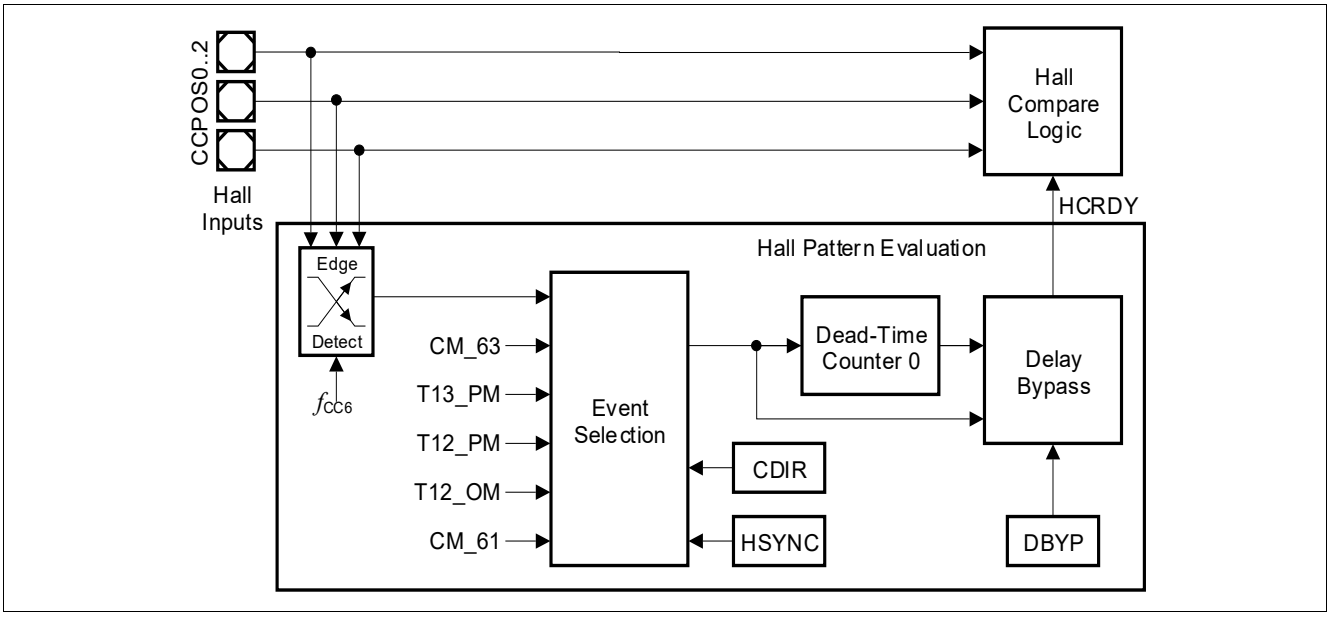


Figure 158 Hall Pattern Evaluation

If the evaluation signal HCRDY (Hall Compare Ready, see **Figure 159**) becomes activated, the Hall inputs are sampled and the Hall compare logic starts the evaluation of the Hall inputs.

Figure 158 illustrates the events for Hall pattern evaluation and the noise filter logic, **Table 127** summarizes the selectable trigger input signals.

Table 127 Hall Sensor Mode Trigger Event Selection

HSYNC	Selected Event (see register T12MSEL)
000_B	Any edge at any of the inputs CCPOSx, independent from any PWM signal (permanent check).
001_B	A T13 Compare-Match (CM_63).
010_B	A T13 Period-Match (T13_PM).
011_B	Hall sampling triggered by HW sources is switched off.

Table 127 Hall Sensor Mode Trigger Event Selection (cont'd)

HSYNC	Selected Event (see register T12MSEL)
100_B	A T12 Period-Match while counting up (T12_PM and CDIR = 0).
101_B	A T12 One-Match while counting down (T12_OM and CDIR = 1).
110_B	A T12 Compare-Match of compare channel CC61 while counting up (CM_61 and CDIR = 0).
111_B	A T12 Compare-Match of compare channel CC61 while counting down (CM_61 and CDIR = 1).

18.7.2 Hall Pattern Compare Logic

Figure 159 gives an overview on the double-register structure and the pattern compare logic. Software writes the next modulation pattern (MCMPS) and the corresponding current (CURHS) and expected (EXPHS) Hall patterns into the shadow register MCMOUTS. Register MCMOUT holds the actually used values CURH and EXPH. The modulation pattern MCMP is provided to the T12 Output Modulation block. The current (CURH) and expected (EXPH) Hall patterns are compared to the sampled Hall sensor inputs (visible in register CMPSTAT). Sampling of the inputs and the evaluation of the comparator outputs is triggered by the evaluation signal HCRDY (Hall Compare Ready), that is detailed in the next section.

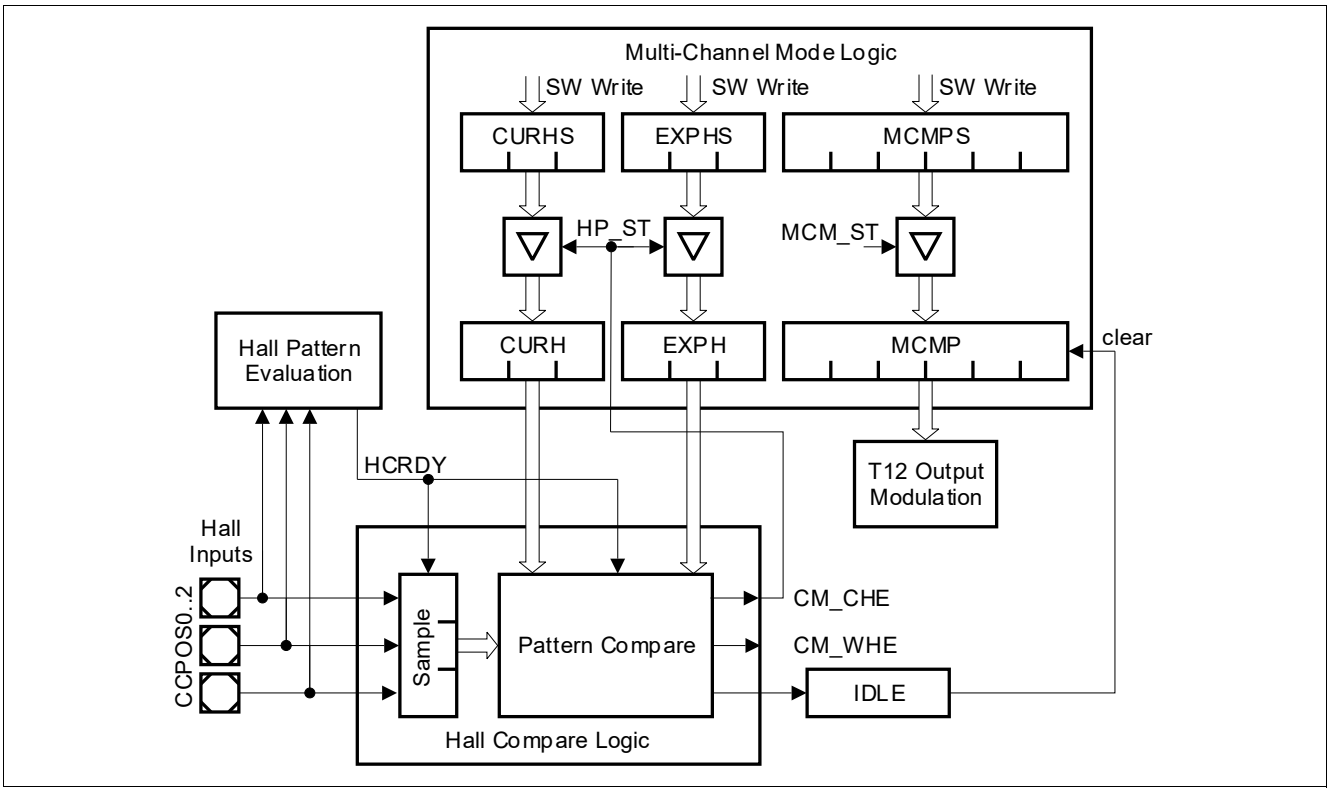


Figure 159 Hall Pattern Compare Logic

- If the sampled Hall pattern matches the value programmed in CURH, the detected transition was a spike (no Hall event) and no further actions are necessary.
- If the sampled Hall pattern matches the value programmed in EXPH, the detected transition was the expected event (correct Hall event CM_CHE) and the MCMP value has to change.
- If the sampled Hall pattern matches neither CURH nor EXPH, the transition was due to a major error (wrong Hall event CM_WHE) and can lead to an emergency shut down (IDLE).

At every correct Hall event (CM_CHE), the next Hall patterns are transferred from the shadow register MCMOUTS into MCMOUT (Hall pattern shadow transfer HP_ST), and a new Hall pattern with its corresponding output pattern can be loaded (e.g. from a predefined table in memory) by software into MCMOUTS. For the Modulation patterns, signal MCM_ST is used to trigger the transfer.

Loading this shadow register can also be done by writing MCMOUTS.STRHP = 1 (for EXPH and CURH) or MCMOUTS.STRMCMP = 1 (for MCMP).

18.7.3 Hall Mode Flags

Depending on the Hall pattern compare operation, a number of flags are set in order to indicate the status of the module and to trigger further actions and interrupt requests.

Flag IS.CHE (Correct Hall Event) is set by signal CM_CHE when the sampled Hall pattern matches the expected one (EXPH). This flag can also be set by SW by setting bit ISS.SCHE = 1. If enabled by bit IEN.ENCHE = 1, the set signal for CHE can also generate an interrupt request to the CPU. Bit field INP.INPCHE defines which service request output becomes activated in case of an interrupt request. To clear flag CHE, SW needs to write ISR.RCHE = 1.

Flag IS.WHE indicates a Wrong Hall Event. Its handling for flag setting and resetting as well as interrupt request generation are similar to the mechanism for flag CHE.

The implementation of flag STR is done in the same way as for CHE and WHE. This flag is set by HW by the shadow transfer signal MCM_ST (see also **Figure 157**).

Please note that for flags CHE, WHE, and STR, the interrupt request generation is triggered by the set signal for the flag. That means, a request can be generated even if the flag is already set. There is no need to clear the flag in order to enable further interrupt requests.

The implementation for the IDLE flag is different. It is set by HW through signal CM_WHE if enabled by bit ENIDLE. Software can also set the flag via bit SIDLE. As long as bit IDLE is set, the modulation pattern field MCMP is cleared to force the outputs to the passive state. Flag IDLE must be cleared by software by writing RIDLE = 1 in order to return to normal operation. To fully restart from IDLE mode, the transfer requests for the bit fields in register MCMOUTS to register MCMOUT have to be initiated by software via bits STRMCM and STRHP in register MCMOUTS. In this way, the release from IDLE mode is under software control, but can be performed synchronously to the PWM signal.

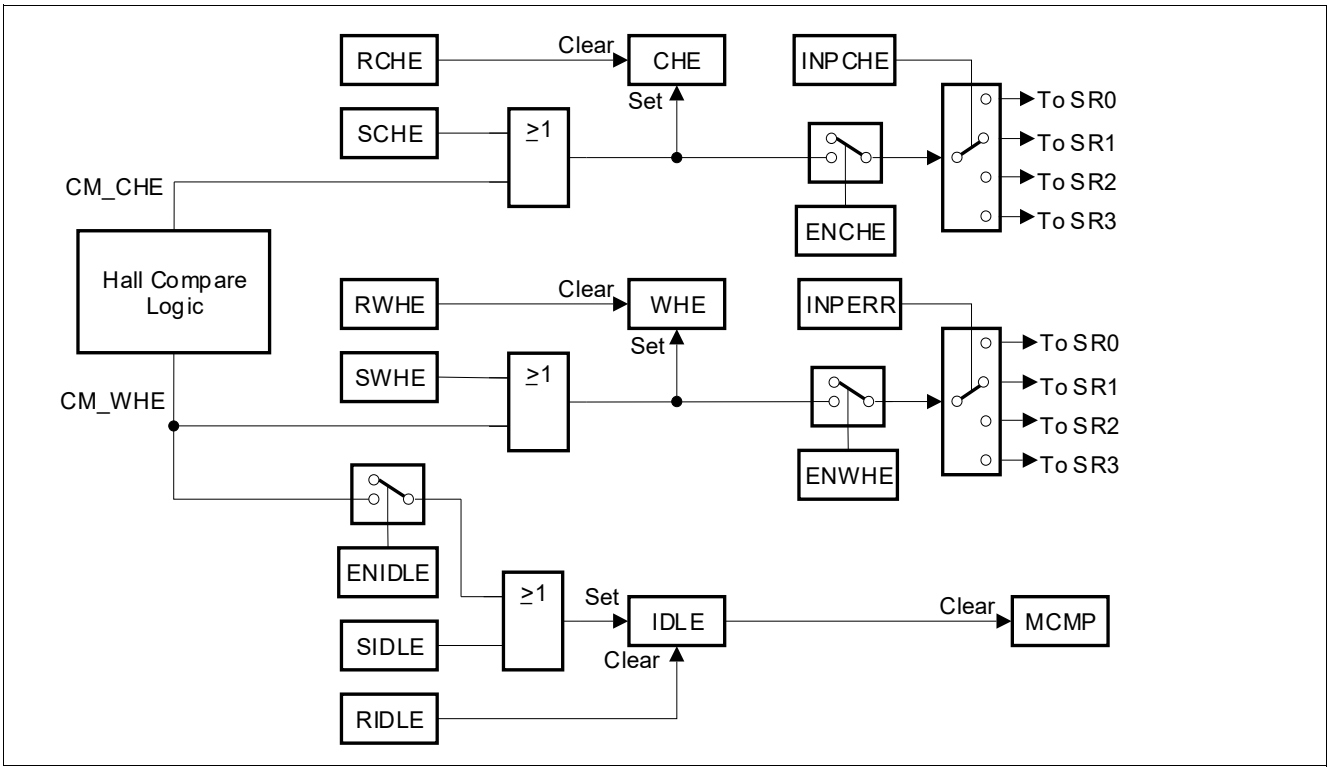


Figure 160 Hall Mode Flags

18.7.4 Hall Mode for Brushless DC-Motor Control

The CCU6 provides a mode for the Timer T12 Block especially targeted for convenient control of block commutation patterns for Brushless DC-Motors. This mode is selected by setting all T12MSEL.MSEL6x bit fields of the three T12 Channels to 1000_B.

In this mode, illustrated in **Figure 161**, channel CC60 is placed in capture mode to measure the time elapsed between the last two correct Hall events, channel CC61 in compare mode to provide a programmable phase delay between the Hall event and the application of a new PWM output pattern, and channel CC62 also in compare mode as first time-out criterion. A second time-out criterion can be built by the T12 period match event.

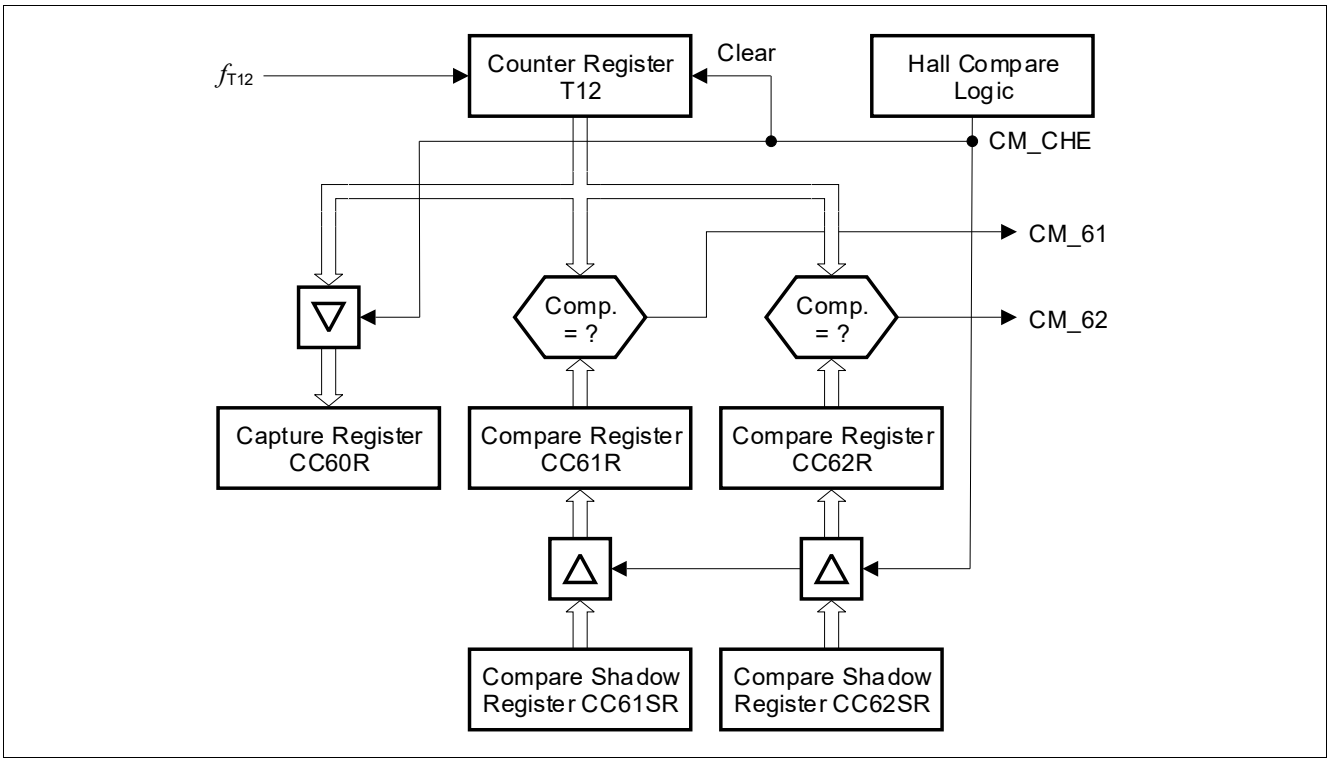


Figure 161 T12 Block in Hall Sensor Mode

The signal CM_CHE from the Hall compare logic is used to transfer the new compare values from the shadow registers CC6xSR into the actual compare registers CC6xR, performs the shadow transfer for the T12 period register, to capture the current T12 contents into register CC60R, and to clear T12.

Note: In this mode, the shadow transfer signal T12_ST is not generated. Not all shadow bits, such as the PSLy bits, will be transferred to their main registers. To program the main registers, SW needs to write to these registers while Timer T12 is stopped. In this case, a SW write actualizes both registers.

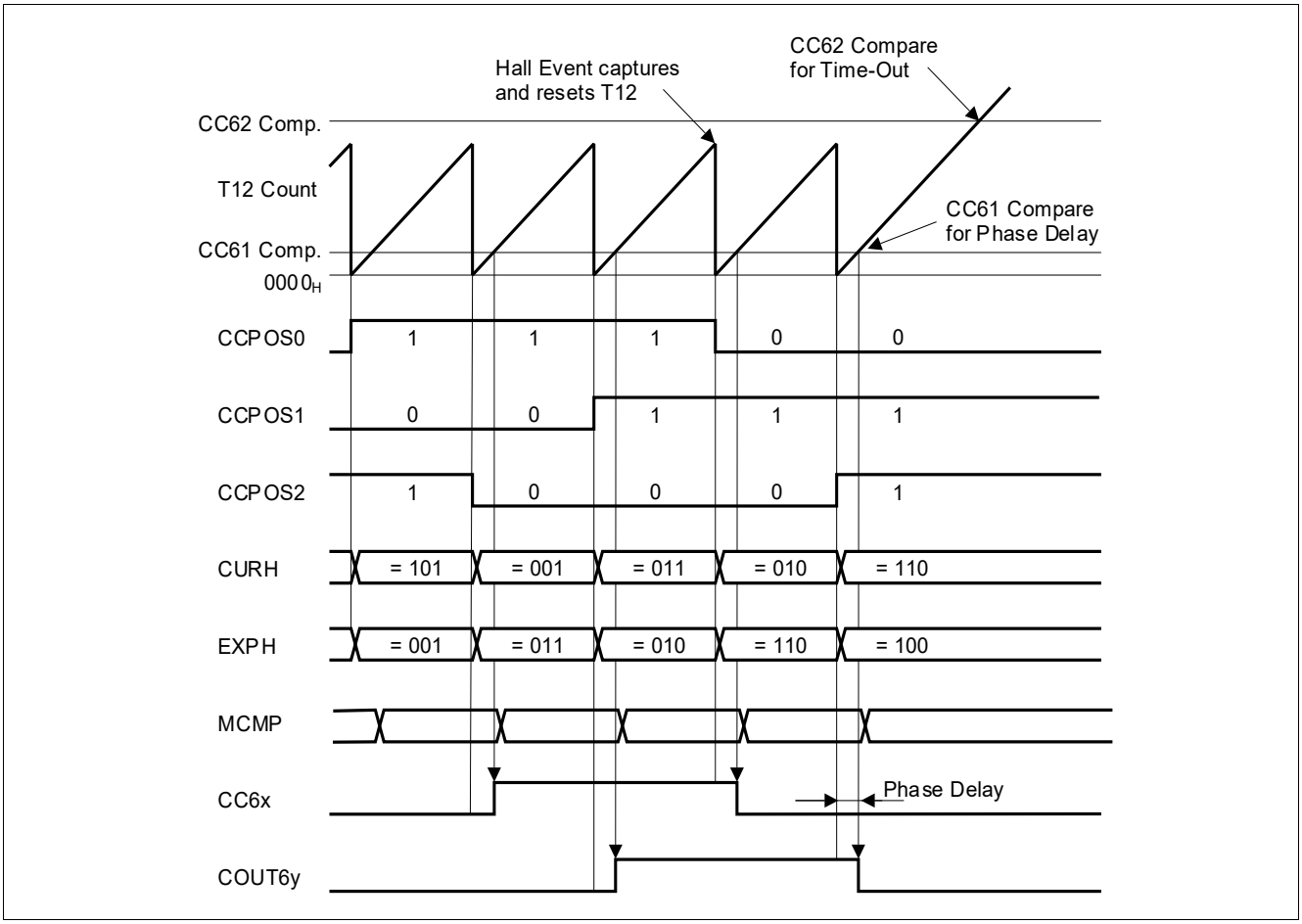


Figure 162 Brushless DC-Motor Control Example (all MSEL6x = 1000_B)

After the detection of an expected Hall pattern (CM_CHE active), the T12 count value is captured into channel CC60 (representing the actual rotor speed by measuring the elapsed time between the last two correct Hall events), and T12 is reset. When the timer reaches the compare value in channel CC61, the next multi-channel state is switched by triggering the shadow transfer of bit field MCMP (if enabled in bit field **SWEN**). This trigger event can be combined with the synchronization of the next multi-channel state to the PWM source (to avoid spikes on the output lines, see **Section 18.6**). This compare function of channel CC61 can be used as a phase delay from the position sensor input signals to the switching of the output signals, that is necessary if a sensorless back-EMF technique or Hall sensors are used. The compare value in channel CC62 can be used as a time-out trigger (interrupt), indicating that the actual motor speed is far below the desired destination value. An abnormal load change can be detected with this feature and PWM generation can be disabled.

18.8 General Module Operation

This section provides information about the:

- Input selection (see **Section 18.8.1**)

18.8.1 Input Selection

Each CCU6 input signal can be selected from a vector of four or eight possible inputs by programming the port input select registers PISEL0 and PISEL2. This permits to adapt the pin functionality of the device to the application requirements.

The output pins for the module output signals are chosen in the ports.

Note: All functional inputs of the CCU6 are synchronized to f_{CC6} before they affect the module internal logic. The resulting delay of $2/f_{CC6}$ and for asynchronous signals an additional uncertainty of $1/f_{CC6}$ have to be taken into account for precise timing calculation. An edge of an input signal can only be correctly detected if the high phase and the low phase of the input signal are both longer than $1/f_{CC6}$.

18.9 Module Interfaces

This section describes the CCU6 module interfaces with the clock control, port connections, interrupt control, and address decoding.

18.9.1 Interfaces of the CCU6 Module

An overview of the CCU6 kernel I/O interface is shown in **Figure 163**.

The interrupt lines of the CCU6 are connected to the CPU interrupt controller via the SCU. An interrupt pulse can be generated at one of the four interrupt output lines SRCx (x=0 to 4) of the module. More than one CCU6 interrupt source can be connected to each CCU6 interrupt line.

The General Purpose IO (GPIO) Ports provide the interface from the CCU6 to the external world. Please refer to **Chapter 14** for Port implementation details.

The CCU6 kernel is clocked on PCLK frequency where $f_{CCU} = f_{PCLK}$.

Debug Suspend of Timers

The timers of CCU6, T12 and T13, can be suspended immediately when Debug Mode enters Monitor Mode and has the Debug-Suspend signal activated – provided the respective timer suspend bits, T12SUSP and T13SUSP (in SCU SFR MODSUSP), are set. When suspended, the respective timer stops and its PWM outputs enabled for the trap condition (**TRPCTR**.TRPENx = 1) are set to respective passive levels (similar to TRAP state). In addition, all CCU6 inputs are frozen. Refer to SCU **Chapter 6.9**.

Flexible Peripheral Management (Kernel Clock Gating) of CCU6

When not in use, the CCU6 kernel may be disabled where the kernel clock input is gated. When the **PMCON1**.CCU_DIS request bit is set, both T12 and T13 are immediately stopped and PWM outputs enabled for the trap condition (**TRPCTR**.TRPENx = 1) are set to respective passive levels (similar to TRAP state). In addition, all CCU6 inputs are frozen. Finally, the kernel clock input is gated. Refer to SCU **Chapter 6.8**.

The following figure shows all interrupt and interface signals and GPIO interface associated with the CCU6 module kernel.

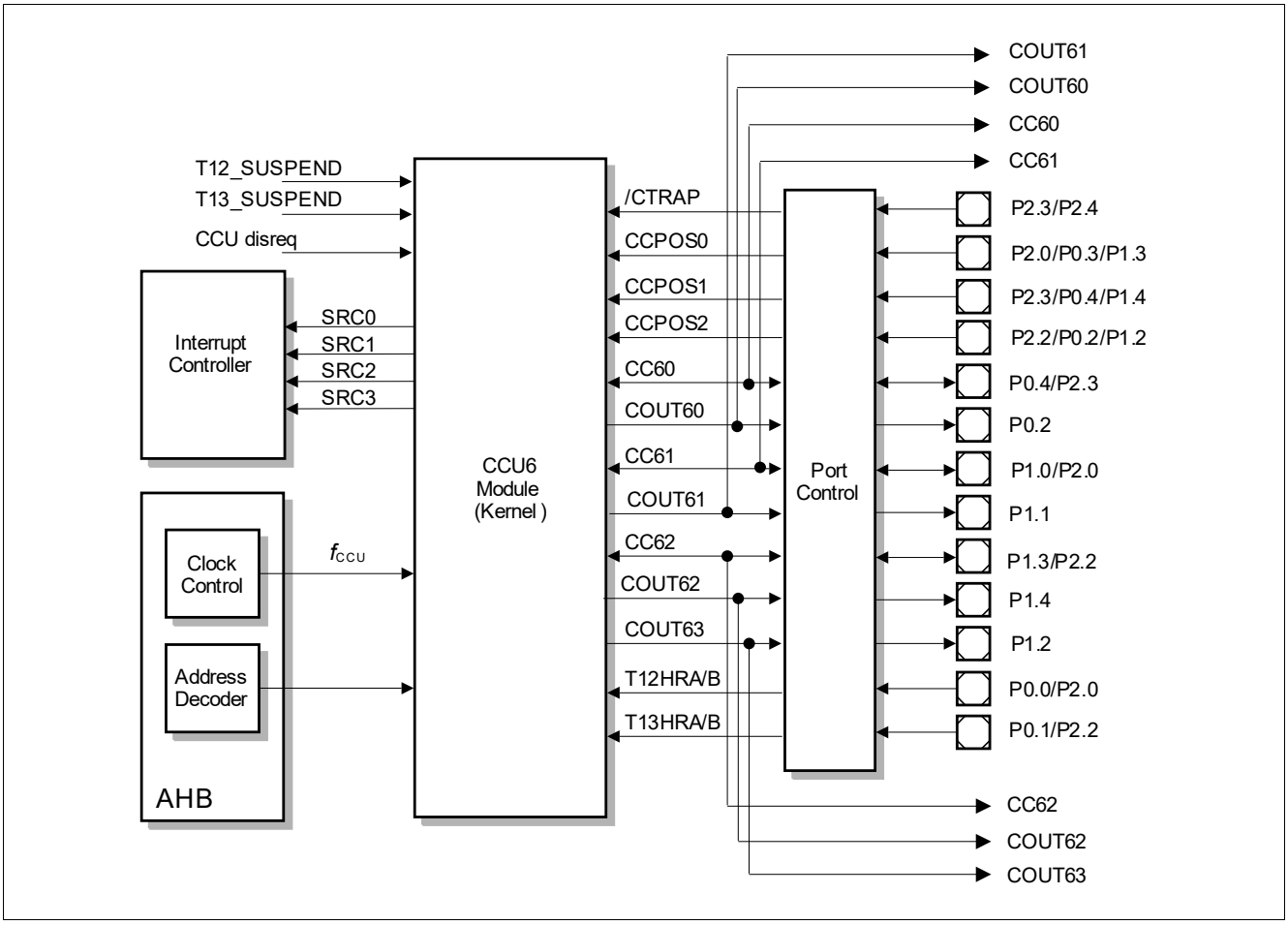


Figure 163 Interconnections of the CCU6 Module

18.10 Interrupt Handling

This section describes the interrupt handling of the CCU6 module.

18.10.1 Interrupt Structure

The HW interrupt event or the SW setting of the corresponding interrupt set bit (in register ISS) sets the event indication flags (in register IS) and can trigger the interrupt generation. The interrupt pulse is generated independently from the interrupt status flag in register IS (it is not necessary to clear the related status bit to be able to generate another interrupt). The interrupt flag can be cleared by SW by writing to the corresponding bit in register ISR.

If enabled by the related interrupt enable bit in register IEN, an interrupt pulse can be generated on one of the four service request outputs (SR0 to SR3) of the module. If more than one interrupt source is connected to the same interrupt node pointer (in register INP), the requests are logically OR-combined to one common service request output (see **Figure 164**).

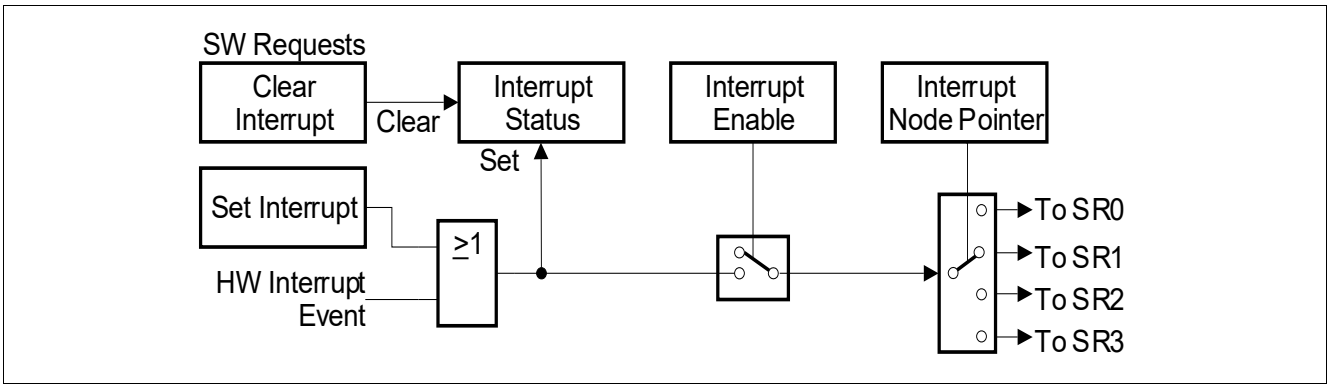


Figure 164 General Interrupt Structure

The available interrupt events in the CCU6 are shown in **Figure 165**.

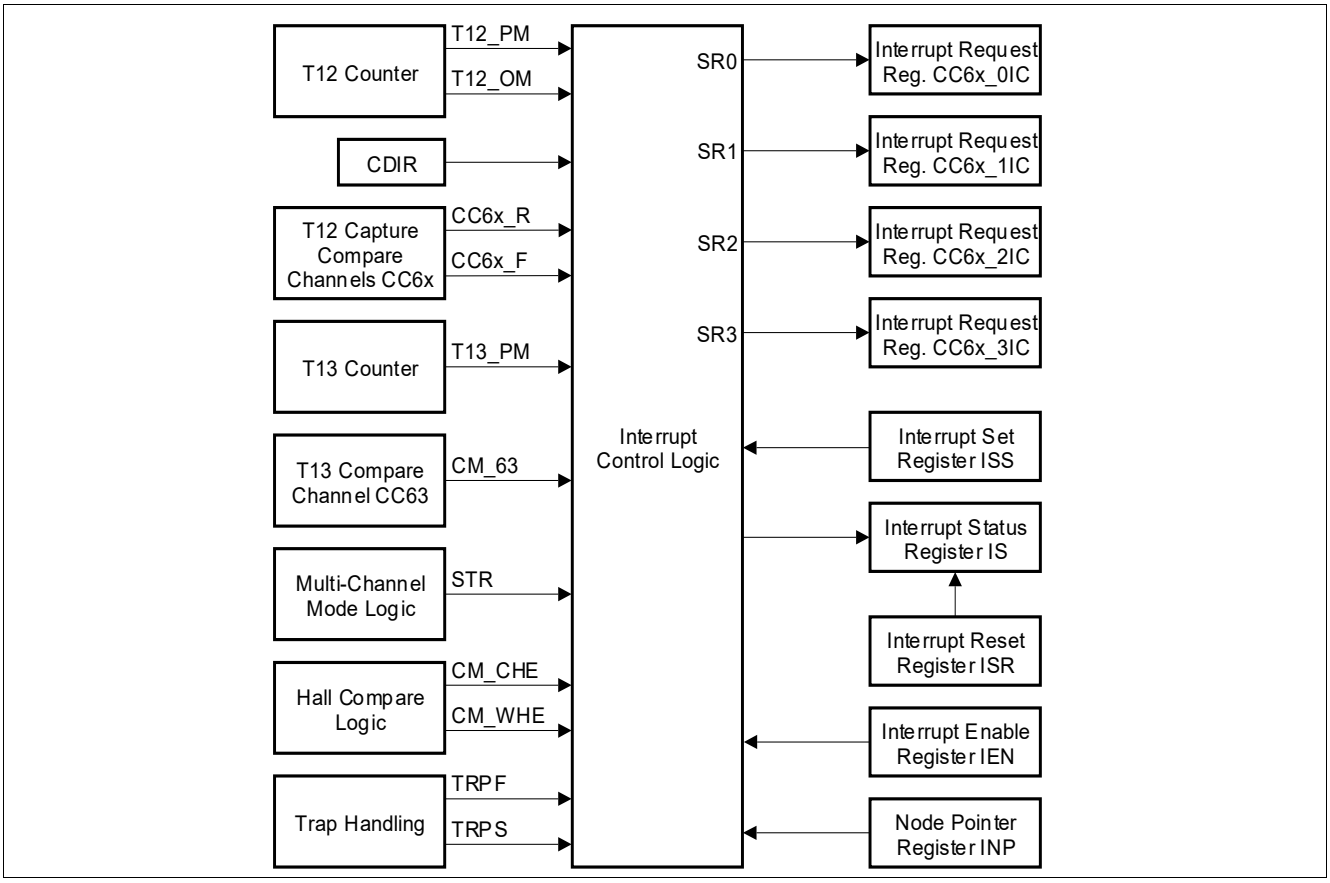


Figure 165 Interrupt Sources and Events

18.11 Register Definition

18.11.1 CCU6 Register Description

All CCU6 kernel register names described in this section will be referenced in other parts of this specification with the module name prefix “CCU6_”.

Note: If a hardware and a software request to modify a bit occur simultaneously, the software wins.

Table 128 Registers Overview

Register Short Name	Register Long Name	Description see
System Registers		
PISEL0	Port Input Select Register 0	**Page 574**
PISEL2	Port Input Select Register 2	**Page 576**
Timer12 – Related Registers		
T12	Timer T12 Counter Register	**Page 581**
T12PR	Timer T12 Period Register	**Page 582**
CC6xR (x = 0-2)	Capture/Compare Register for Channel CC6x	**Page 582**
CC6xSR (x = 0-2)	Compare Shadow Register for Channel CC6x	**Page 583**
T12DTC	Timer T12 Dead-Time Control Register	**Page 584**
Timer13 – Related Registers		
T13	Timer T13 Counter Register	**Page 585**
T13PR	Timer T13 Period Register	**Page 585**
CC63R	Capture/Compare Register for Channel CC63	**Page 586**
CC63SR	Capture/Compare Shadow Register for Channel CC63	**Page 586**
Capture/Compare Control Registers		
CMPSTAT	Compare State Register	**Page 587**
CMPMODIF	Compare State Modification Register	**Page 588**
T12MSEL	Capture/Compare T12 Mode Select Register	**Page 579**
TCTR0	Timer Control Register 0	**Page 590**
TCTR2	Timer Control Register 2	**Page 593**
TCTR4	Timer Control Register 4	**Page 595**
Global Modulation Control Registers, **Multi-Channel Modulation Control Registers**		
MODCTR	Modulation Control Register	**Page 597**
TRPCTR	Trap Control Register	**Page 599**
PSLR	Passive State Level Register	**Page 601**
MCMOUTS	Multi-Channel Mode Output Shadow Register	**Page 602**
MCMOUT	Multi-Channel Mode Output Register	**Page 603**
MCMCTR	Multi-Channel Mode Control Register	**Page 605**

Table 128 Registers Overview (cont'd)

Register Short Name	Register Long Name	Description see
Interrupt Control Registers		
IS	Capture/Compare Interrupt Status Register	**Page 607**
ISS	Capture/Compare Interrupt Status Set Register	**Page 609**
ISR	Capture/Compare Interrupt Status Reset Register	**Page 611**
IEN	Capture/Compare Interrupt Enable Register	**Page 612**
INP	Capture/Compare Interrupt Node Pointer Register	**Page 615**

18.11.1.1 System Registers

Registers PISEL0 and PISEL2 contain bit fields that select the actual input port/signal for the module inputs. This permits the adaptation of the pin functionality of the device to the application's requirements. The output pins are chosen according to the registers in the ports.

PISEL0
Port Input Select Register 0 **($6C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IST12HR		ISPOS2		ISPOS1		ISPOS0		ISTRP		ISCC62		ISCC61		ISCC60	
rw		rw		rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
ISCC60	1:0	rw	**Input Select for CC60** This bit field defines the port pin that is used for the CC60 capture input signal. 00_B The input pin for CC60_0. 01_B The input pin for CC60_1. 10_B Reserved 11_B Reserved
ISCC61	3:2	rw	**Input Select for CC61** This bit field defines the port pin that is used for the CC61 capture input signal. 00_B The input pin for CC61_0. 01_B Reserved 10_B Reserved 11_B Reserved
ISCC62	5:4	rw	**Input Select for CC62** This bit field defines the port pin that is used for the CC62 capture input signal. 00_B The input pin for CC62_0. 01_B The input pin for CC62_1. 10_B The input pin for CC62_2. 11_B Reserved
ISTRP	7:6	rw	**Input Select for CTRAP** This bit field defines the port pin that is used for the $\overline{\text{CTRAP}}$ input signal. 00_B The input pin for $\overline{\text{CTRAP_0}}$. 01_B The input pin for $\overline{\text{CTRAP_1}}$. 10_B Reserved 11_B Reserved

Field	Bits	Type	Description
ISPOS0	9:8	rw	**Input Select for CCPOS0** This bit field defines the port pin that is used for the CCPOS0 input signal. 00_B Reserved 01_B The input pin for CCPOS0_1. 10_B The input pin for CCPOS0_2. 11_B The input pin for CCPOS0_3 if **BEMFC_CTRL_STS**.CCPOS_INSEL = 0. 11_B BEMF Comparator if **BEMFC_CTRL_STS**.CCPOS_INSEL = 1.
ISPOS1	11:10	rw	**Input Select for CCPOS1** This bit field defines the port pin that is used for the CCPOS1 input signal. 00_B The input pin for CCPOS1_0. 01_B The input pin for CCPOS1_1. 10_B The input pin for CCPOS1_2. 11_B Reserved if **BEMFC_CTRL_STS**.CCPOS_INSEL = 0. 11_B BEMF Comparator if **BEMFC_CTRL_STS**.CCPOS_INSEL = 1.
ISPOS2	13:12	rw	**Input Select for CCPOS2** This bit field defines the port pin that is used for the CCPOS2 input signal. 00_B Reserved 01_B The input pin for CCPOS2_1. 10_B The input pin for CCPOS2_2. 11_B The input pin for CCPOS2_3 if **BEMFC_CTRL_STS**.CCPOS_INSEL = 0. 11_B BEMF Comparator if **BEMFC_CTRL_STS**.CCPOS_INSEL = 1.
IST12HR	15:14	rw	**Input Select for T12HR** This bit field defines the input signal used as T12HR input. 00_B Either signal T12HRA (if T12EXT = 0) or T12HRE (if T12EXT = 1) is selected. 01_B Either signal T12HRB (if T12EXT = 0) or T12HRF (if T12EXT = 1) is selected. 10_B Either signal T12HRC (if T12EXT = 0) or T12HRG (if T12EXT = 1) is selected. 11_B Either signal T12HRD (if T12EXT = 0) or T12HRH (if T12EXT = 1) is selected.

PISEL2
Port Input Select Register 2 **(74_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
T13EXT	**T12EXT**	**ISCNT13**		**ISCNT12**		**IST13HR**	
rw	rw	rw		rw		rw	

Field	Bits	Type	Description
IST13HR	1:0	rw	**Input Select for T13HR** This bit field defines the input signal used as T13HR input. 00_B Either signal T13HRA (if T13EXT = 0) or T13HRE (if T13EXT = 1) is selected. 01_B Either signal T13HRB (if T13EXT = 0) or T13HRF (if T13EXT = 1) is selected. 10_B Either signal T13HRC (if T13EXT = 0) or T13HRG (if T13EXT = 1) is selected. 11_B Either signal T13HRD (if T13EXT = 0) or T13HRH (if T13EXT = 1) is selected.
ISCNT12	3:2	rw	**Input Select for T12 Counting Input** This bit field defines the input event leading to a counting action of T12. 00_B The T12 prescaler generates the counting events. Bit TCTR4.T12CNT is not taken into account. 01_B Bit TCTR4.T12CNT written with 1 is a counting event. The T12 prescaler is not taken into account. 10_B The timer T12 is counting each rising edge detected in the selected T12HR signal. 11_B The timer T12 is counting each falling edge detected in the selected T12HR signal.
ISCNT13	5:4	rw	**Input Select for T13 Counting Input** This bit field defines the input event leading to a counting action of T13. 00_B The T13 prescaler generates the counting events. Bit TCTR4.T13CNT is not taken into account. 01_B Bit TCTR4.T13CNT written with 1 is a counting event. The T13 prescaler is not taken into account. 10_B The timer T13 is counting each rising edge detected in the selected T13HR signal. 11_B The timer T13 is counting each falling edge detected in the selected T13HR signal.
T12EXT	6	rw	**Extension for T12HR Inputs** This bit extends the 2-bit field IST12HR. 0_B One of the signals T12HR[D:A] is selected. 1_B One of the signals T12HR[H:E] is selected.

Field	Bits	Type	Description
T13EXT	7	rw	**Extension for T13HR Inputs** This bit extends the 2-bit field IST13HR. 0_B One of the signals T13HR[D:A] is selected. 1_B One of the signals T13HR[H:E] is selected.

18.11.2 Timer12 – Related Registers

The generation of the patterns for a 3-channel PWM is based on timer T12. The registers related to timer T12 can be concurrently updated (with well-defined conditions) in order to ensure consistency of the three PWM channels.

Timer T12 supports capture and compare modes, which can be independently selected for the three channels CC60, CC61, and CC62.

Register T12MSEL contains control bits to select the capture/compare functionality of the three channels of timer T12. **Table 129**, **Table 130** and **Table 131** define and elaborate some of the capture/compare modes selectable. Refer to the following register description for the selection.

Table 129 Double-Register Capture Modes

Description	
0100	The contents of T12 are stored in CC6nR after a rising edge and in CC6nSR after a falling edge on the input pin CC6n.
0101	The value stored in CC6nSR is copied to CC6nR after a rising edge on the input pin CC6n. The actual timer value of T12 is simultaneously stored in the shadow register CC6nSR. This feature is useful for time measurements between consecutive rising edges on pins CC6n. COUT6n is I/O.
0110	The value stored in CC6nSR is copied to CC6nR after a falling edge on the input pin CC6n. The actual timer value of T12 is simultaneously stored in the shadow register CC6nSR. This feature is useful for time measurements between consecutive falling edges on pins CC6n. COUT6n is I/O.
0111	The value stored in CC6nSR is copied to CC6nR after any edge on the input pin CC6n. The actual timer value of T12 is simultaneously stored in the shadow register CC6nSR. This feature is useful for time measurements between consecutive edges on pins CC6n. COUT6n is I/O.

Table 130 Combined T12 Modes

Description	
1000	Hall Sensor mode: Capture mode for channel 0, compare mode for channels 1 and 2. The contents of T12 are captured into CC60 at a valid hall event (which is a reference to the actual speed). CC61 can be used for a phase delay function between hall event and output switching. CC62 can act as a time-out trigger if the expected hall event comes too late. The value 1000_B must be programmed to MSEL0, MSEL1 and MSEL2 if the hall signals are used. In this mode, the contents of timer T12 are captured in CC60 and T12 is reset after the detection of a valid hall event. In order to avoid noise effects, the dead-time counter channel 0 is started after an edge has been detected at the hall inputs. On reaching the value of 000001_B, the hall inputs are sampled and the pattern comparison is done.
1001	Hysteresis-like control mode with dead-time generation: The negative edge of the CCPOSx input signal is used to reset bit CC6nST. As a result, the output signals can be switched to passive state immediately and switch back to active state (with dead-time) if the CCPOSx is high and the bit CC6nST is set by a compare event.

Table 131 Multi-Input Capture Modes

Description	
1010	The timer value of T12 is stored in CC6nR after a rising edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a falling edge at the input pin CCPOSx.
1011	The timer value of T12 is stored in CC6nR after a falling edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a rising edge at the input pin CCPOSx.
1100	The timer value of T12 is stored in CC6nR after a rising edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a rising edge at the input pin CCPOSx.
1101	The timer value of T12 is stored in CC6nR after a falling edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after a falling edge at the input pin CCPOSx.
1110	The timer value of T12 is stored in CC6nR after any edge at the input pin CC6n. The timer value of T12 is stored in CC6nSR after any edge at the input pin CCPOSx.
1111	reserved (no capture or compare action)

T12MSEL
Capture/Compare T12 Mode Select Register **(40_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBYP	**HSYNC**			**MSEL62**				**MSEL61**				**MSEL60**			
rw	rw			rw				rw				rw			

Field	Bits	Type	Description
MSEL60, MSEL61	3:0, 7:4	rw	**Capture/Compare Mode Selection** These bit fields select the operating mode of the three timer T12 capture/compare channels. Each channel (n = 0, 1, 2) can be programmed individually either for compare or capture operation according to: 0000_BCompare outputs disabled, pins CC6n and COUT6n can be used for I/O. No capture action. 0001_BCompare output on pin CC6n, pin COUT6n can be used for I/O. No capture action. 0010_BCompare output on pin COUT6n, pin CC6n can be used for I/O. No capture action. 0011_BCompare output on pins COUT6n and CC6n. $01XX_B$Double-Register Capture modes, see **Table 129**. 1000_BHall Sensor mode, see **Table 130**. In order to enable the hall edge detection, all three MSEL6x must be programmed to Hall Sensor mode. 1001_BHysteresis-like mode, see **Table 130**. $101X_B$Multi-Input Capture modes, see **Table 131**. $11XX_B$Multi-Input Capture modes, see **Table 131**.
MSEL62	11:8	rw	**Capture/Compare Mode Selection** These bit fields select the operating mode of the three timer T12 capture/compare channels. Each channel (n = 0, 1, 2) can be programmed individually either for compare or capture operation according to: 0000_BCompare outputs disabled, pins CC6n and COUT6n can be used for I/O. No capture action. 0001_BCompare output on pin CC6n, pin COUT6n can be used for I/O. No capture action. 0010_BCompare output on pin COUT6n, pin CC6n can be used for I/O. No capture action. 0011_BCompare output on pins COUT6n and CC6n. $01XX_B$Double-Register Capture modes, see **Table 129**. 1000_BHall Sensor mode, see **Table 130**. In order to enable the hall edge detection, all three MSEL6x must be programmed to Hall Sensor mode. 1001_BHysteresis-like mode, see **Table 130**. $101X_B$Multi-Input Capture modes, see **Table 131**. $11XX_B$Multi-Input Capture modes, see **Table 131**.

Field	Bits	Type	Description
HSYNC	14:12	rw	**Hall Synchronization** Bit field HSYNC defines the source for the sampling of the Hall input pattern and the comparison to the current and the expected Hall pattern bit fields. In all modes, a trigger by software by writing a 1 to bit SWHC is possible. 000_B Any edge at one of the inputs CCPOSx (x = 0, 1, 2) triggers the sampling. 001_B A T13 compare-match triggers the sampling. 010_B A T13 period-match triggers the sampling. 011_B The Hall sampling triggered by hardware sources is switched off. 100_B A T12 period-match (while counting up) triggers the sampling. 101_B A T12 one-match (while counting down) triggers the sampling. 110_B A T12 compare-match of channel 1 (while counting up) triggers the sampling. 111_B A T12 compare-match of channel 1 (while counting down) triggers the sampling.
DBYP	15	rw	**Delay Bypass** Bit DBYP defines if the source signal for the sampling of the Hall input pattern (selected by HSYNC) uses the dead-time counter DTC0 of timer T12 as additional delay or if the delay is bypassed. 0_B The delay bypass is not active. The dead-time counter DTC0 is generating a delay after the source signal becomes active. 1_B The delay bypass is active. The dead-time counter DTC0 is not used by the sampling of the Hall pattern.

Register T12 represents the counting value of timer T12. It can only be written while the timer T12 is stopped. Write actions while T12 is running are not taken into account. Register T12 can always be read by software.

In edge-aligned mode, T12 only counts up, whereas in center-aligned mode, T12 can count up and down.

T12
Timer T12 Counter Register **(78_H)** **Reset Value: 00_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T12CV															
rwh															

Field	Bits	Type	Description
T12CV	15:0	rwh	**Timer T12 Counter Value** This register represents the 16-bit counter value of timer T12.

Notes

1. *While timer T12 is stopped, the internal clock divider is reset in order to ensure reproducible timings and delays.*
2. *The timer period, compare values, passive state selects bits and passive levels bits for both timers are written to shadow registers and not directly to the actual registers. Thus, the values for a new output signal can be programmed without disturbing the currently generated signal(s). The transfer from the shadow registers to the actual registers is enabled by setting the respective shadow transfer enable bit STEx.*
   If the transfer is enabled, the shadow registers are copied to the respective registers as soon as the associated timer reaches the value zero the next time (being cleared in edge-aligned mode or counting down from 1 in center-aligned mode). When timer T12 is operating in center-aligned mode, it will also copy the registers (if enabled by STE12) if it reaches the currently programmed period value (counting up).
   When a timer is stopped (TxR = 0), the shadow transfer takes place immediately if the corresponding bit STEx is set.
   After the transfer, the respective bit STEx is cleared automatically.

Register T12PR contains the period value for timer T12. The period value is compared to the actual counter value of T12 and the resulting counter actions depend on the defined counting rules. This register has a shadow register and the shadow transfer is controlled by bit STE12. A read action by software delivers the value which is currently used for the compare action, whereas the write action targets a shadow register. The shadow register structure allows a concurrent update of all T12-related values.

T12PR
Timer T12 Period Register **(24_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T12PV															
rwh															

Field	Bits	Type	Description
T12PV	15:0	rwh	**T12 Period Value** The value T12PV defines the counter value for T12, which leads to a period-match. On reaching this value, the timer T12 is set to zero (edge-aligned mode) or changes its count direction to down counting (center-aligned mode). *Note: T12PV = 0xFFFF shall not be used to avoid overflow conditions when using the Zero Match and Period Match Trigger signals (PM and ZM).*

In compare mode, the registers CC6xR (x = 0, 1, 2) are the actual compare registers for T12. The values stored in CC6xR are compared (all three channels in parallel) to the counter value of T12. In capture mode, the current value of the T12 counter register is captured by registers CC6xR if the corresponding capture event is detected.

CC6xR (x = 0-2)
Capture/Compare Register for Channel CC6x **(34_H+x*4_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCV															
rh															

Field	Bits	Type	Description
CCV	15:0	rh	**Channel x Capture/Compare Value** In compare mode, the bit fields CCV contain the values that are compared to the T12 counter value. In capture mode, the captured value of T12 can be read from these registers.

The registers CC6xR can only be read by software, the modification of the value is done by a shadow register transfer from register CC6xSR. The corresponding shadow registers CC6xSR can be read and written by software. In capture mode, the value of the T12 counter register can also be captured by registers CC6xSR if the selected capture event is detected (depending on the selected mode).

CC6xSR (x = 0-2)
Capture/Compare Shadow Register for Channel CC6x

(14_H+x*4_H) **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCS															
rwh															

Field	Bits	Type	Description
CCS	15:0	rwh	**Shadow Register for Channel x Capture/Compare Value** In compare mode, the contents of bit field CCS are transferred to the bit field CCV for the corresponding channel during a shadow transfer. In capture mode, the captured value of T12 can be read from these registers.

Note: The shadow registers can also be written by SW in capture mode. In this case, the HW capture event wins over the SW write if both happen in the same cycle (the SW write is discarded).

Register T12DTC controls the dead-time generation for the timer T12 compare channels. Each channel can be independently enabled/disabled for dead-time generation. If enabled, the transition from passive state to active state is delayed by the value defined by bit field DTM. The dead-time counter can only be reloaded while it is zero.

The dead time counters are clocked with the same frequency as T12. This structure allows symmetrical dead-time generation in center-aligned and in edge-aligned PWM mode. A duty cycle of 50% leads to CC6x, COUT6x switched on for: 0.5 * period - dead time.

Note: The dead-time counters are not reset by bit T12RES, but by bit DTRES.

T12DTC
Timer T12 Dead-Time Control Register **($2C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**DTR2**	**DTR1**	**DTR0**	**RES**	**DTE2**	**DTE1**	**DTE0**	**DTM**							
r	rh	rh	rh	r	rw	rw	rw	rw							

Field	Bits	Type	Description
DTM	7:0	rw	**Dead-Time** Bit field DTM determines the programmable delay between switching from the passive state to the active state of the selected outputs. The switching from the active state to the passive state is not delayed.
DTE0, **DTE1,** **DTE2**	8, 9, 10	rw	**Dead-Time Enable Bits** Bits DTE0..DTE2 enable and disable the dead-time generation for each compare channel (0, 1, 2) of timer T12. 0_B Dead-time generation is disabled. The corresponding outputs switch from the passive state to the active state (according to the actual compare status) without any delay. 1_B Dead-time generation is enabled. The corresponding outputs switch from the passive state to the active state (according to the compare status) with the delay programmed in bit field DTM.
DTR0, **DTR1,** **DTR2**	12, 13, 14	rh	**Dead-Time Run Indication Bits** Bits DTR0..DTR2 indicate the status of the dead-time generation for each compare channel (0, 1, 2) of timer T12. 0_B The value of the corresponding dead-time counter channel is 0. 1_B The value of the corresponding dead-time counter channel is not 0.
RES	11, 15	r	**Reserved** Returns 0 if read; should be written with 0.

Notes

1. *The dead-time counters are clocked with the same frequency as T12.*
   This structure allows symmetrical dead-time generation in center-aligned and in edge-aligned PWM mode.
   *A duty cycle of 50% leads to CC6x, COUT6x switched on for: 0.5 * period - dead-time.*
2. *The dead-time counters are not reset by bit T12RES, but by bit DTRES.*

18.11.3 Timer13 – Related Registers

The generation of the patterns for a single channel pulse width modulation (PWM) is based on timer T13. The registers related to timer T13 can be concurrently updated (with well-defined conditions) in order to ensure consistency of the PWM signal. T13 can be synchronized to several timer T12 events.

Timer T13 supports only compare mode on its compare channel CC63.

Register T13 represents the counting value of timer T13. It can only be written while the timer T13 is stopped. Write actions while T13 is running are not taken into account. Register T13 can always be read by software.

Timer T13 supports only edge-aligned mode (counting up).

T13
Timer T13 Counter Register **($7C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13CV															
rwh															

Field	Bits	Type	Description
T13CV	15:0	rwh	**Timer T13 Counter Value** This register represents the 16-bit counter value of timer T13.

Note: While timer T13 is stopped, the internal clock divider is reset in order to ensure reproducible timings and delays.

Register T13PR contains the period value for timer T13. The period value is compared to the actual counter value of T13 and the resulting counter actions depend on the defined counting rules. This register has a shadow register and the shadow transfer is controlled by bit STE13. A read action by software delivers the value which is currently used for the compare action, whereas the write action targets a shadow register. The shadow register structure allows a concurrent update of all T13-related values.

T13PR
Timer T13 Period Register **(28_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13PV															
rwh															

Field	Bits	Type	Description
T13PV	15:0	rwh	**T13 Period Value** The value T13PV defines the counter value for T13, which leads to a period-match. On reaching this value, the timer T13 is set to zero.

Register CC63R is the actual compare register for T13. The value stored in CC63R is compared to the counter value of T13. The State Bit CC63ST is located in register CMPSTAT.

CC63R
Capture/Compare Register for Channel CC63 (00_H) Reset Value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCV															
rh															

Field	Bits	Type	Description
CCV	15:0	rh	**Channel CC63 Compare Value** The bit field CCV contains the value that is compared to the T13 counter value.

The register CC63R can only be read by software and the modification of the value is done by a shadow register transfer from register CC63SR. The corresponding shadow register CC63SR can be read and written by software.

CC63SR
Capture/Compare Shadow Register for Channel CC63
(20_H) Reset Value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCS															
rw															

Field	Bits	Type	Description
CCS	15:0	rw	**Shadow Register for Channel CC63 Compare Value** The contents of bit field CCS are transferred to the bit field CCV during a shadow transfer.

18.11.4 Capture/Compare Control Registers

The Compare State Register CMPSTAT contains status bits monitoring the current capture and compare state, and control bits defining the active/passive state of the compare channels.

CMPSTAT
Compare State Register **(80_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13IM	**COUT 63PS**	**COUT 62PS**	**CC62P S**	**COUT 61PS**	**CC61P S**	**COUT 60PS**	**CC60P S**	**RES**	**CC63S T**	**CCPO S2**	**CCPO S1**	**CCPO S0**	**CC62S T**	**CC61S T**	**CC60S T**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	r	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
CC60ST, **CC61ST,** **CC62ST,** **CC63ST**	0, 1, 2, 6	rh	**Capture/Compare State Bits (x = 0, 1, 2, 3)** Bits CC6xST monitor the state of the capture/compare channels. Bits CC6xST are related to T12; bit CC63ST is related to T13. These bits are set and reset according to the T12 and T13 switching rules. 0_B In compare mode, the timer count is less than the compare value. In capture mode, the selected edge has not yet been detected since the bit has been reset by software the last time. 1_B In compare mode, the counter value is greater than or equal to the compare value. In capture mode, the selected edge has been detected.
CCPOS0, **CCPOS1,** **CCPOS2**	3, 4, 5	rh	**Sampled Hall Pattern Bits (x = 0, 1, 2)** Bits CCPSOx indicate the value of the input Hall pattern that has been compared to the current and expected value. The value is sampled when the event hcrdy (Hall compare ready) occurs. 0_B The input CCPOSx has been sampled as 0. 1_B The input CCPOSx has been sampled as 1.
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.

Field	Bits	Type	Description
CC60PS, **CC61PS,** **CC62PS,** **COUT60PS,** **COUT61PS,** **COUT62PS,** **COUT63PS**	8, 10, 12, 9, 11, 13, 14	rwh	**Passive State Select for Compare Outputs** Bits CC6xPS (x = 0, 1, 2), COUT6xPS (x = 0, 1, 2, 3) select the state of the corresponding compare channel, which is considered to be the passive state. During the passive state, the passive level (defined in register PSLR) is driven by the output pin. Bits CC6xPS, COUT6xPS (x = 0, 1, 2) are related to T12, bit COUT63PS is related to T13. These bits have shadow bits and are updated in parallel to the capture/compare registers of T12 and T13, respectively. A read action targets the actually used values, whereas a write action targets the shadow bits. In capture mode, these bits are not used. 0_B The corresponding compare output drives passive level while CC6xST is 0. 1_B The corresponding compare output drives passive level while CC6xST is 1.
T13IM	15	rwh	**T13 Inverted Modulation** Bit T13IM inverts the T13 signal for the modulation of the CC6x and COUT6x (x = 0, 1, 2) signals. This bit has a shadow bit and is updated in parallel to the compare and period registers of T13. A read action targets the actually used values, whereas a write action targets the shadow bit. 0_B T13 output is not inverted. 1_B T13 output is inverted for further modulation.

The Compare Status Modification Register CMPMODIF provides software-control (independent set and clear conditions) for the channel state bits CC6xST. This feature enables the user to individually change the status of the output lines by software, for example when the corresponding compare timer is stopped.

CMPMODIF
Compare State Modification Register **(10_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**MCC63R**	**RES**			**MCC62R**	**MCC61R**	**MCC60R**	**RES**	**MCC63S**	**RES**			**MCC62S**	**MCC61S**	**MCC60S**
r	w	r			w	w	w	r	w	r			w	w	w

Field	Bits	Type	Description
MCC60S, **MCC61S,** **MCC62S,** **MCC63S**	0, 1, 2, 6	w	**Capture/Compare Status Modification Bits (Set) (x = 0, 1, 2, 3)** These bits are used to set the corresponding CC6xST bits by software. This feature allows the user to individually change the status of the output lines by software, e.g. when the corresponding compare timer is stopped. This allows a bit manipulation of CC6xST-bits by a single data write action. Functionality see **Table 132**.

Field	Bits	Type	Description
RES	5:3, 7	r	**Reserved** Returns 0 if read; should be written with 0.
MCC60R, MCC61R, MCC62R, MCC63R	8, 9, 10, 14	w	**Capture/Compare Status Modification Bits (Reset) (x = 0, 1, 2, 3)** These bits are used to reset the corresponding CC6xST bits by software. This feature allows the user to individually change the status of the output lines by software, e.g. when the corresponding compare timer is stopped. This allows a bit manipulation of CC6xST-bits by a single data write action. Functionality see **Table 132**.
RES	13:11, 15	r	**Reserved** Returns 0 if read; should be written with 0.

Table 132 Capture/Compare Status Modification Bits (Set and Reset)

Field	Bits	Description
MCC60S, MCC61S, MCC62S, MCC63S	0 1 2 6	The following functionality of a write access to bits concerning the same capture/compare state bit is provided (x = 0, 1, 2, 3): MCC6xR, MCC6xS = 00_B Bit CC6xST is not changed. 01_B Bit CC6xST is set. 10_B Bit CC6xST is reset. 11_B Reserved (toggle)
MCC60R, MCC61R, MCC62R, MCC63R	8 9 10 14	

Register TCTR0 controls the basic functionality of both timers T12 and T13.

Note: A write action to the bit fields T12CLK or T12PRE is only taken into account while the timer T12 is not running (T12R = 0). A write action to the bit fields T13CLK or T13PRE is only taken into account while the timer T13 is not running (T13R = 0).

TCTR0
Timer Control Register 0 **(30_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		STE13	T13R	T13PRE	T13CLK			CTM	CDIR	STE12	T12R	T12PRE	T12CLK		
r		rh	rh	rw	rw			rw	rh	rh	rh	rw	rw		

Field	Bits	Type	Description
T12CLK	2:0	rw	**Timer T12 Input Clock Select** Selects the input clock for timer T12 which is derived from the peripheral clock according to the equation $f_{T12} = f_{CCU} / 2^{<T12CLK>}$. 000_B $f_{T12} = f_{CCU}$ 001_B $f_{T12} = f_{CCU} / 2$ 010_B $f_{T12} = f_{CCU} / 4$ 011_B $f_{T12} = f_{CCU} / 8$ 100_B $f_{T12} = f_{CCU} / 16$ 101_B $f_{T12} = f_{CCU} / 32$ 110_B $f_{T12} = f_{CCU} / 64$ 111_B $f_{T12} = f_{CCU} / 128$
T12PRE	3	rw	**Timer T12 Prescaler Bit** In order to support higher clock frequencies, an additional prescaler factor of 1/256 can be enabled for the prescaler for T12. 0_B The additional prescaler for T12 is disabled. 1_B The additional prescaler for T12 is enabled.
T12R	4	rh	**Timer T12 Run Bit** T12R starts and stops timer T12. It is set/reset by software by setting bits T12RS or T12RR, or it is reset by hardware according to the function defined by bit field T12SSC. A concurrent set/reset action on T12R (from T12SSC, T12RR or T12RS) will have no effect. The bit T12R will remain unchanged. 0_B Timer T12 is stopped. 1_B Timer T12 is running.

Field	Bits	Type	Description
STE12	5	rh	**Timer T12 Shadow Transfer Enable** Bit STE12 enables or disables the shadow transfer of the T12 period value, the compare values and passive state select bits and levels from their shadow registers to the actual registers if a T12 shadow transfer event is detected. Bit STE12 is cleared by hardware after the shadow transfer. A T12 shadow transfer event is a period-match while counting up or a one-match while counting down. 0_B The shadow register transfer is disabled. 1_B The shadow register transfer is enabled.
CDIR	6	rh	**Count Direction of Timer T12** This bit is set/reset according to the counting rules of T12. 0_B T12 counts up. 1_B T12 counts down.
CTM	7	rw	**T12 Operating Mode** 0_B Edge-aligned Mode: T12 always counts up and continues counting from zero after reaching the period value. 1_B Center-aligned Mode: T12 counts down after detecting a period-match and counts up after detecting a one-match.
T13CLK	10:8	rw	**Timer T13 Input Clock Select** Selects the input clock for timer T13 which is derived from the peripheral clock according to the equation $f_{T13} = f_{CCU}/2^{<T13CLK>}$. 000_B $f_{T13} = f_{CCU}$ 001_B $f_{T13} = f_{CCU} / 2$ 010_B $f_{T13} = f_{CCU} / 4$ 011_B $f_{T13} = f_{CCU} / 8$ 100_B $f_{T13} = f_{CCU} / 16$ 101_B $f_{T13} = f_{CCU} / 32$ 110_B $f_{T13} = f_{CCU} / 64$ 111_B $f_{T13} = f_{CCU} / 128$
T13PRE	11	rw	**Timer T13 Prescaler Bit** In order to support higher clock frequencies, an additional prescaler factor of 1/256 can be enabled for the prescaler for T13. 0_B The additional prescaler for T13 is disabled. 1_B The additional prescaler for T13 is enabled.
T13R	12	rh	**Timer T13 Run Bit** T13R starts and stops timer T13. It is set/reset by software by setting bits T13RS or T13RR or it is set/reset by hardware according to the function defined by bit fields T13SSC, T13TEC and T13TED. A concurrent set/reset action on T13R (from T13SSC, T13TEC, T13RR or T13RS) will have no effect. The bit T13R will remain unchanged. 0_B Timer T13 is stopped. 1_B Timer T13 is running.

Field	Bits	Type	Description
STE13	13	rh	**Timer T13 Shadow Transfer Enable** Bit STE13 enables or disables the shadow transfer of the T13 period value, the compare value and passive state select bit and level from their shadow registers to the actual registers if a T13 shadow transfer event is detected. Bit STE13 is cleared by hardware after the shadow transfer. A T13 shadow transfer event is a period-match. 0_B The shadow register transfer is disabled. 1_B The shadow register transfer is enabled.
RES	15:14	r	**Reserved** Returns 0 if read; should be written with 0.

Register TCTR2 controls the single-shot and the synchronization functionality of both timers T12 and T13. Both timers can run in single-shot mode. In this mode, they stop their counting sequence automatically after one counting period with a count value of zero. The single-shot mode and the synchronization feature of T13 to T12 allow the generation of events with a programmable delay after well-defined PWM actions of T12. For example, this feature can be used to trigger AD conversions, after a specified delay (to avoid problems due to switching noise), synchronously to a PWM event.

TCTR2
Timer Control Register 2 **(58_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				T13RSEL		T12RSEL		RES	T13TED		T13TEC			T13SSC	T12SSC
r				rw		rw		r	rw		rw			rw	rw

Field	Bits	Type	Description
T12SSC	0	rw	**Timer T12 Single Shot Control** This bit controls the single shot-mode of T12. 0_B The single-shot mode is disabled, no hardware action on T12R. 1_B The single shot mode is enabled, the bit T12R is reset by hardware if: - T12 reaches its period value in edge-aligned mode - T12 reaches the value 1 while down counting in center-aligned mode. In parallel to the reset action of bit T12R, the bits CC6xST (x = 0, 1, 2) are reset.
T13SSC	1	rw	**Timer T13 Single Shot Control** This bit controls the single shot-mode of T13. 0_B No hardware action on T13R 1_B The single-shot mode is enabled, the bit T13R is reset by hardware if T13 reaches its period value. In parallel to the reset action of bit T13R, the bit CC63ST is reset.
T13TEC	4:2	rw	**T13 Trigger Event Control** Bit field T13TEC selects the trigger event to start T13 (automatic set of T13R for synchronization to T12 compare signals) according to following combinations: 000_B no action 001_B set T13R on a T12 compare event on channel 0 010_B set T13R on a T12 compare event on channel 1 011_B set T13R on a T12 compare event on channel 2 100_B set T13R on any T12 compare event on the channels 0, 1, or 2 101_B set T13R upon a period-match of T12 110_B set T13R upon a zero-match of T12 (while counting up) 111_B set T13R on any edge of inputs CCPOSx

Field	Bits	Type	Description
T13TED	6:5	rw	**Timer T13 Trigger Event Direction** Bit field T13TED delivers additional information to control the automatic set of bit T13R in the case that the trigger action defined by T13TEC is detected. 00_B no action 01_B while T12 is counting up 10_B while T12 is counting down 11_B independent on the count direction of T12
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
T12RSEL	9:8	rw	**Timer T12 External Run Selection** Bit field T12RSEL defines the event of signal T12HR that can set the run bit T12R by hardware. 00_B The external setting of T12R is disabled. 01_B Bit T12R is set if a rising edge of signal T12HR is detected. 10_B Bit T12R is set if a falling edge of signal T12HR is detected. 11_B Bit T12R is set if an edge of signal T12HR is detected.
T13RSEL	11:10	rw	**Timer T13 External Run Selection** Bit field T13RSEL defines the event of signal T13HR that can set the run bit T13R by hardware. 00_B The external setting of T13R is disabled. 01_B Bit T13R is set if a rising edge of signal T13HR is detected. 10_B Bit T13R is set if a falling edge of signal T13HR is detected. 11_B Bit T13R is set if an edge of signal T13HR is detected.
RES	15:12	r	**Reserved** Returns 0 if read; should be written with 0.

Example

If the timer T13 is intended to start at any compare event on T12 (T13TEC = 100_B), the trigger event direction can be programmed to:

- counting up >> a T12 channel 0, 1, 2 compare match triggers T13R only while T12 is counting up
- counting down >> a T12 channel 0, 1, 2 compare match triggers T13R only while T12 is counting down
- independent from bit CDIR >> each T12 channel 0, 1, 2 compare match triggers T13R

The timer count direction is taken from the value of bit CDIR. As a result, if T12 is running in edge-aligned mode (counting up only), T13 can only be started automatically if bit field T13TED = 01_B or 11_B.

Register TCTR4 provides software-control (independent set and clear conditions) for the run bits T12R and T13R. Furthermore, the timers can be reset (while running) and bits STE12 and STE13 can be controlled by software. Reading these bits always returns 0.

TCTR4
Timer Control Register 4 **(04_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13STD	T13STR	T13CNT	RES		T13RES	T13RS	T13RR	T12STD	T12STR	T12CNT	RES	DTRES	T12RES	T12RS	T12RR
w	w	w	r		w	w	w	w	w	w	r	w	w	w	w

Field	Bits	Type	Description
T12RR	0	w	**Timer T12 Run Reset** Setting this bit resets the T12R bit. 0_B T12R is not influenced. 1_B T12R is cleared, T12 stops counting.
T12RS	1	w	**Timer T12 Run Set** Setting this bit sets the T12R bit. 0_B T12R is not influenced. 1_B T12R is set, T12 counts.
T12RES	2	w	**Timer T12 Reset** 0_B No effect on T12. 1_B The T12 counter register is reset to zero. The switching of the output signals is according to the switching rules. Setting of T12RES has no impact on bit T12R.
DTRES	3	w	**Dead-Time Counter Reset** 0_B No effect on the dead-time counters. 1_B The three dead-time counter channels are reset to zero.
T12CNT	5	w	**Timer T12 Count Event** 0_B No action 1_B If enabled (PISEL2), timer T12 counts one step.
T12STR	6	w	**Timer T12 Shadow Transfer Request** 0_B No action 1_B STE12 is set, enabling the shadow transfer.
T12STD	7	w	**Timer T12 Shadow Transfer Disable** 0_B No action 1_B STE12 is reset without triggering the shadow transfer.
RES	4	r	**Reserved** Returns 0 if read; should be written with 0.
T13RR	8	w	**Timer T13 Run Reset** Setting this bit resets the T13R bit. 0_B T13R is not influenced. 1_B T13R is cleared, T13 stops counting.

Field	Bits	Type	Description
T13RS	9	w	**Timer T13 Run Set** Setting this bit sets the T13R bit. 0_B T13R is not influenced. 1_B T13R is set, T13 counts.
T13RES	10	w	**Timer T13 Reset** 0_B No effect on T13. 1_B The T13 counter register is reset to zero. The switching of the output signals is according to the switching rules. Setting of T13RES has no impact on bit T13R.
T13CNT	13	w	**Timer T13 Count Event** 0_B No action 1_B If enabled (PISEL2), timer T13 counts one step.
T13STR	14	w	**Timer T13 Shadow Transfer Request** 0_B No action 1_B STE13 is set, enabling the shadow transfer.
T13STD	15	w	**Timer T13 Shadow Transfer Disable** 0_B No action 1_B STE13 is reset without triggering the shadow transfer.
RES	12:11	r	**Reserved** Returns 0 if read; should be written with 0.

Note: A simultaneous write of a 1 to bits which set and reset the same bit will trigger no action. The corresponding bit will remain unchanged.

18.11.5 Global Modulation Control Registers

Register MODCTR contains control bits enabling the modulation of the corresponding output signal by PWM pattern generated by the timers T12 and T13. Furthermore, the multi-channel mode can be enabled as additional modulation source for the output signals.

MODCTR
Modulation Control Register **($5C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECT13O	RES	T13MODEN						MCMEN	RES	T12MODEN					
rw	r	rw						rw	r	rw					

Field	Bits	Type	Description
T12MODEN	5:0	rw	**T12 Modulation Enable** Setting these bits enables the modulation of the corresponding compare channel by a PWM pattern generated by timer T12. The bit positions are corresponding to the following output signals: Bit 0: modulation of CC60 Bit 1: modulation of COUT60 Bit 2: modulation of CC61 Bit 3: modulation of COUT61 Bit 4: modulation of CC62 Bit 5: modulation of COUT62 The enable feature of the modulation is defined as follows: 0_B The modulation of the corresponding output signal by a T12 PWM pattern is disabled. 1_B The modulation of the corresponding output signal by a T12 PWM pattern is enabled.
MCMEN	7	rw	**Multi-Channel Mode Enable** 0_B The modulation of the corresponding output signal by a multi-channel pattern according to bit field MCMOUT is disabled. 1_B The modulation of the corresponding output signal by a multi-channel pattern according to bit field MCMOUT is enabled.
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.

Field	Bits	Type	Description
T13MODEN	13:8	rw	**T13 Modulation Enable** Setting these bits enables the modulation of the corresponding compare channel by a PWM pattern generated by timer T13. The bit positions are corresponding to the following output signals: Bit 0: modulation of CC60 Bit 1: modulation of COUT60 Bit 2: modulation of CC61 Bit 3: modulation of COUT61 Bit 4: modulation of CC62 Bit 5: modulation of COUT62 The enable feature of the modulation is defined as follows: 0_B The modulation of the corresponding output signal by a T13 PWM pattern is disabled. 1_B The modulation of the corresponding output signal by a T13 PWM pattern is enabled.
ECT13O	15	rw	**Enable Compare Timer T13 Output** 0_B The alternate output function COUT63 is disabled. 1_B The alternate output function COUT63 is enabled for the PWM signal generated by T13.
RES	14	r	**Reserved** Returns 0 if read; should be written with 0.

The register TRPCTR controls the trap functionality. It contains independent enable bits for each output signal and control bits to select the behavior in case of a trap condition. The trap condition is a low-level on the $\overline{\text{CTRAP}}$ input pin, which is monitored (inverted level) by bit IS.TRPF. While TRPF = 1 (trap input active), the trap state bit IS.TRPS is set to 1.

TRPCTR
Trap Control Register **(60_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRPPEN	**TRPEN13**	**TRPEN**						**RES**					**TRPM2**	**TRPM1**	**TRPM0**
rw	rw	rw						r					rw	rw	rw

Field	Bits	Type	Description
TRPM0, TRPM1	0, 1	rw	**Trap Mode Control Bits 1, 0** These two bits define the behavior of the selected outputs when leaving the trap state after the trap condition has become inactive again. A synchronization to the timer driving the PWM pattern permits to avoid unintended short pulses when leaving the trap state. See **Table 133**.
TRPM2	2	rw	**Trap Mode Control Bit 2** 0_B The trap state can be left (return to normal operation = bit TRPS = 0) as soon as the input $\overline{\text{CTRAP}}$ becomes inactive. Bit TRPF is automatically cleared by hardware if the input pin $\overline{\text{CTRAP}}$ becomes 1. Bit TRPS is automatically cleared by hardware if bit TRPF is 0 and if the synchronization condition (according to TRPM0,1) is detected. 1_B The trap state can be left (return to normal operation = bit TRPS = 0) as soon as bit TRPF is reset by software after the input $\overline{\text{CTRAP}}$ becomes inactive (TRPF is not cleared by hardware). Bit TRPS is automatically cleared by hardware if bit TRPF = 0 and if the synchronization condition (according to TRPM0,1) is detected.
RES	7:3	r	**Reserved** Returns 0 if read; should be written with 0.

Field	Bits	Type	Description
TRPEN	13:8	rw	**Trap Enable Control** Setting these bits enables the trap functionality for the following corresponding output signals: Bit 0: trap functionality of CC60 Bit 1: trap functionality of COUT60 Bit 2: trap functionality of CC61 Bit 3: trap functionality of COUT61 Bit 4: trap functionality of CC62 Bit 5: trap functionality of COUT62 The enable feature of the trap functionality is defined as follows: 0_B The trap functionality of the corresponding output signal is disabled. The output state is independent from bit TRPS. 1_B The trap functionality of the corresponding output signal is enabled. The output is set to the passive state while TRPS = 1.
TRPEN13	14	rw	**Trap Enable Control for Timer T13** 0_B The trap functionality for T13 is disabled. Timer T13 (if selected and enabled) provides PWM functionality even while TRPS = 1. 1_B The trap functionality for T13 is enabled. The timer T13 PWM output signal is set to the passive state while TRPS = 1.
TRPPEN	15	rw	**Trap Pin Enable** 0_B The trap functionality based on the input pin $\overline{\text{CTRAP}}$ is disabled. A trap can only be generated by software by setting bit TRPF. 1_B The trap functionality based on the input pin $\overline{\text{CTRAP}}$ is enabled. A trap can be generated by software by setting bit TRPF or by $\overline{\text{CTRAP}}$ = 0.

Table 133 Trap Mode Control Bits 1, 0

Field	Bits	Description
TRPM0, TRPM1	0 1	A synchronization to the timer driving the PWM pattern permits to avoid unintended short pulses when leaving the trap state. The combination (TRPM1, TRPM0) leads to: 00_B The trap state is left (return to normal operation according to TRPM2) when a zero-match of T12 (while counting up) is detected (synchronization to T12). 01_B The trap state is left (return to normal operation according to TRPM2) when a zero-match of T13 is detected (synchronization to T13). 10_B reserved 11_B The trap state is left (return to normal operation according to TRPM2) immediately without any synchronization to T12 or T13.

Register PSLR defines the passive state level driven by the output pins of the module. The passive state level is the value that is driven by the port pin during the passive state of the output. During the active state, the corresponding output pin drives the active state level, which is the inverted passive state level. The passive state level permits the adaptation of the driven output levels to the driver polarity (inverted, not inverted) of the connected power stage. The bits in this register have shadow bit fields to permit a concurrent update of all PWM-related parameters (bit field PSL is updated with T12_ST, whereas PSL63 is updated with T13_ST). The actually used values can be read (attribute “rh”), whereas the shadow bits can only be written (attribute “w”).

PSLR
Passive State Level Register **(50_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
PSL63	**RES**	**PSL**					
rwh	r	rwh					

Field	Bits	Type	Description
PSL	5:0	rwh	**Compare Outputs Passive State Level** The bits of this bit field define the passive level driven by the module outputs during the passive state. The bit positions are: Bit 0: passive level for output CC60 Bit 1: passive level for output COUT60 Bit 2: passive level for output CC61 Bit 3: passive level for output COUT61 Bit 4: passive level for output CC62 Bit 5: passive level for output COUT62 The value of each bit position is defined as: 0_B The passive level is 0. 1_B The passive level is 1.
PSL63	7	rwh	**Passive State Level of Output COUT63** This bit field defines the passive level of the output pin COUT63. 0_B The passive level is 0. 1_B The passive level is 1.
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.

Notes

1. *Bit field PSL has a shadow register to allow for updates without undesired pulses on the output lines. The bits are updated with the T12 shadow transfer. A read action targets the actually used values, whereas a write action targets the shadow bits.*
2. *Bit field PSL63 has a shadow register to allow for updates without undesired pulses on the output line. The bit is updated with the T13 shadow transfer. A read action targets the actually used values, whereas a write action targets the shadow bits.*

18.11.6 Multi-Channel Modulation Control Registers

Register MCMOUTS contains bits used as pattern input for the multi-channel mode and the Hall mode. This register is a shadow register (that can be read and written) for register MCMOUT, which indicates the currently active signals.

MCMOUTS
Multi-Channel Mode Output Shadow Register **(08_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STRHP	RES	CURHS			EXPHS			STRMCM	RES	MCMPS					
w	r	rw			rw			w	r	rw					

Field	Bits	Type	Description
MCMPS	5:0	rw	**Multi-Channel PWM Pattern Shadow** Bit field MCMPS is the shadow bit field for bit field MCMP. The multi-channel shadow transfer is triggered according to the transfer conditions defined by register MCMCTR.
STRMCM	7	w	**Shadow Transfer Request for MCMPS** Setting this bit during a write action leads to an immediate update of bit field MCMP by the value written to bit field MCMPS. This functionality permits an update triggered by software. When read, this bit always delivers 0. 0_B Bit field MCMP is updated according to the defined hardware action. The write access to bit field MCMPS does not modify bit field MCMP. 1_B Bit field MCMP is updated by the value written to bit field MCMPS.
RES	6	r	**Reserved** Returns 0 if read; should be written with 0.
EXPHS	10:8	rw	**Expected Hall Pattern Shadow** Bit field EXPHS is the shadow bit field for bit field EXPH. The bit field is transferred to bit field EXPH if an edge on the hall input pins CCPOSx (x = 0, 1, 2) is detected.
CURHS	13:11	rw	**Current Hall Pattern Shadow** Bit field CURHS is the shadow bit field for bit field CURH. The bit field is transferred to bit field CURH if an edge on the hall input pins CCPOSx (x = 0, 1, 2) is detected.

Field	Bits	Type	Description
STRHP	15	w	**Shadow Transfer Request for the Hall Pattern** Setting these bits during a write action leads to an immediate update of bit fields CURH and EXPH by the value written to bit fields CURHS and EXPH. This functionality permits an update triggered by software. When read, this bit always delivers 0. 0_B The bit fields CURH and EXPH are updated according to the defined hardware action. The write access to bit fields CURHS and EXPHS does not modify the bit fields CURH and EXPH. 1_B The bit fields CURH and EXPH are updated by the value written to the bit fields CURHS and EXPHS.
RES	14	r	**Reserved** Returns 0 if read; should be written with 0.

Register MCMOUT shows the multi-channel control bits that are currently used. Register MCMOUT is defined as follows:

MCMOUT
Multi-Channel Mode Output Register **(64_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		**CURH**			**EXPH**			**RES**	**R**	**MCMP**					
r		rh			rh			r	rh	rh					

Field	Bits	Type	Description
MCMP	5:0	rh	**Multi-Channel PWM Pattern** Bit field MCMP is written by a shadow transfer from bit field MCMPS. It contains the output pattern for the multi-channel mode. If this mode is enabled by bit MCMEN in register MODCTR, the output state of the following output signal can be modified: Bit 0: multi-channel state for output CC60 Bit 1: multi-channel state for output COUT60 Bit 2: multi-channel state for output CC61 Bit 3: multi-channel state for output COUT61 Bit 4: multi-channel state for output CC62 Bit 5: multi-channel state for output COUT62 The multi-channel patterns can set the related output to the passive state. While IDLE = 1, bit field MCMP is cleared. 0_B The output is set to the passive state. The PWM generated by T12 or T13 is not taken into account. 1_B The output can deliver the PWM generated by T12 or T13 (according to register MODCTR).

Field	Bits	Type	Description
R	6	rh	**Reminder Flag** This reminder flag indicates that the shadow transfer from bit field MCMPS to MCMP has been requested by the selected trigger source. This bit is cleared when the shadow transfer takes place and while MCMEN = 0. 0_B Currently, no shadow transfer from MCMPS to MCMP is requested. 1_B A shadow transfer from MCMPS to MCMP has been requested by the selected trigger source, but it has not yet been executed, because the selected synchronization condition has not yet occurred.
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.
EXPH	10:8	rh	**Expected Hall Pattern** Bit field EXPH is written by a shadow transfer from bit field EXPHS. The contents are compared after every detected edge at the hall input pins with the pattern at the hall input pins in order to detect the occurrence of the next desired (= expected) hall pattern or a wrong pattern. If the current hall pattern at the hall input pins is equal to the bit field EXPH, bit CHE (correct hall event) is set and an interrupt request is generated (if enabled by bit ENCHE). If the current hall pattern at the hall input pins is not equal to the bit fields CURH or EXPH, bit WHE (wrong hall event) is set and an interrupt request is generated (if enabled by bit ENWHE).
CURH	13:11	rh	**Current Hall Pattern** Bit field CURH is written by a shadow transfer from bit field CURHS. The contents are compared after every detected edge at the hall input pins with the pattern at the hall input pins in order to detect the occurrence of the next desired (= expected) hall pattern or a wrong pattern. If the current hall input pattern is equal to bit field CURH, the detected edge at the hall input pins has been an invalid transition (e.g. a spike).
RES	15:14	r	**Reserved** Returns 0 if read; should be written with 0.

Note: The bits in the bit fields EXPH and CURH correspond to the hall patterns at the input pins CCPOSx (x = 0, 1, 2) in the following order (EXPH.2, EXPH.1, EXPH.0), (CURH.2, CURH.1, CURH.0), (CCPOS2, CCPOS.1, CCPOS0).

Register MCMCTR contains control bits for the multi-channel functionality.

MCMCTR
Multi-Channel Mode Control Register **(54_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					STE13U	STE12D	STE12U	RES	RES	SWSYN		RES	SWSEL		
r					rw	rw	rw	r	r	rw		r	rw		

Field	Bits	Type	Description
SWSEL	2:0	rw	**Switching Selection** Bit field SWSEL selects one of the following trigger request sources (next multi-channel event) for the shadow transfer from MCMPS to MCMP. The trigger request is stored in the reminder flag R until the shadow transfer is done and flag R is cleared automatically with the shadow transfer. The shadow transfer takes place synchronously with an event selected in bit field SWSYN. 000_B no trigger request will be generated 001_B correct hall pattern on CCPOSx detected 010_B T13 period-match detected (while counting up) 011_B T12 one-match (while counting down) 100_B T12 channel 1 compare-match detected (phase delay function) 101_B T12 period match detected (while counting up) else reserved, no trigger request will be generated
SWSYN	5:4	rw	**Switching Synchronization** Bit field SWSYN triggers the shadow transfer between MCMPS and MCMP if it has been requested before (flag R set by an event selected by SWSEL). This feature permits the synchronization of the outputs to the PWM source, that is used for modulation (T12 or T13). 00_B direct; the trigger event directly causes the shadow transfer 01_B T13 zero-match triggers the shadow transfer 10_B a T12 zero-match (while counting up) triggers the shadow transfer 11_B reserved; no action
RES	3, 6, 7	r	**Reserved** Returns 0 if read; should be written with 0.
STE12U	8	rw	**Shadow Transfer Enable for T12 Upcounting** This bit enables the shadow transfer T12_ST if flag MCMOUT.R is set or becomes set while a T12 period match is detected while counting up. 0_B No action 1_B The T12_ST shadow transfer mechanism is enabled if MCMEN = 1.

Field	Bits	Type	Description
STE12D	9	rw	**Shadow Transfer Enable for T12 Downcounting** This bit enables the shadow transfer T12_ST if flag MCMOUT.R is set or becomes set while a T12 one match is detected while counting down. 0_B No action 1_B The T12_ST shadow transfer mechanism is enabled if MCMEN = 1.
STE13U	10	rw	**Shadow Transfer Enable for T13 Upcounting** This bit enables the shadow transfer T13_ST if flag MCMOUT.R is set or becomes set while a T13 period match is detected. 0_B No action 1_B The T13_ST shadow transfer mechanism is enabled if MCMEN = 1.
RES	15:11	r	**Reserved** Returns 0 if read; should be written with 0.

18.11.7 Interrupt Control Registers

Register IS contains the individual interrupt request bits. This register can only be read; write actions have no impact on the contents of this register. The software can set or reset the bits individually by writing to the registers ISS (to set the bits) or to register ISR (to reset the bits).

The interrupt generation is independent from the value of the bits in register IS, e.g. the interrupt will be generated (if enabled) even if the corresponding bit is already set. The trigger for an interrupt generation is the detection of a set condition (by HW or SW) for the corresponding bit in register IS.

In compare mode (and hall mode), the timer-related interrupts are only generated while the timer is running (T1xR = 1). In capture mode, the capture interrupts are also generated while the timer T12 is stopped.

Note: Not all bits in register IS can generate an interrupt. Other status bits have been added, that have a similar structure for their set and clear actions. It is recommended that SW checks the interrupt bits bit-wisely (instead of common OR over the bits).

IS
Capture/Compare Interrupt Status Register (68_H) Reset Value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STR	**IDLE**	**WHE**	**CHE**	**TRPS**	**TRPF**	**T13PM**	**T13CM**	**T12PM**	**T12OM**	**ICC62F**	**ICC62R**	**ICC61F**	**ICC61R**	**ICC60F**	**ICC60R**
rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
ICC60R, ICC61R, ICC62R	0, 2, 4	rh	**Capture, Compare-Match Rising Edge Flag (x = 0, 1, 2)** In compare mode, a compare-match has been detected while T12 was counting up. In capture mode, a rising edge has been detected at the input CC6x. 0_B The event has not yet occurred since this bit has been reset for the last time. 1_B The event described above has been detected.
ICC60F, ICC61F, ICC62F	1, 3, 5	rh	**Capture, Compare-Match Falling Edge Flag (x = 0, 1, 2)** In compare mode, a compare-match has been detected while T12 was counting down. In capture mode, a falling edge has been detected at the input CC6x. 0_B The event has not yet occurred since this bit has been reset for the last time. 1_B The event described above has been detected.
T12OM	6	rh	**Timer T12 One-Match Flag** 0_B A timer T12 one-match (while counting down) has not yet been detected since this bit has been reset for the last time. 1_B A timer T12 one-match (while counting down) has been detected.
T12PM	7	rh	**Timer T12 Period-Match Flag** 0_B A timer T12 period-match (while counting up) has not yet been detected since this bit has been reset for the last time. 1_B A timer T12 period-match (while counting up) has been detected.

Field	Bits	Type	Description
T13CM	8	rh	**Timer T13 Compare-Match Flag** 0_B A timer T13 compare-match has not yet been detected since this bit has been reset for the last time. 1_B A timer T13 compare-match has been detected.
T13PM	9	rh	**Timer T13 Period-Match Flag** 0_B A timer T13 period-match has not yet been detected since this bit has been reset for the last time. 1_B A timer T13 period-match has been detected.
TRPF	10	rh	**Trap Flag** The trap flag TRPF will be set by hardware if TRPPEN = 1 and $\overline{\text{CTRAP}}$ = 0 or by software. If TRPM2 = 0, bit TRPF is reset by hardware if the input $\overline{\text{CTRAP}}$ becomes inactive (TRPPEN = 1). If TRPM2 = 1, bit TRPF must be reset by software in order to leave the trap state. 0_B The trap condition has not been detected. 1_B The trap condition has been detected (input $\overline{\text{CTRAP}}$ has been 0 or by software).
TRPS	11	rh	**Trap State** During the trap state, the selected outputs are set to the passive state. The logic level driven during the passive state is defined by the corresponding bit in register PSLR. Bit TRPS = 1 and TRPF = 0 can occur if the trap condition is no longer active but the selected synchronization has not yet taken place. 0_B The trap state is not active. 1_B The trap state is active. Bit TRPS is set while bit TRPF = 1. It is reset according to the mode selected in register TRPCTR.
CHE	12	rh	**Correct Hall Event** On every valid hall edge, the contents of EXPH are compared with the pattern on pin CCPOSx and if equal bit CHE is set. 0_B A transition to a correct (= expected) hall event has not yet been detected since this bit has been reset for the last time. 1_B A transition to a correct (= expected) hall event has been detected.
WHE	13	rh	**Wrong Hall Event** On every valid hall edge, the contents of EXPH are compared with the pattern on pin CCPOSx. If both comparisons (CURH and EXPH with CCPOSx) are not true, bit WHE (wrong hall event) is set. 0_B A transition to a wrong hall event (not the expected one) has not yet been detected since this bit has been reset for the last time. 1_B A transition to a wrong hall event (not the expected one) has been detected.
IDLE	14	rh	**IDLE State** This bit is set together with bit WHE (wrong hall event) and it must be reset by software. 0_B No action. 1_B Bit field MCMP is cleared and held to 0, the selected outputs are set to passive state.

Field	Bits	Type	Description
STR	15	rh	**Multi-Channel Mode Shadow Transfer Request** This bit is set when a shadow transfer from MCMOUTS to MCMOUT takes places in multi-channel mode. 0_B The shadow transfer has not yet taken place. 1_B The shadow transfer has taken place.

Register ISS contains individual interrupt request set bits to generate a CCU6 interrupt request by software. Writing a 1 sets the bit(s) in register IS at the corresponding bit position(s) and can generate an interrupt event (if available and enabled). All bit positions read as 0.

ISS
Capture/Compare Interrupt Status Set Register (4C$_H$) **Reset Value: 0000$_H$**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSTR	**SIDLE**	**SWHE**	**SCHE**	**SWHC**	**STRPF**	**ST13PM**	**ST13CM**	**ST12PM**	**ST12OM**	**SCC62F**	**SCC62R**	**SCC61F**	**SCC61R**	**SCC60F**	**SCC60R**
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
SCC60R	0	w	**Set Capture, Compare-Match Rising Edge Flag** 0_B No action 1_B Bit CC60R in register IS will be set.
SCC60F	1	w	**Set Capture, Compare-Match Falling Edge Flag** 0_B No action 1_B Bit CC60F in register IS will be set.
SCC61R	2	w	**Set Capture, Compare-Match Rising Edge Flag** 0_B No action 1_B Bit CC61R in register IS will be set.
SCC61F	3	w	**Set Capture, Compare-Match Falling Edge Flag** 0_B No action 1_B Bit CC61F in register IS will be set.
SCC62R	4	w	**Set Capture, Compare-Match Rising Edge Flag** 0_B No action 1_B Bit CC62R in register IS will be set.
SCC62F	5	w	**Set Capture, Compare-Match Falling Edge Flag** 0_B No action 1_B Bit CC62F in register IS will be set.
ST12OM	6	w	**Set Timer T12 One-Match Flag** 0_B No action 1_B Bit T12OM in register IS will be set.
ST12PM	7	w	**Set Timer T12 Period-Match Flag** 0_B No action 1_B Bit T12PM in register IS will be set.

Field	Bits	Type	Description
ST13CM	8	w	**Set Timer T13 Compare-Match Flag** 0_B No action 1_B Bit T13CM in register IS will be set.
ST13PM	9	w	**Set Timer T13 Period-Match Flag** 0_B No action 1_B Bit T13PM in register IS will be set.
STRPF	10	w	**Set Trap Flag** 0_B No action 1_B Bits TRPF and TRPS in register IS will be set.
SWHC	11	w	**Software Hall Compare** 0_B No action 1_B The Hall compare action is triggered.
SCHE	12	w	**Set Correct Hall Event Flag** 0_B No action 1_B Bit CHE in register IS will be set.
SWHE	13	w	**Set Wrong Hall Event Flag** 0_B No action 1_B Bit WHE in register IS will be set.
SIDLE	14	w	**Set IDLE Flag** 0_B No action 1_B Bit IDLE in register IS will be set.
SSTR	15	w	**Set STR Flag** 0_B No action 1_B Bit STR in register IS will be set.

Note: If the setting by hardware of the corresponding flags can lead to an interrupt, the setting by software has the same effect.

Register ISR contains bits to individually clear the interrupt event flags by software. Writing a 1 clears the bit(s) in register IS at the corresponding bit position(s). All bit positions read as 0.

ISR
Capture/Compare Interrupt Status Reset Register
($0C_H$) **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSTR	**RIDLE**	**RWHE**	**RCHE**	**RES**	**RTRPF**	**RT13PM**	**RT13CM**	**RT12PM**	**RT12OM**	**RCC62F**	**RCC62R**	**RCC61F**	**RCC61R**	**RCC60F**	**RCC60R**
w	w	w	w	r	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
RCC60R	0	w	**Reset Capture, Compare-Match Rising Edge Flag** 0_B No action 1_B Bit CC60R in register IS will be reset.
RCC60F	1	w	**Reset Capture, Compare-Match Falling Edge Flag** 0_B No action 1_B Bit CC60F in register IS will be reset.
RCC61R	2	w	**Reset Capture, Compare-Match Rising Edge Flag** 0_B No action 1_B Bit CC61R in register IS will be reset.
RCC61F	3	w	**Reset Capture, Compare-Match Falling Edge Flag** 0_B No action 1_B Bit CC61F in register IS will be reset.
RCC62R	4	w	**Reset Capture, Compare-Match Rising Edge Flag** 0_B No action 1_B Bit CC62R in register IS will be reset.
RCC62F	5	w	**Reset Capture, Compare-Match Falling Edge Flag** 0_B No action 1_B Bit CC62F in register IS will be reset.
RT12OM	6	w	**Reset Timer T12 One-Match Flag** 0_B No action 1_B Bit T12OM in register IS will be reset.
RT12PM	7	w	**Reset Timer T12 Period-Match Flag** 0_B No action 1_B Bit T12PM in register IS will be reset.
RT13CM	8	w	**Reset Timer T13 Compare-Match Flag** 0_B No action 1_B Bit T13CM in register IS will be reset.
RT13PM	9	w	**Reset Timer T13 Period-Match Flag** 0_B No action 1_B Bit T13PM in register IS will be reset.

Field	Bits	Type	Description
RTRPF	10	w	**Reset Trap Flag** 0_B No action 1_B Bit TRPF in register IS will be reset (not taken into account while input $\overline{CTRAP}$ = 0 and TRPPEN = 1).
RES	11	r	**Reserved** Returns 0 if read; should be written with 0.
RCHE	12	w	**Reset Correct Hall Event Flag** 0_B No action 1_B Bit CHE in register IS will be reset.
RWHE	13	w	**Reset Wrong Hall Event Flag** 0_B No action 1_B Bit WHE in register IS will be reset.
RIDLE	14	w	**Reset IDLE Flag** 0_B No action 1_B Bit IDLE in register IS will be reset.
RSTR	15	w	**Reset STR Flag** 0_B No action 1_B Bit STR in register IS will be reset.

Register IEN contains the interrupt enable bits and a control bit to enable the automatic idle function in the case of a wrong hall pattern.

IEN

Capture/Compare Interrupt Enable Register **(44_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENSTR	**ENIDLE**	**ENWHE**	**ENCHE**	**RES**	**ENTRPF**	**ENT13PM**	**ENT13CM**	**ENT12PM**	**ENT12OM**	**ENCC62F**	**ENCC62R**	**ENCC61F**	**ENCC61R**	**ENCC60F**	**ENCC60R**
rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
ENCC60R	0	rw	**Capture, Compare-Match Rising Edge Interrupt Enable for Channel 0** 0_B No interrupt will be generated if the set condition for bit CC60R in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CC60R in register IS occurs. The interrupt line that will be activated is selected by bit field INPCC60.

Field	Bits	Type	Description
ENCC60F	1	rw	**Capture, Compare-Match Falling Edge Interrupt Enable for Channel 0** 0_B No interrupt will be generated if the set condition for bit CC60F in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CC60F in register IS occurs. The interrupt line that will be activated is selected by bit field INPCC60.
ENCC61R	2	rw	**Capture, Compare-Match Rising Edge Interrupt Enable for Channel 1** 0_B No interrupt will be generated if the set condition for bit CC61R in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CC61R in register IS occurs. The interrupt line that will be activated is selected by bit field INPCC61.
ENCC61F	3	rw	**Capture, Compare-Match Falling Edge Interrupt Enable for Channel 1** 0_B No interrupt will be generated if the set condition for bit CC61F in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CC61F in register IS occurs. The interrupt line that will be activated is selected by bit field INPCC61.
ENCC62R	4	rw	**Capture, Compare-Match Rising Edge Interrupt Enable for Channel 2** 0_B No interrupt will be generated if the set condition for bit CC62R in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CC62R in register IS occurs. The interrupt line that will be activated is selected by bit field INPCC62.
ENCC62F	5	rw	**Capture, Compare-Match Falling Edge Interrupt Enable for Channel 2** 0_B No interrupt will be generated if the set condition for bit CC62F in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CC62F in register IS occurs. The interrupt line that will be activated is selected by bit field INPCC62.
ENT12OM	6	rw	**Enable Interrupt for T12 One-Match** 0_B No interrupt will be generated if the set condition for bit T12OM in register IS occurs. 1_B An interrupt will be generated if the set condition for bit T12OM in register IS occurs. The interrupt line that will be activated is selected by bit field INPT12.
ENT12PM	7	rw	**Enable Interrupt for T12 Period-Match** 0_B No interrupt will be generated if the set condition for bit T12PM in register IS occurs. 1_B An interrupt will be generated if the set condition for bit T12PM in register IS occurs. The interrupt line that will be activated is selected by bit field INPT12.

Field	Bits	Type	Description
ENT13CM	8	rw	**Enable Interrupt for T13 Compare-Match** 0_B No interrupt will be generated if the set condition for bit T13CM in register IS occurs. 1_B An interrupt will be generated if the set condition for bit T13CM in register IS occurs. The interrupt line that will be activated is selected by bit field INPT13.
ENT13PM	9	rw	**Enable Interrupt for T13 Period-Match** 0_B No interrupt will be generated if the set condition for bit T13PM in register IS occurs. 1_B An interrupt will be generated if the set condition for bit T13PM in register IS occurs. The interrupt line that will be activated is selected by bit field INPT13.
ENTRPF	10	rw	**Enable Interrupt for Trap Flag** 0_B No interrupt will be generated if the set condition for bit TRPF in register IS occurs. 1_B An interrupt will be generated if the set condition for bit TRPF in register IS occurs. The interrupt line that will be activated is selected by bit field INPERR.
RES	11	r	**Reserved** Returns 0 if read; should be written with 0.
ENCHE	12	rw	**Enable Interrupt for Correct Hall Event** 0_B No interrupt will be generated if the set condition for bit CHE in register IS occurs. 1_B An interrupt will be generated if the set condition for bit CHE in register IS occurs. The interrupt line that will be activated is selected by bit field INPCHE.
ENWHE	13	rw	**Enable Interrupt for Wrong Hall Event** 0_B No interrupt will be generated if the set condition for bit WHE in register IS occurs. 1_B An interrupt will be generated if the set condition for bit WHE in register IS occurs. The interrupt line that will be activated is selected by bit field INPERR.
ENIDLE	14	rw	**Enable Idle** This bit enables the automatic entering of the idle state (bit IDLE will be set) after a wrong hall event has been detected (bit WHE is set). During the idle state, the bit field MCMP is automatically cleared. 0_B The bit IDLE is not automatically set when a wrong hall event is detected. 1_B The bit IDLE is automatically set when a wrong hall event is detected.
ENSTR	15	rw	**Enable Multi-Channel Mode Shadow Transfer Interrupt** 0_B No interrupt will be generated if the set condition for bit STR in register IS occurs. 1_B An interrupt will be generated if the set condition for bit STR in register IS occurs. The interrupt line that will be activated is selected by bit field INPCHE.

Register INP contains the interrupt node pointers allowing a flexible interrupt handling. These bit fields define which service request output will be activated if the corresponding interrupt event occurs and the interrupt generation for this event is enabled.

INP
Capture/Compare Interrupt Node Pointer Register

(48_H) **Reset Value: 3940_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		INPT13		INPT12		INPERR		INPCHE		INPCC62		INPCC61		INPCC60	
r		rw		rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
INPCC60	1:0	rw	**Interrupt Node Pointer for Channel 0 Interrupts** This bit field defines the interrupt output line, which is activated due to a set condition for bit CC60R (if enabled by bit ENCC60R) or for bit CC60F (if enabled by bit ENCC60F). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.
INPCC61	3:2	rw	**Interrupt Node Pointer for Channel 1 Interrupts** This bit field defines the interrupt output line, which is activated due to a set condition for bit CC61R (if enabled by bit ENCC61R) or for bit CC61F (if enabled by bit ENCC61F). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.
INPCC62	5:4	rw	**Interrupt Node Pointer for Channel 2 Interrupts** This bit field defines the interrupt output line, which is activated due to a set condition for bit CC62R (if enabled by bit ENCC62R) or for bit CC62F (if enabled by bit ENCC62F). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.
INPCHE	7:6	rw	**Interrupt Node Pointer for the CHE Interrupt** This bit field defines the interrupt output line, which is activated due to a set condition for bit CHE (if enabled by bit ENCHE) or for bit STR (if enabled by bit ENSTR). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.

Field	Bits	Type	Description
INPERR	9:8	rw	**Interrupt Node Pointer for Error Interrupts** This bit field defines the interrupt output line, which is activated due to a set condition for bit TRPF (if enabled by bit ENTRPF) or for bit WHE (if enabled by bit ENWHE). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.
INPT12	11:10	rw	**Interrupt Node Pointer for Timer T12 Interrupts** This bit field defines the interrupt output line, which is activated due to a set condition for bit T12OM (if enabled by bit ENT12OM) or for bit T12PM (if enabled by bit ENT12PM). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.
INPT13	13:12	rw	**Interrupt Node Pointer for Timer T13 Interrupts** This bit field defines the interrupt output line, which is activated due to a set condition for bit T13CM (if enabled by bit ENT13CM) or for bit T13PM (if enabled by bit ENT13PM). 00_B Interrupt output line SR0 is selected. 01_B Interrupt output line SR1 is selected. 10_B Interrupt output line SR2 is selected. 11_B Interrupt output line SR3 is selected.
RES	15:14	r	**Reserved** Returns 0 if read; should be written with 0.

18.11.8 Register Map

Table 134 shows the CCU6 module base addresses.

Table 135 lists the addresses of the CCU6 SFRs.

Table 134 Register Address Space

Module	Base Address	End Address	Note
CCU6	$4000C000_H$	$4000FFFF_H$	

Table 135 Register Overview

Register Short Name	Register Long Name	Offset Address	Page Number
CC63R	Capture/Compare Register for Channel CC63	00_H	**586**
TCTR4	Timer Control Register 4	04_H	**595**
MCMOUTS	Multi-Channel Mode Output Shadow Register	08_H	**602**
ISR	Capture/Compare Interrupt Status Reset Register	$0C_H$	**611**
CMPMODIF	Compare State Modification Register	10_H	**588**
CC6xSR (x = 0-2)	Capture/Compare Shadow Register for Channel CC60SR	14_H	**583**
CC6xSR (x = 0-2)	Capture/Compare Shadow Register for Channel CC61SR	18_H	**583**
CC6xSR (x = 0-2)	Capture/Compare Shadow Register for Channel CC62SR	$1C_H$	**583**
CC63SR	Capture/Compare Shadow Register for Channel CC63	20_H	**586**
T12PR	Timer T12 Period Register	24_H	**582**
T13PR	Timer T13 Period Register	28_H	**585**
T12DTC	Timer T12 Dead-Time Control Register	$2C_H$	**584**
TCTR0	Timer Control Register 0	30_H	**590**
CC6xR (x = 0-2)	Capture/Compare Register for Channel CC60R	34_H	**582**
CC6xR (x = 0-2)	Capture/Compare Register for Channel CC61R	38_H	**582**
CC6xR (x = 0-2)	Capture/Compare Register for Channel CC62R	$3C_H$	**582**
T12MSEL	Capture/Compare T12 Mode Select Register	40_H	**579**
IEN	Capture/Compare Interrupt Enable Register	44_H	**612**
INP	Capture/Compare Interrupt Node Pointer Register	48_H	**615**
ISS	Capture/Compare Interrupt Status Set Register	$4C_H$	**609**
PSLR	Passive State Level Register	50_H	**601**
MCMCTR	Multi-Channel Mode Control Register	54_H	**605**
TCTR2	Timer Control Register 2	58_H	**593**
MODCTR	Modulation Control Register	$5C_H$	**597**
TRPCTR	Trap Control Register	60_H	**599**
MCMOUT	Multi-Channel Mode Output Register	64_H	**603**
IS	Capture/Compare Interrupt Status Register	68_H	**607**
PISEL0	Port Input Select Register 0	$6C_H$	**574**
PISEL2	Port Input Select Register 2	74_H	**576**

Table 135 Register Overview (cont'd)

Register Short Name	Register Long Name	Offset Address	Page Number
T12	Timer T12 Counter Register	78_H	**581**
T13	Timer T13 Counter Register	$7C_H$	**585**
CMPSTAT	Compare State Register	80_H	**587**

19 UART1/UART2

The description in this chapter applies to both UART1 and UART2.

19.1 Features

- Full-duplex asynchronous modes
  - 8-bit or 9-bit data frames, LSB first.
  - Fixed or variable baud rate.
- Receive-buffered.
- Multiprocessor communication.
- Interrupt are generated when data transmission or receptions are complete.
- Baud-rate generator with fractional divider for generating a wide range of baud rates.
- Hardware logic for break and synch byte detection.

19.2 Introduction

The UART provides a full-duplex asynchronous receiver/transmitter, i.e., it can transmit and receive simultaneously. It is also receive-buffered, i.e., it can commence reception of a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time when the reception of the second byte is complete, one of the bytes will be lost. The serial port receive and transmit registers are both accessed through the special function register (SFR) SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

19.2.1 Block diagram

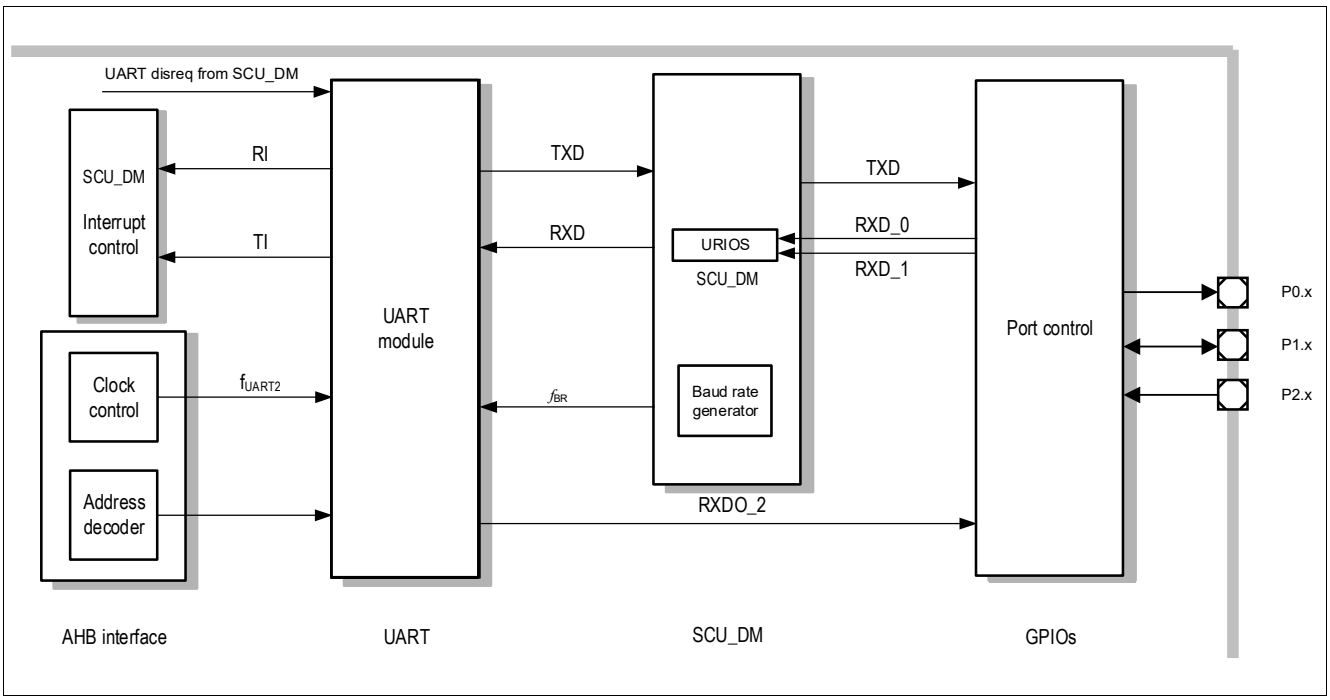


Figure 166 UART block diagram

19.3 UART modes

The UART can be used in four different modes. In mode 0, it operates as an 8-bit shift register. In mode 1, it operates as an 8-bit serial port. In modes 2 and 3, it operates as a 9-bit serial port. The only difference between mode 2 and mode 3 is the baud rate, which is fixed in mode 2 but variable in mode 3. The variable baud rate is set by the underflow rate on the dedicated baud-rate generator.

The different modes are selected by setting bits SM0 and SM1 to the appropriate values, as shown in **Table 136**.

Table 136 UART modes

SM0	SM1	Operating mode	Baud rate
0	0	Mode 0: 8-bit shift register	$f_{PCLK}/2$
0	1	Mode 1: 8-bit shift UART	Variable
1	0	Mode 2: 9-bit shift UART	$f_{PCLK}/64$
1	1	Mode 3: 9-bit shift UART	Variable

UART1 is connected to the integrated LIN transceiver, and to GPIO for test purposes. UART2 is connected to GPIO only.

19.3.1 Mode 0, 8-Bit Shift Register, Fixed Baud Rate

In mode 0, the serial port behaves as an 8-bit shift register. Data is shifted in through RXD, and out through RXDO, while the TXD line is used to provide a shift clock which can be used by external devices to clock data in and out.

The transmission cycle is activated by a write to SBUF. For the next seven machine cycles, the contents of the transmit shift register are shifted right one position and a zero shifted in from the left so that when the MSB of the data byte is at the output position, it has a 1 and a sequence of zeros to its left. The control block then executes one last shift before setting the TI bit.

Reception is started by the condition REN = 1 and RI = 0. At the start of the reception cycle, 11111110_B is written to the receive shift register. In each machine cycle that follows, the contents of the shift register are shifted left one position and the value sampled on the RXD line in the same machine cycle is shifted in from the right. When the 0 of the initial byte reaches the leftmost position, the control block executes one last shift, loads SBUF and sets the RI bit.

The baud rate for the transfer is fixed at $f_{PCLK}/2$ where f_{PCLK} is the input clock frequency, i.e. one bit per machine cycle.

19.3.2 Mode 1, 8-Bit UART, Variable Baud Rate

In mode 1, the UART behaves as an 8-bit serial port. A start bit (0), 8 data bits, and a stop bit (1) are transmitted on TXD or received on RXD at a variable baud rate.

The transmission cycle is activated by a write to SBUF. The data is transferred to the transmit shift register and a 1 is loaded to the 9th bit position (as in mode 0). At phase 1 of the machine cycle after the next rollover in the divide-by-16 counter, the start bit is copied to TXD, and data is activated one bit time later. One bit time after the data is activated, the data starts getting shifted right with zeros shifted in from the left. When the MSB gets to the output position, the control block executes one last shift and sets the TI bit.

Reception is started by a high to low transition on RXD (sampled at 16 times the baud rate). The divide-by-16 counter is then reset and $1111\ 1111_B$ is written to the receive register. If a valid start bit (0) is then detected (based on two out of three samples), it is shifted into the register followed by 8 data bits. If the transition is not followed by a valid start bit, the controller goes back to looking for a high to low transition on RXD. When the

start bit reaches the leftmost position, the control block executes one last shift, then loads SBUF with the 8 data bits, loads RB8 (SCON.2) with the stop bit, and sets the RI bit, provided RI = 0, and either SM2 = 0 (see **Section 19.4**) or the received stop bit = 1. If none of these conditions is met, the received byte is lost.

The associated timings for transmit/receive in mode 1 are illustrated in **Figure 167**.

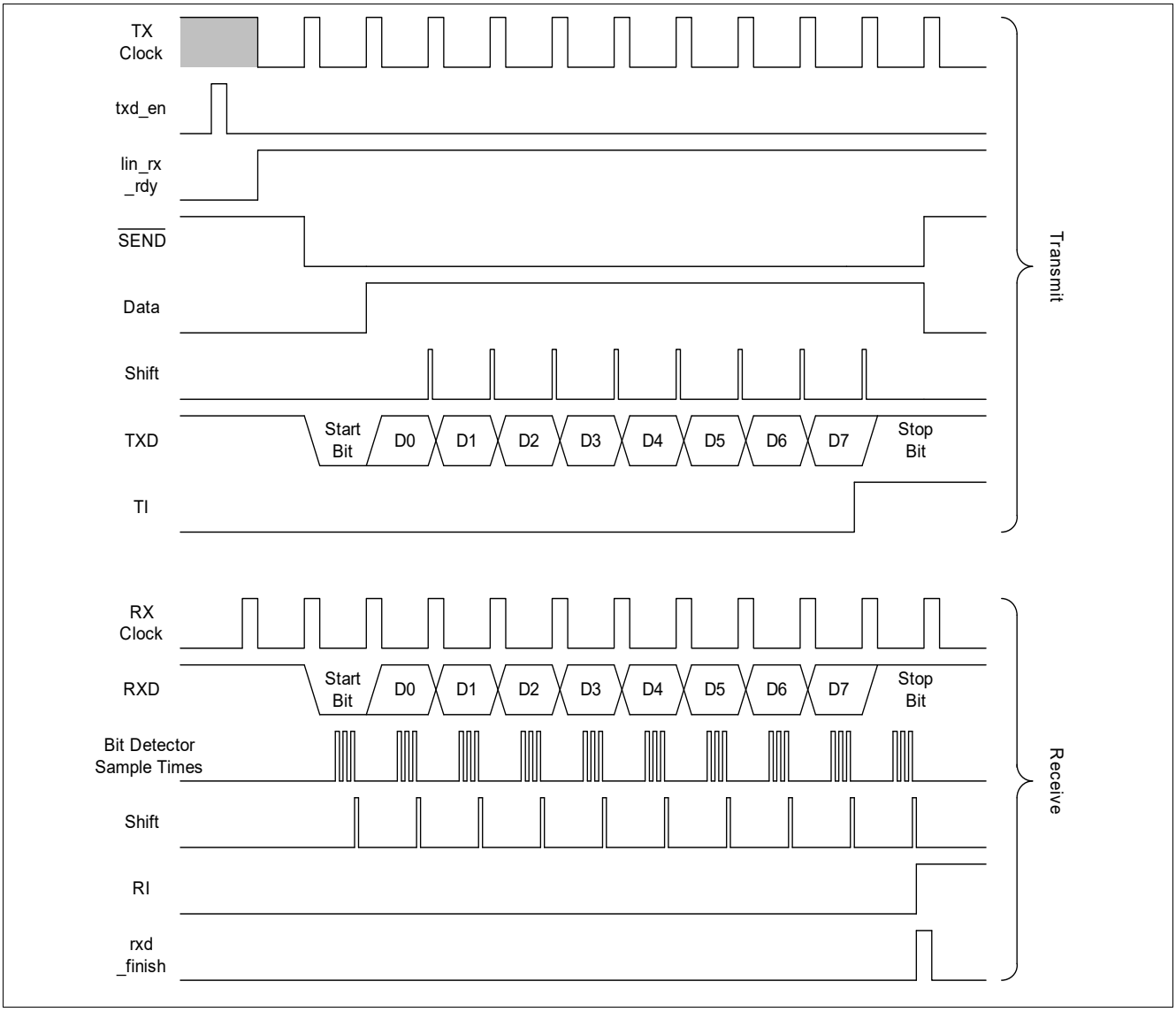


Figure 167 Serial Interface, Mode 1, Timing Diagram

19.3.3 Mode 2, 9-Bit UART, Fixed Baud Rate

In mode 2, the UART behaves as a 9-bit serial port. A start bit (0), 8 data bits plus a programmable 9th bit and a stop bit (1) are transmitted on TXD or received on RXD. The 9th bit for transmission is taken from TB8 (SCON.3) while for reception, the 9th bit received is placed in RB8 (SCON.2).

The transmission cycle is activated by a write to SBUF. The data is transferred to the transmit shift register and TB8 is copied into the 9th bit position. At phase 1 of the machine cycle following the next rollover in the divide-by-16 counter, the start bit is copied to TXD and data is activated one bit time later. One bit time after the data is activated, the data starts shifting right. For the first shift, a stop bit (1) is shifted in from the left and for subsequent shifts, zeros are shifted in. When the TB8 bit gets to the output position, the control block executes one last shift and sets the TI bit.

Reception is started by a high to low transition on RXD (sampled at 16 times the baud rate). The divide-by-16 counter is then reset and $1111\ 1111_B$ is written to the receive register. If a valid start bit (0) is then detected (based on two out of three samples), it is shifted into the register followed by 8 data bits. If the transition is not followed by a valid start bit, the controller goes back to looking for a high to low transition on RXD. When the start bit reaches the leftmost position, the control block executes one last shift, then loads SBUF with the 8 data bits, loads RB8 (SCON.2) with the 9th data bit, and sets the RI bit, provided RI = 0, and either SM2 = 0 (see **Section 19.4**) or the 9th bit = 1. If none of these conditions is met, the received byte is lost.

The baud rate for the transfer is fixed at $f_{PCLK}/64$.

19.3.4 Mode 3, 9-Bit UART, Variable Baud Rate

Mode 3 is the same as mode 2 in all respects except that the baud rate is variable.

In all modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in the modes by the incoming start bit if REN = 1.

The serial interface also provides interrupt requests when transmission or reception of the frames has been completed. The corresponding interrupt request flags are TI or RI, respectively. If the serial interrupt is not used (i.e., serial interrupt not enabled), TI and RI can also be used for polling the serial interface.

The associated timings for transmit/receive in modes 2 and 3 are illustrated in **Figure 168**.

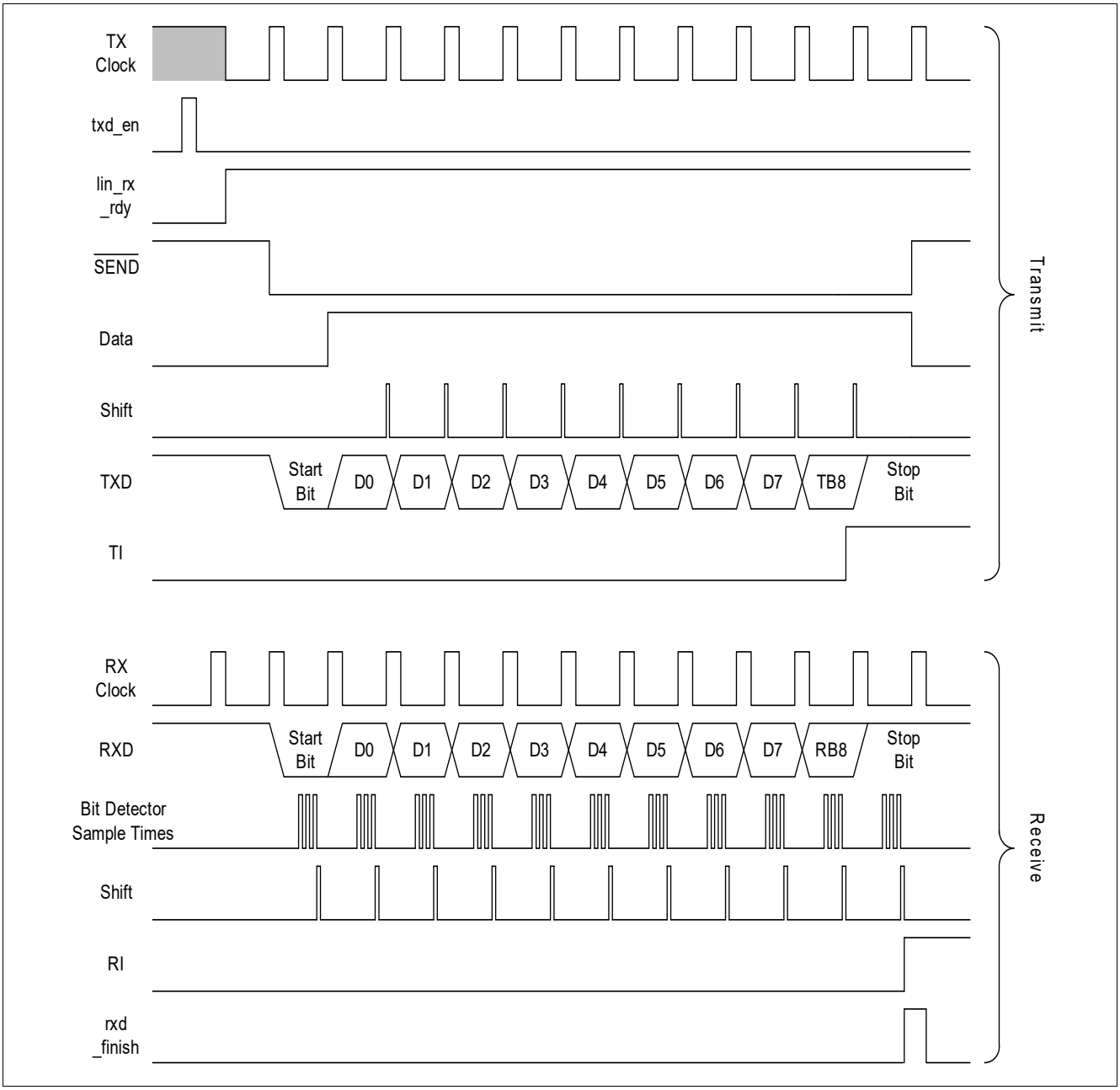


Figure 168 Serial Interface, Modes 2 and 3, Timing Diagram

19.4 Multiprocessor Communication

Modes 2 and 3 have a special provision for multiprocessor communication using a system of address bytes with bit 9 = 1 and data bytes with bit 9 = 0. In these modes, 9 data bits are received. The 9th data bit goes into RB8. The communication always ends with one stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt will be activated only if RB8 = 1.

This feature is enabled by setting bit SM2 in SCON. One of the ways to use this feature in multiprocessor systems is described in the following paragraph.

When the master processor wants to transmit a block of data to one of several slaves, it first sends out an address byte that identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. The slaves that were not being addressed retain their SM2s as set and ignore the incoming data bytes.

Bit SM2 has no effect in mode 0. SM2 can be used in mode 1 to check the validity of the stop bit. In a mode 1 reception, if SM2 = 1, the receive interrupt will not be activated unless a valid stop bit is received.

19.5 Baud Rate Generation

There are several ways to generate the baud rate clock for the serial port, depending on the mode in which they are operating.

The baud rates in modes 0 and 2 are fixed to $f_{PCLK}/2$ and $f_{PCLK}/64$ respectively, while the variable baud rate in modes 1 and 3 is generated based on the setting of the Baud-rate generator in SCU (see **Section 19.5.1**).

"Baud rate clock" and "baud rate" must be distinguished from each other. The serial interface requires a clock rate that is 16 times the baud rate for internal synchronization. Therefore, the UART baud-rate generator must provide a "baud rate clock" to the serial interface where it is divided by 16 to obtain the actual "baud rate". The abbreviation f_{PCLK} refers to the input clock frequency.

19.5.1 Baud-rate Generator

The baud-rate generator in SCU is used to generate the variable baud rate for the UART in modes 1 and 3. It has programmable 11-bit reload value, 3-bit prescaler and 5-bit fractional divider.

The baud-rate generator is clocked derived via a prescaler (f_{DIV}) from the input clock f_{PCLK}. The baud rate timer counts downwards and can be started or stopped through the baud rate control run bit BCON.R. Each underflow of the timer provides one clock pulse to the serial channel. The timer is reloaded with the 11-bit BR_VALUE stored in its reload register BG each time it underflows. The duration between underflows depends on the 'n' value in the fractional divider, which can be selected by the bits BG.FD_SEL. 'n' times out of 32, the timer counts one cycle more than specified by BR_VALUE. The prescaler is selected by the bits BCON.BRPRE.

Register BG is the dual-function Baud-rate Generator/Reload register. Reading BG returns the contents of the timer, while writing to BG (low byte) always updates the reload register.

The BG should be written only when BCON.R is 0. An auto-reload of the timer with the contents of the reload register is performed one instruction cycle after the next time BCON.R is set. Any write to BG, while BCON.R is set, will be ignored.

The baud rate of the baud-rate generator depends on the following bits and register values:

- Input clock f_{PCLK}
- Value of bit field BCON.BRPRE.
- Value of bit field BG.FD_SEL
- Value of the 11-bit reload value BG.BR_VALUE

Figure 169 shows a simplified block diagram of the baud rate generator.

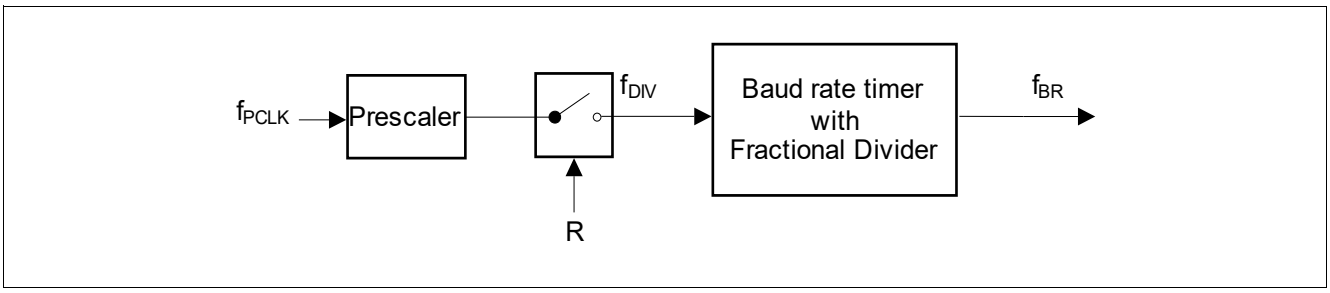


Figure 169 Simplified Baud Rate Generator Block Diagram

The following formula calculate the final baud rate.

$$\text{Baud rate} = \frac{f_{PCLK}}{16 \times PRE \times \left(BR_VALUE + \frac{n}{32}\right)} \tag{19.1}$$

The value of PRE (prescaler) is chosen by the bit field BCON.BRPRE. BR_VALUE represents the contents of the reload value, taken as unsigned 11-bit integer from the bit field BG.BR_VALUE. n/32 is defined by the fractional divider selection in bit field BG.FDSEL.

The maximum baud rate that can be generated is limited to $f_{PCLK}/32$. Hence, for module clocks of 40 MHz and 24 MHz, the maximum achievable baud rate is 1.25 MBaud and 0.75 MBaud respectively.

Table 137 and **Table 138** list various commonly used baud rates together with their corresponding parameter settings and the deviation errors compared to the intended baud rate.

Table 137 Typical Baud Rates of UART (f_{PCLK} = 40 MHz)

Baud rate (f_{PCLK} = 40 MHz)	PRE	Reload Value (BR_VALUE)	Fractional Divider Selection (FD_NUM)	BG Register[1)]	Deviation Error
115.2 kBaud	1 (BRPRE = 000)	21 (15_H)	22 (16_H)	$02B6_H$	+0.06%
20 kBaud	1 (BRPRE = 000)	125 ($7D_H$)	0 (0_H)	$0FA0_H$	0.00%
19.2 kBaud	1 (BRPRE = 000)	130 (82_H)	7 (7_H)	1047_H	-0.01%
9600 Baud	2 (BRPRE = 001)	130 (82_H)	7 (7_H)	1047_H	-0.01%
4800 Baud	4 (BRPRE = 010)	130 (82_H)	7 (7_H)	1047_H	-0.01%
2400 Baud	8 (BRPRE = 011)	130 (82_H)	7 (7_H)	1047_H	-0.01%

1) The value of the 16-bit BG register is obtained by concatenation the 11-bit BRVALUE and 5-bit FD_NUM into a 16-bit value.

Table 138 Typical Baud Rates of UART (f_{PCLK} = 24 MHz)

Baud rate (f_{PCLK} = 24 MHz)	PRE	Reload Value (BR_VALUE)	Fractional Divider Selection (FD_NUM)	BG Register[1)]	Deviation Error
115.2 kBaud	1 (BRPRE = 000)	13 ($0D_H$)	1 (01_H)	$01A1_H$	-0.08%
20 kBaud	1 (BRPRE = 000)	75 ($4B_H$)	0 (00_H)	0960_H	+0.00%
19.2 kBaud	1 (BRPRE = 000)	78 ($4E_H$)	4 (04_H)	$09C4_H$	+0.00%
9600 Baud	2 (BRPRE = 001)	78 ($4E_H$)	4 (04_H)	$09C4_H$	+0.00%
4800 Baud	4 (BRPRE = 010)	78 ($4E_H$)	4 (04_H)	$09C4_H$	+0.00%
2400 Baud	8 (BRPRE = 011)	78 ($4E_H$)	4 (04_H)	$09C4_H$	+0.00%

1) The value of the 16-bit BG register is obtained by concatenation the 11-bit BRVALUE and 5-bit FD_NUM into a 16-bit value.

19.6 LIN Support in UART

The UART module can be used to support the Local Interconnect Network (LIN) protocol for both master and slave operations. The LIN baud rate detection feature, which consists of the hardware logic for Break and Synch Byte detection, provides the capability to detect the baud rate within LIN protocol using Timer2. This allows the UART module to be synchronized to the LIN baud rate for data transmission and reception.

19.6.1 LIN Protocol

LIN is a holistic communication concept for local interconnected networks in vehicles. The communication is based on the SCI (UART) data format, a single-master/multiple-slave concept, a clock synchronization for nodes without stabilized time base. An attractive feature of LIN is self-synchronization of the slave nodes without a crystal or ceramic resonator, which significantly reduces the cost of hardware platform. Hence, the baud rate must be calculated and returned with every message frame.

The structure of a LIN frame is shown in **Figure 170**. The frame consists of the:

- header, which comprises a Break (13-bit time low), Synch Byte (55_H), and ID field
- response time
- data bytes (according to UART protocol)
- checksum

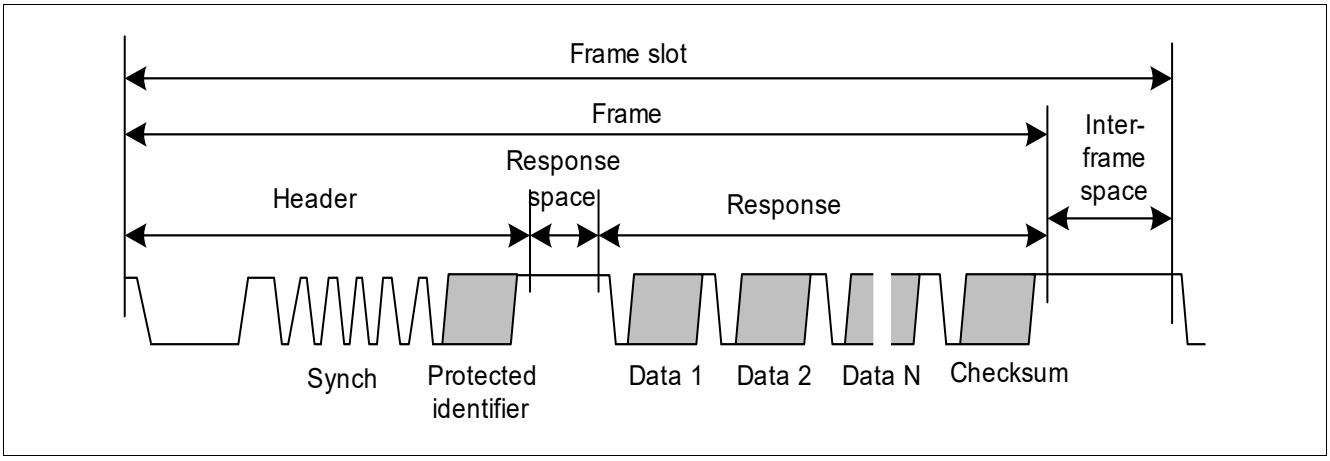


Figure 170 The Structure of LIN Frame

Each byte field is transmitted as a serial byte, as shown in **Figure 171**. 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 a bit with value one (recessive).

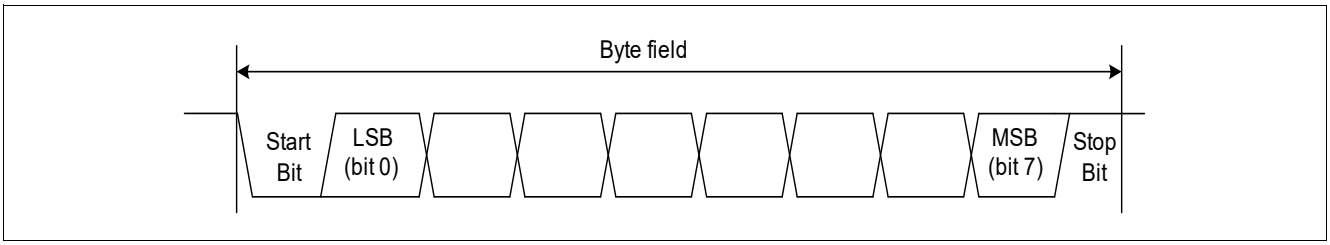


Figure 171 The Structure of Byte Field

The break is used to signal the beginning of a new frame. It is the only field that does not comply with **Figure 171**. A break is always generated by the master task (in the master mode) and it must be at least 13 bits

of dominant value, including the start bit, followed by a break delimiter, as shown in **Figure 172**. The break delimiter will be at least one nominal bit time long.

A slave node will use a break detection threshold of 11 nominal bit times.

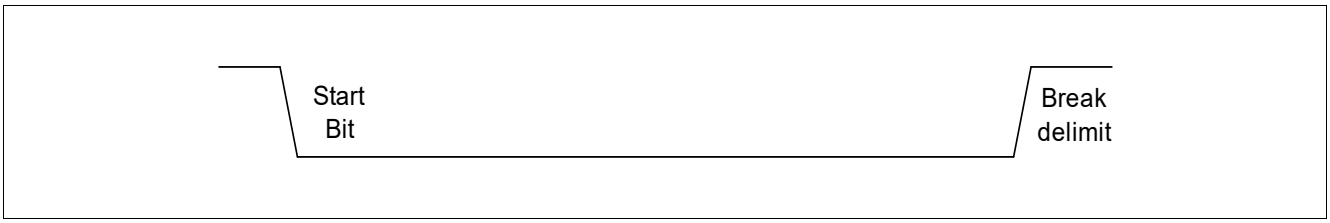


Figure 172 The Break Field

Synch Byte is a specific pattern for determination of time base. The byte field is with the data value 55_H, as shown in **Figure 173**.

A slave task is always able to detect the Break/Synch sequence, even if it expects a byte field (assuming the byte fields are separated from each other). If this happens, detection of the Break/Synch sequence will abort the transfer in progress and processing of the new frame will commence.

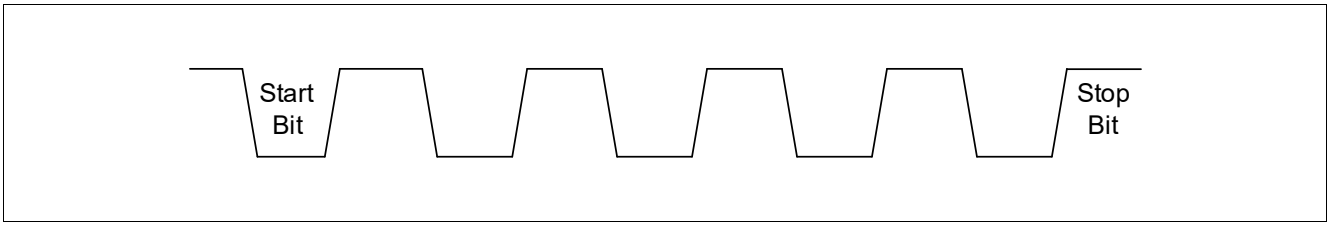


Figure 173 The Synch Byte Field

The slave task will receive and transmit data when an appropriate ID is sent by the master:

1. Slave waits for Synch Break
2. Slave synchronizes on Synch Byte
3. Slave snoops for ID
4. According to ID, slave determines whether to receive or transmit data, or do nothing
5. When transmitting, the slave sends 2, 4 or 8 data bytes, followed by check byte

19.6.2 LIN Header Transmission

LIN header transmission is only applicable in master mode. In the LIN communication, a master task decides when and which frame is to be transferred on the bus. It also identifies a slave task to provide the data transported by each frame. The information needed for the handshaking between the master and slave tasks is provided by the master task through the header portion of the frame.

The header consists of a break and synch pattern followed by an identifier. Among these three fields, only the break pattern cannot be transmitted as a normal 8-bit UART data. The break must contain a dominant value of 13 bits or more to ensure proper synchronization of slave nodes.

In the LIN communication, a slave task is required to be synchronized at the beginning of the protected identifier field of frame. For this purpose, every frame starts with a sequence consisting of a break field followed by a synch byte field. This sequence is unique and provides enough information for any slave task to detect the beginning of a new frame and be synchronized at the start of the identifier field.

19.6.2.1 Automatic Synchronization to the Host

Upon entering LIN communication, a connection is established and the transfer speed (baud rate) of the serial communication partner (host) is automatically synchronized in the following steps that are to be included in user software:

STEP 1: Initialize interface for reception and timer for baud rate measurement

STEP 2: Wait for an incoming LIN frame from host

STEP 3: Synchronize the baud rate to the host

STEP 4: Enter for Master Request Frame or for Slave Response Frame

The next sections, **Section 19.6.2.2**, **Section 19.6.2.3** and **Section 19.6.2.4** provide some hints on setting up the microcontroller for baud rate detection of LIN.

Note: Re-synchronization and setup of baud rate are always done for **every** *Master Request Header or Slave Response Header LIN frame.*

19.6.2.2 Initialization of Break/Synch Field Detection Logic

The LIN baud rate detection feature provides the capability to detect the baud rate within the LIN protocol using Timer2. Initialization consists of:

- Serial port of the microcontroller set to Mode 1 (8-bit UART, variable baud rate) for communication.
- Provide the baud rate range via bit field BCON.BGSEL.
- Toggle BCON.BRDIS bit (set the bit to 1 before clearing it back to 0) to initialize the Break/Synch detection logic.
- Clear all status flags LINST.BRK, LINST.EOFSYN and LINST.ERRSYN to 0.
- Timer2 is set to capture mode with falling edge trigger at pin T2EX. Bit T2MOD.EDGESEL is set to 0 by default and bit T2CON.CP/$\overline{\text{RL2}}$ is set to 1.
- Timer2 external events are enabled. T2CON. EXEN2 is set to 1. (EXF2 flag is set when a negative transition occurs at pin T2EX)
- f_{T2} can be configured by bit field T2MOD.T2PRE.

19.6.2.3 Baud Rate Range Selection

The Synch Break and Synch Byte detection logic supports a maximum number of bits in the Break field as defined by **Equation (19.2)**.

$$\text{Maximum number of bits} = \text{Baud Rate} \times \frac{4095}{\text{Sample Frequency}} \tag{19.2}$$

The sample frequency is given by **Equation (19.3)**.

$$\text{Sample Frequency} = \frac{f_{PCLK}}{8 \times 2^{BGSEL}} \tag{19.3}$$

If the maximum number of bits in the Break field is exceeded, the internal counter will overflow, which results in a baudrate detection error. Therefore, an appropriate BGSEL value has to be selected for the required baudrate detection range.

The baud rate range defined by different BGSEL settings is shown in the following table. The BGSEL settings and baudrate ranges are valid for a synch break field of 13 bits.

Table 139 BGSEL Bit Field Definition for Different Input Frequencies

f_{PCLK}	BGSEL	Baud Rate Select for Detection f_{pclk}/(2184*2^BGSEL) to f_{pclk}/(72*2^BGSEL)
40 MHz	00_B	18.3 kHz to 555.6 kHz
	01_B	9.2 kHz to 277.8 kHz
	10_B	4.6 kHz to 138.9 kHz
	11_B	2.3 kHz to 69.4 kHz
24 MHz	00_B	11 kHz to 333.3 kHz
	01_B	5.5 kHz to 166.7 kHz
	10_B	2.8 kHz to 83.3 kHz
	11_B	1.4 kHz to 41.7 kHz

Each BGSEL setting supports a range of baud rate for Synch Break and Synch Byte detection. If the baud rate used is outside the defined range, the Synch Break and Synch Byte may not be detected correctly. In order to determine the right BGSEL range the following equation helps selecting the right range:

$n = f_{SYS}$ / Master_BaudRate

if n > 576 then BGSEL = 3

if n > 288 then BGSEL = 2

if n > 144 then BGSEL = 1

if n > 72 then BGSEL = 0

For any value n smaller than 72 a proper Synch Break and Synch Byte detection may not be given.

Since a given BGSEL value defines a certain master baudrate range, alternatively the recognized synch break length can be considered for given master baudrates. The **Table 140** lists the recognized synch break length in master baudrate bit times depending on the system frequency, the BGSEL setting and the master baudrate. The table lists nominal values, clock tolerances are not taken into account. Furthermore these values are only for user guidance, proper evaluation in the user application should still be considered.

Table 140 Recognized Synch Break Length in bit times for a given system frequency, BGSEL selection and master baudrate

f_{PCLK}	BGSEL	9600 baud [bits]	10400 baud [bits]	19200 baud [bits]
40 MHz	00_B	-	-	11..15
	01_B	11..15	11..16	11..30
	10_B	11..30	11..32	11..60
	11_B	11..60	11..65	11..120
24 MHz	00_B	11..12	11..13	11..25
	01_B	11..25	11..27	11..50
	10_B	11..50	11..54	11..100
	11_B	11..100	11..108	11..200

19.6.2.4 LIN Baud Rate Detection

This chapter gives an example on how the hardware features can be used to detect a Synch Break and how to measure the LIN master baud rate.

The baud rate detection for LIN is shown in **Figure 174**.

The Header LIN frame consists of the:

- Synch Break (13 bit times low or higher)
- Synch Byte (55_H)
- Protected ID field

where only the Synch Break and Synch Byte is of interest for the baud rate detection. The LIN header is not even displayed in **Figure 174**.

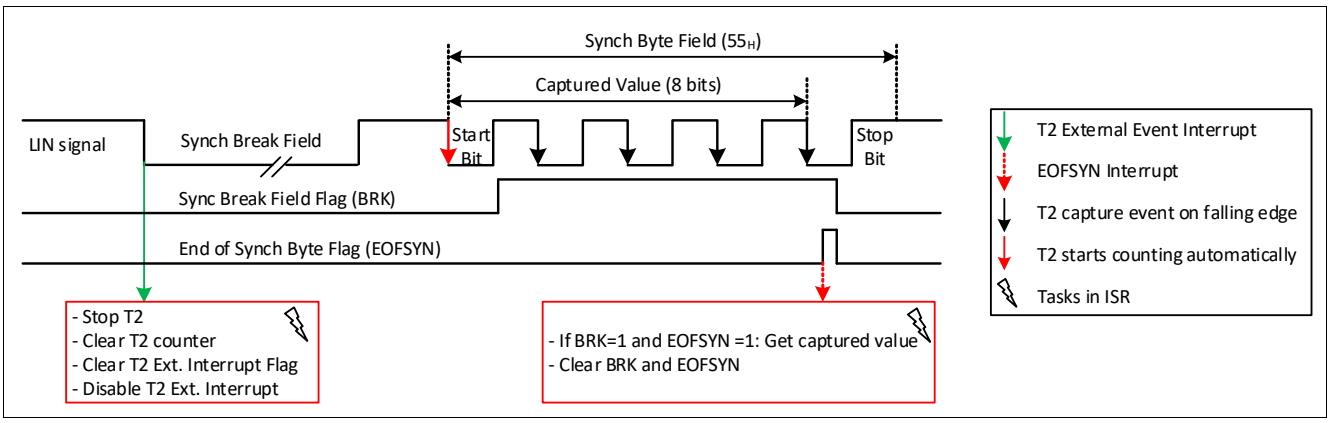


Figure 174 LIN Auto Baud Rate Detection

The Timer2 (**Chapter 16**) is used to measure the baud rate of the LIN master. For this purpose the Timer2 should be configured to operate in capture mode. Timer2 should start on a falling edge on the T2EX signal, which is by default connected to the internal LIN transceiver. Furthermore the Timer2 should be configured to capture T2EX on the falling edge as well. The **LINST**.BGSEL should be set to a value which covers the expected baud rate of the LIN master to be able to recognize the Synch Break. **Table 139** gives possible values for f_{SYS} = 24MHz and 40MHz. For any other system frequencies the correct LINST.BGSEL values has to be calculated according to the formula given along with the table. E.g. for f_{SYS} = 40 MHz a value of LINST.BGSEL = 3 would be a good choice.

Timer2 starts with the falling edge on LIN line at the beginning of the Synch Break and triggers an interrupt. Inside this ISR the Timer2 should be stopped by the user and the Timer2 counter values has to be reset, because at this time the Timer2 value is not of interest, the main purpose of this ISR is to reset and arm the Timer2 again. To avoid any undesired interrupts during the following Synch Byte the Timer2 external interrupt should be disabled. The Timer2 is now ready to start again on the next falling edge of the LIN line, with the start bit of the Synch Byte Field. The Timer2 now captures the current count value in its capture register on every falling edge of the LIN signal (Synch Byte). On every falling edge the previous captured timer value will be overwritten.

The **LINST**.BRK signals a valid Synch Break Field, means the Synch Break Field was at least 11 times longer than the first bit of the Synch Byte. The signal **LINST**.EOFSYN gets set once a valid Synch Byte was received. The LINST.EOFSYN can be used to trigger an interrupt in which the LINST.BRK should be checked and along with LINST.EOFSYN these flags should be cleared (**LINSCLR**). Inside this EOFSYS interrupt service routine the user reads the Timer2 capture value which reflects the Timer2 count value at the 8th bit of the Synch Byte. The Timer2 capture value can now be used to calculate the master baud rate and program the UART1 accordingly.

If the Synch Byte was not received properly then the signal **LINST**.ERRSYN gets set instead of LINST.EOFSYN. Also the signal LINST.ERRSYN can issue an interrupt which could be used to clear the flags and implement an error handling.

Latest at the end of the LIN frame or with the reception of the LINST.ERRSYN signal, the Timer2 external interrupt should be enabled again to be able to trigger on the next falling edge of the Synch Break.

19.7 Module Interfaces

An overview of the UART I/O interface is shown in **Figure 175** (UART1) and **Figure 177** (UART2).

In mode 0 (the serial port behaves as shift register), data is shifted in through RXD and out through RXDO, while the TXD line is used to provide a shift clock which can be used by external devices to clock data in and out. In modes 1, 2 and 3, the port behaves as an UART. Data is transmitted on TXD and received on RXD.

UART1:

Data that is shifted into and out of the UART through RXD and TXD respectively.

RXD from UART1 can be selected from different sources (LIN or RXD1). This selection is performed within the SCU via SFR bit MODPISEL.URIOS1, see **Table 141**.

RXD and TXD from UART1 are connected to LIN can be monitored via RXD1 and TXD1 outputs on GPIO (by enabling the alternate outputs), UART1 in LIN mode or LIN monitoring mode, see **Table 141**.

In addition, UART1 transmission can be disconnected from LIN and provided via input port TXD1 (selected via . MODPISEL.U_TX_CONDIS), UART1 not used, see **Table 141**.

If UART1 is used as UART, RXD1 is input and TXD1 is output.

UART2:

RXD and TXD from UART2 can be selected from different sources. This selection is performed by the SCU via SFR bit MODPISEL3.URIOS2.

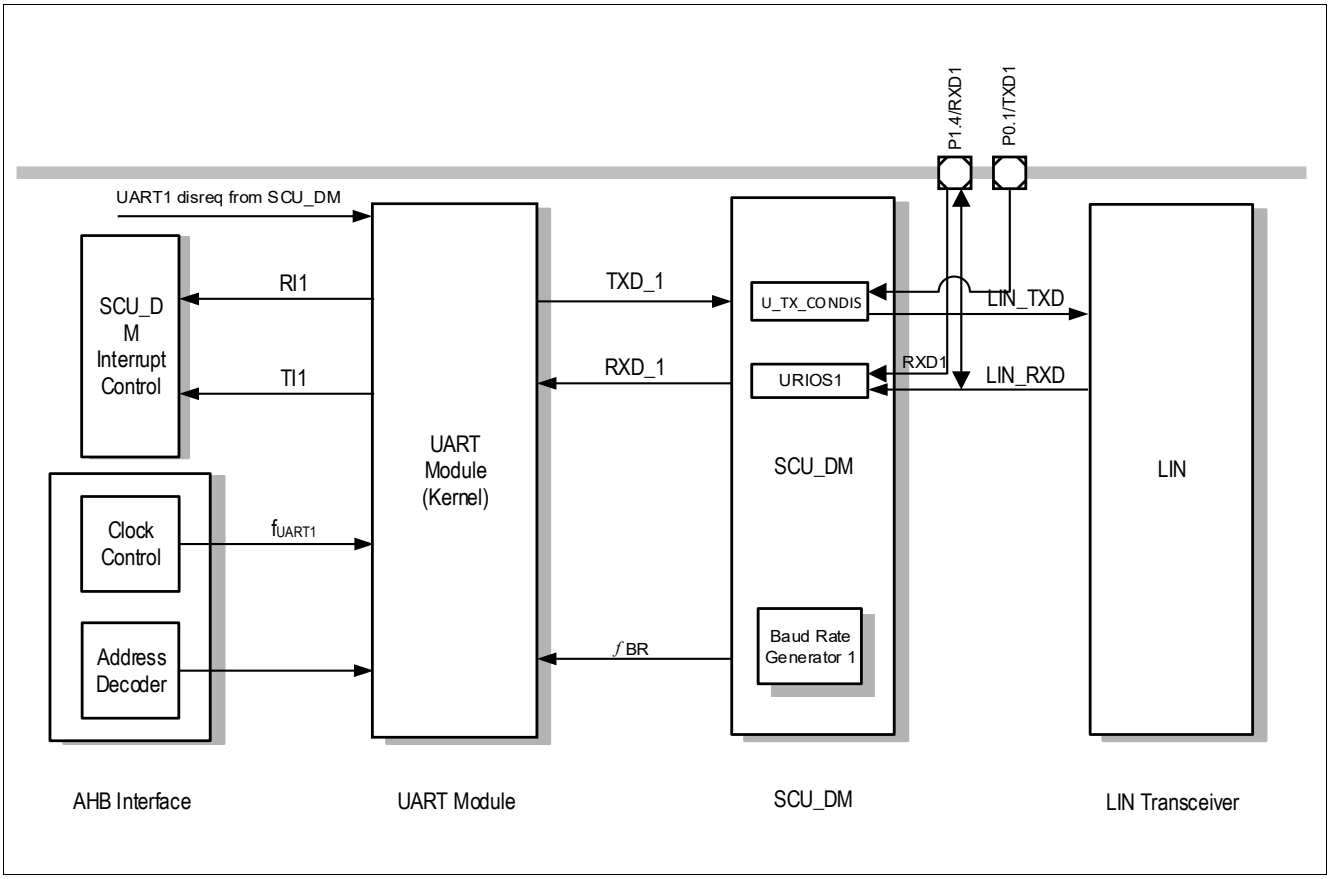


Figure 175 UART1 Module I/O Interface

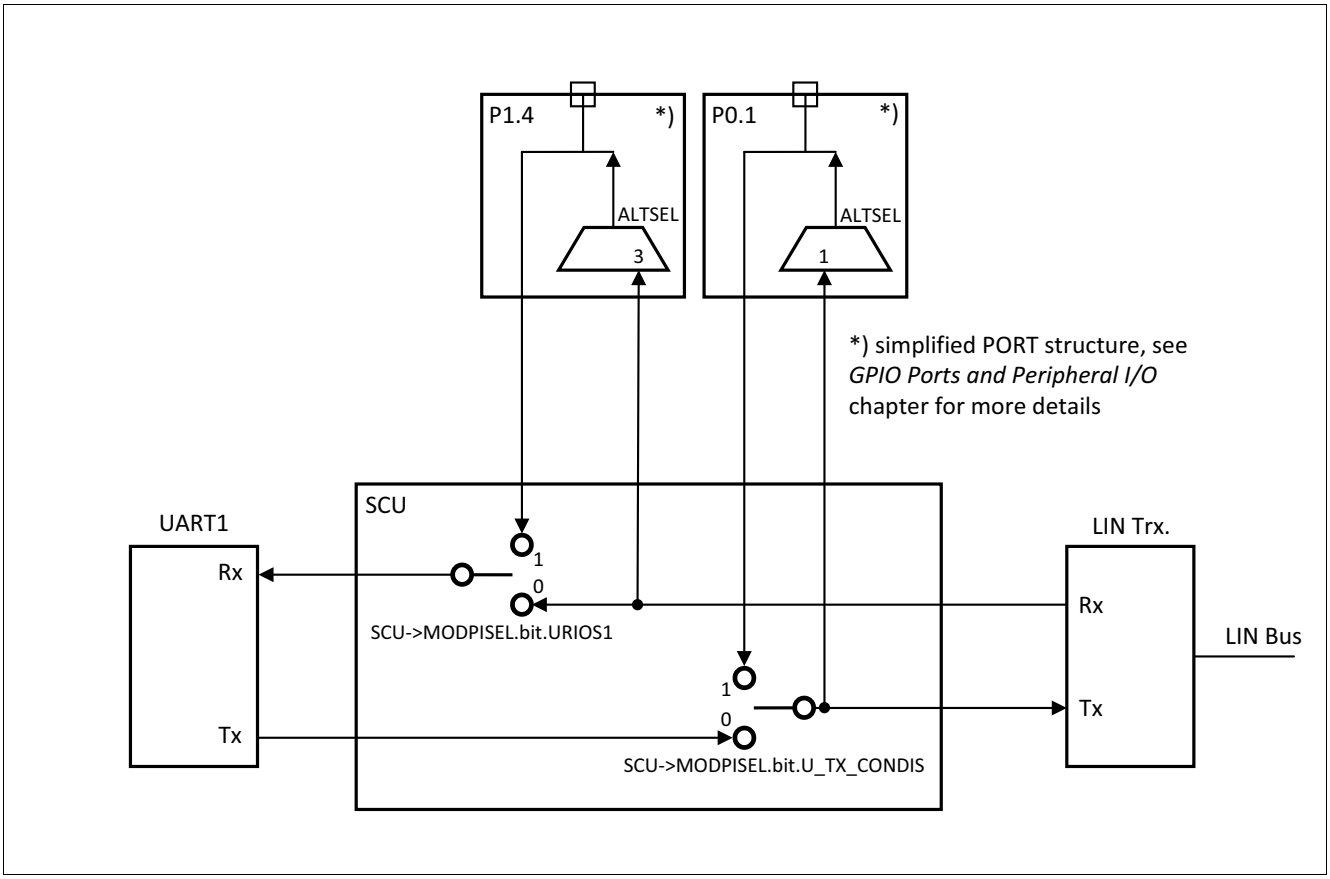


Figure 176 UART1 Module I/O Interface, detailed

Figure 175 shows the connectivity of the UART1 to the integrated LIN transceiver and the GPIOs, P1.4 and P0.1. The switches in the SCU, **MODPISEL**.URIOS1 and **MODPISEL**.U_TX_CONDIS can be used to either connect the UART1 to the LIN transceiver, the UART1 to the GPIOs, or the GPIOs to the LIN transceiver. **Table 141** lists the possible combinations and its connection result.

Table 141 Connection options

MODPISEL.URIOS1	MODPISEL.U_TX_CONDIS	Connection
0	0	UART1 connected to LIN Trx. UART1.Tx to LIN.Tx (optional to P0.1 output, monitoring) UART1.Rx to LIN.Rx (optional to P1.4 output, monitoring)
0	1	LIN Trx. connected to GPIOs (like standalone LIN Trx.) P0.1 (input) connected to LIN.Tx P1.4 (output) connected to LIN.Rx (UART1.Rx) *Note: The LIN transceiver will switch into receive only mode if P0.1 remains '0' for $t > t_{timeout}$, see* ***Chapter 20.3.3***

Table 141 Connection options

MODPISEL.URIOS1	MODPISEL.U_TX_CONDIS	Connection
1	0	UART1 connected to GPIOs P0.1 (output) connected to UART1.Tx (and connected to LIN.Tx, set LIN to sleep mode in case not needed) P1.4 (input) connected to UART1.Rx
1	1	LIN Trx. connected to GPIOs (like standalone LIN Trx.) P0.1 (input) connected to LIN.Tx P1.4 (output) connected to LIN.Rx (UART1.Rx) *Note: The LIN transceiver will switch into receive only mode if P0.1 remains '0' for $t > t_{timeout}$, see* ***Chapter 20.3.3***

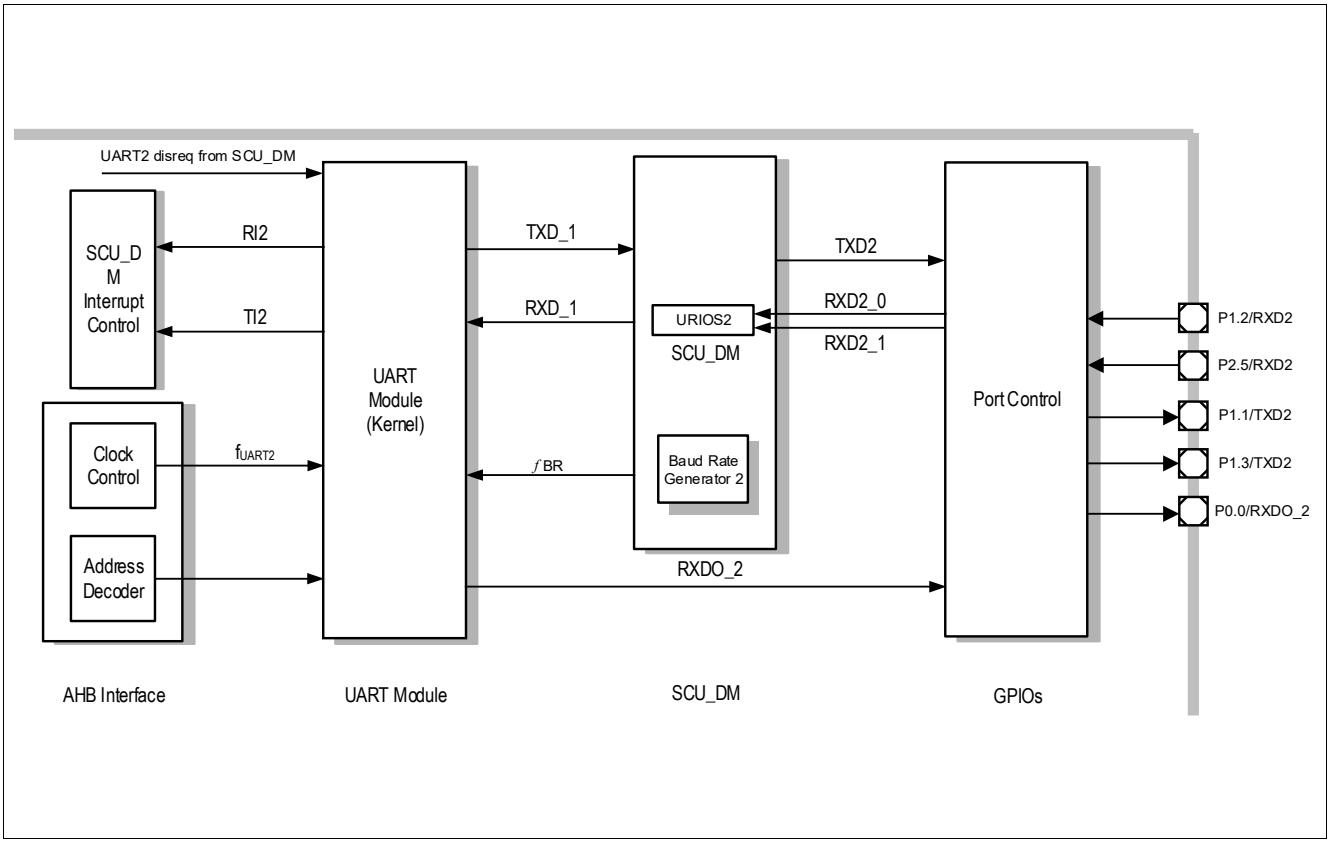


Figure 177 UART2 Module I/O Interface

19.8 Interrupts

The two UART interrupts can be separately enabled or disabled by setting or clearing their corresponding enable bits in SCU SFR MODIEN. An overview of the UART interrupt sources is shown in **Table 142** and **Table 143**.

Table 142 UART1 Interrupt Sources

Interrupt	Flag	Interrupt Enable Bit
Reception completed	SCON.RI	SCU_MODIEN1.RIEN
Transmission completed	SCON.TI	SCU_MODIEN1.TIEN

Table 143 UART2 Interrupt Sources

Interrupt	Flag	Interrupt Enable Bit
Reception completed	SCON.RI	SCU_MODIEN2.RIEN
Transmission completed	SCON.TI	SCU_MODIEN2.TIEN

19.9 Register Definition

19.9.1 UART Registers

19.9.1.1 UART Control Registers

UART contains the two Special Function Registers (SFRs), **SCON** and **SBUF**. **SCON** is the control register and **SBUF** is the data register. On reset, both **SCON** and **SBUF** return 00_H.

The serial port control and status register is the SFR **SCON**. This register contains not only the mode selection bits, but also the 9th data bit for transmit and receive (TB8 and RB8) and the serial port interrupt bits (TI and RI).

SBUF is the receive and transmit buffer of the serial interface. Writing to **SBUF** loads the transmit register and initiates transmission. This register is used for both transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the two paths are independent.

Reading out SBUF accesses a physically separate receive register.

SBUF
Serial Data Buffer **(04_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
VAL							
rwh							

Field	Bits	Type	Description
VAL	[7:0]	rwh	**Serial Interface Buffer Register**

SCON
Serial Channel Control Register **(00_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
SM0	**SM1**	**SM2**	**REN**	**TB8**	**RB8**	**TI**	**RI**
rw	rw	rw	rw	rw	rwh	rwh	rwh

Field	Bits	Type	Description
RI	0	rwh	**Receive Interrupt Flag** This is set by hardware at the end of the 8th bit on mode 0, or at the half point of the stop bit in modes 1, 2, and 3. Flag can be set by software by writing 1 to RI Flag must be cleared by software in SCONCLR.RICLR.
TI	1	rwh	**Transmit Interrupt Flag** This is set by hardware at the end of the 8th bit in mode 0, or at the beginning of the stop bit in modes 1, 2, and 3. Flag can be set by software by writing 1 to TI Flag must be cleared by software in SCONCLR.TICLR.
RB8	2	rwh	**Serial Port Receiver Bit 9** In modes 2 and 3, this is the 9th data bit received. In mode 1, this is the stop bit received. In mode 0, this bit is not used.
TB8	3	rw	**Serial Port Transmitter Bit 9** In modes 2 and 3, this is the 9th data bit sent.
REN	4	rw	**Enable Receiver of Serial Port** 0_B Serial reception is disabled. 1_B Serial reception is enabled.
SM2	5	rw	**Enable Serial Port Multiprocessor Communication in Modes 2 and 3** In mode 2 or 3, if SM2 is set to 1, RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 is set to 1, RI will not be activated if a valid stop bit (RB8) was not received. In mode 0, SM2 should be 0.
SM1, SM0	6, 7	rw	**Serial Port Operating Mode Selection** 00_B Mode 0: 8-bit shift register, fixed baud rate ($f_{PCLK}/2$). 01_B Mode 1: 8-bit UART, variable baud rate. 10_B Mode 2: 9-bit UART, fixed baud rate ($f_{PCLK}/64$). 11_B Mode 3: 9-bit UART, variable baud rate.

19.9.1.2 Baud-Rate Generator Control and Status Registers

BCON1
Baud Rate Control Register 1 **(88_H)** **Reset Value: 00_H**
BCON2
Baud Rate Control Register 2 **(98_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES				**BRPRE**			**R**
r				rw			rw

Field	Bits	Type	Description
R	0	rw	**Baud Rate Generator Run Control Bit** 0_B Baud-rate generator disabled. 1_B Baud-rate generator enabled. *Note: BR_VALUE should only be written if R = 0.*
BRPRE	[3:1]	rw	**Prescaler Bit** Selects the input clock for f_{DIV} which is derived from the peripheral clock. 000_B $f_{DIV} = f_{PCLK}$ 001_B $f_{DIV} = f_{PCLK}/2$ 010_B $f_{DIV} = f_{PCLK}/4$ 011_B $f_{DIV} = f_{PCLK}/8$ 100_B $f_{DIV} = f_{PCLK}/16$ 101_B $f_{DIV} = f_{PCLK}/32$ Others: reserved
RES	[7:4]	r	**Reserved** Returns 0 if read; should be written with 0.

LINST
LIN Status Register **(94_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES	**SYNEN**	**ERRSYN**	**EOFSYN**	**BRK**	**BGSEL**		**BRDIS**
r	rw	r	r	r	rw		rw

Field	Bits	Type	Description
BRDIS	0	rw	**Baud Rate Detection Disable** 0_B Break/Synch detection is enabled. 1_B Break/Synch detection is disabled.
BGSEL	[2:1]	rw	**Baud Rate Select for Detection** For different values of BGSEL, the baud rate range for detection is defined by the following formula: f_{pclk}/(2184*2^BGSEL) < baud rate range < f_{pclk}/(72*2^BGSEL) where BGSEL = 00_B, 01_B, 10_B, 11_B. See **Table 137** for bit field BGSEL definition for different input frequencies.
BRK	3	r	**Break Field Flag** This bit is set by hardware and can only be cleared by software. 0_B Break Field is not detected. 1_B Break Field is detected.
EOFSYN	4	r	**End of SYN Byte Interrupt Flag** This bit is set by hardware and can only be cleared by software. 0_B End of SYN Byte is not detected. 1_B End of SYN Byte is detected.
ERRSYN	5	r	**SYN Byte Error Interrupt Flag** This bit is set by hardware and can only be cleared by software. 0_B Error is not detected in SYN Byte. 1_B Error is detected in SYN Byte.
SYNEN	6	rw	**End of SYN Byte and SYN Byte Error Interrupts Enable** 0_B End of SYN Byte and SYN Byte Error Interrupts are not enabled. 1_B End of SYN Byte and SYN Byte Error Interrupts are enabled.
RES	7	r	**Reserved** Returns 0 if read; should be written with 0.

LINSCLR
LIN Status Clear Register **($0A4_H$)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES		ERRSYNC	EOFSYNC	BRKC	RES		
r		w	w	w		r	

Field	Bits	Type	Description
RES	2:0	r	**Reserved** Returns 0 if read; should be written with 0.
BRKC	3	w	**Break Field Flag Clear** This bit is set by software and can only be cleared by hardware. 0_B Break Field flag is not cleared. 1_B Break Field is cleared.
EOFSYNC	4	w	**End of SYN Byte Interrupt Flag Clear** This bit is set by software and can only be cleared by hardware. 0_B End of SYN Byte Interrupt Flag is not cleared. 1_B End of SYN Byte is cleared.
ERRSYNC	5	w	**SYN Byte Error Interrupt Flag** This bit is set by software and can only be cleared by hardware. 0_B Error in SYN Byte Error Interrupt Flag is not cleared. 1_B Error in SYN Byte cleared.
RES	7:6	r	**Reserved** Returns 0 if read; should be written with 0.

19.9.1.3 Baud-Rate Generator Timer/Reload Registers

The low and high bytes of the baud rate timer/reload register BG contains the 11-bit reload value for the baud rate timer and the 5-bit fractional divider selection.

Reading the low byte of register BG returns the content of the lower three bits of the baud rate timer and the FD_SEL setting, while reading the high byte returns the content of the upper 8 bits of the baud rate timer.

Writing to register BG loads the baud rate timer with the reload and fractional divider values from the BG register, the first instruction cycle after BCON.R is set.

BG should only be written if R = 0.

Note: The Baud-rate Generator Timer/Reload registers are located inside the SCU_DM module. For accessing them, the base address of the SCU_DM module has to be used.

BGL1
Baud Rate Timer/Reload Register 1, Low Byte (8C$_H$) Reset Value: 00$_H$
BGL2
Baud Rate Timer/Reload Register 2, Low Byte (9C$_H$) Reset Value: 00$_H$

7	6	5	4	3	2	1	0
BR_VALUE			**FD_SEL**				
rwh			rw				

Field	Bits	Type	Description
FD_SEL	[4:0]	rw	**Fractional Divider Selection** Selects the fractional divider to be n/32, where n is the value of FD_SEL and is in the range of 0 to 31. For example, writing 0001$_B$ to FD_SEL selects the fractional divider to be1/32. *Note: Fractional divider has no effect if BR_VALUE = 000$_H$.*
BR_VALUE	[7:5]	rwh	**Baud Rate Timer/Reload Value** The lower three bits of the 11-bit Baud Rate Timer/Reload value. See description in BGH register.

BGH1
Baud Rate Timer/Reload Register 1, High Byte (90$_H$) **Reset Value: 00$_H$**
BGH2
Baud Rate Timer/Reload Register 2, High Byte (A0$_H$) **Reset Value: 00$_H$**

7	6	5	4	3	2	1	0
BR_VALUE							
rwh							

Field	Bits	Type	Description
BR_VALUE	[7:0]	rwh	**Baud Rate Timer/Reload Value** The upper 8 bits of the 11-bit Baud Rate Timer/Reload value. The definition of the 11-bit reload value is as follows: 000_H Baud-rate timer is bypassed. 001_H 1 002_H 2 : : : : : : $7FE_H$ 2046 $7FF_H$ 2047

SCONCLR
SCON Clear Register **(08_H)** **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES						TICLR	RICLR
r						w	w

Field	Bits	Type	Description
RES	[7:2]	r	**Reserved** Returns 0 if read; should be written with 0.
TICLR	1	w	**SCON.TI Clear Flag** The definition of the clear flag is as follows: 0_H TI Flag is not cleared. 1_B TI Flag is cleared *Note: Read to TICLR always return 0.*
RICLR	0	w	**SCON.RI Clear Flag** The definition of the clear flag is as follows: 0_H RI Flag is not cleared. 1_B RI Flag is cleared *Note: Read to RICLR always return 0.*

19.9.2 Register Map

There are two UART kernels in the TLE987x, namely UART1 and UART2. UART1 is dedicated for LIN transmission. UART2 is dedicated for external UART communication.

Table 145 lists the addresses of the UART SFRs. **Table 146** lists the addresses of the SCU SFRs dedicated for UART1/2 functionality.

Table 144 shows the UART module base addresses.

Table 144 Register Address Space

Module	Base Address	End Address	Note
UART1	48020000_H	$48021FFF_H$	
UART2	48022000_H	$48023FFF_H$	
SCU	50005000_H	$50005FFF_H$	

Table 145 Register Overview UART

Register Short Name	Register Long Name	Offset Address	Page Number
SCON	Serial Channel Control Register	00_H	**639**
SBUF	Serial Data Buffer	04_H	**638**
SCONCLR	SCON Clear Register	08_H	**645**

Table 146 Register Overview SCU Module

Register Short Name	Register Long Name	Offset Address	Page Number
BCON1, dedicated for UART1	Baud Rate Control Register 1	88_H	**640**
BGL1, dedicated for UART1	Baud Rate Timer/Reload Register 1, Low Byte	$8C_H$	**643**
BGH1, dedicated for UART1	Baud Rate Timer/Reload Register 1, High Byte	90_H	**644**
LINST, dedicated for UART1	LIN Status Register	94_H	**641**
BCON2, dedicated for UART2	Baud Rate Control Register 2	98_H	**640**
BGL2, dedicated for UART2	Baud Rate Timer/Reload Register 2, Low Byte	$9C_H$	**643**
BGH2, dedicated for UART2	Baud Rate Timer/Reload Register 2, High Byte	$A0_H$	**644**
LINSCLR, dedicated for UART1	LIN Status Clear Register	$A4_H$	**642**

20 LIN transceiver

20.1 Features

General functional features

- Compliant with the LIN2.2 standard, backward-compatible with LIN1.3, LIN2.0, and LIN 2.1
- Compliant with SAE J2602 (slew rate, receiver hysteresis)

Special features

- Measurement of the LIN master baudrate via Timer2
- LIN can be used as input/output with SFR bits
- TxD timeout feature (optional, on by default)

Operation mode features

- LIN Sleep mode (LSLM)
- LIN Receive-Only mode (LROM)
- LIN Normal mode (LNM)
- High voltage input/output mode (LHVIO)

Supported baud rates

- Mode for transmission with up to 10.4 kilobaud
- Mode for transmission with up to 20 kilobaud
- Mode for transmission with up to 40 kilobaud
- Mode for transmission with up to 115.2 kilobaud

Slope mode features

- Normal Slope mode (20 kbit/s)
- Low Slope mode (10.4 kbit/s)
- Flash mode (115.2 kbit/s)

Wake-up features

- LIN bus wake-up

20.2 Introduction

The LIN module is a transceiver for the Local Interconnect Network (LIN), compliant with the LIN2.2 standard and backward-compatible with LIN1.3, LIN2.0 and LIN2.1. It operates as a bus driver between the protocol controller and the physical network. The LIN bus is a single-wire, bidirectional bus typically used for in-vehicle networks, using baud rates between 2.4 kilobaud and 20 kilobaud. Additionally, baud rates up to 115.2 kilobaud are implemented.

The LIN module offers several different operation modes, including a LIN Sleep mode and the LIN Normal mode. The integrated slope control allows using several data transmission rates with optimized EMC performance. For data transfer at the end of line, a Flash mode up to 115.2 kilobaud is implemented. In specific conditions, this Flash mode supports data rates of up to 250 kbit/s. (In production environments, in point-to-point communications with reduced wire lengths and limited supply voltages.).

20.2.1 Block diagram

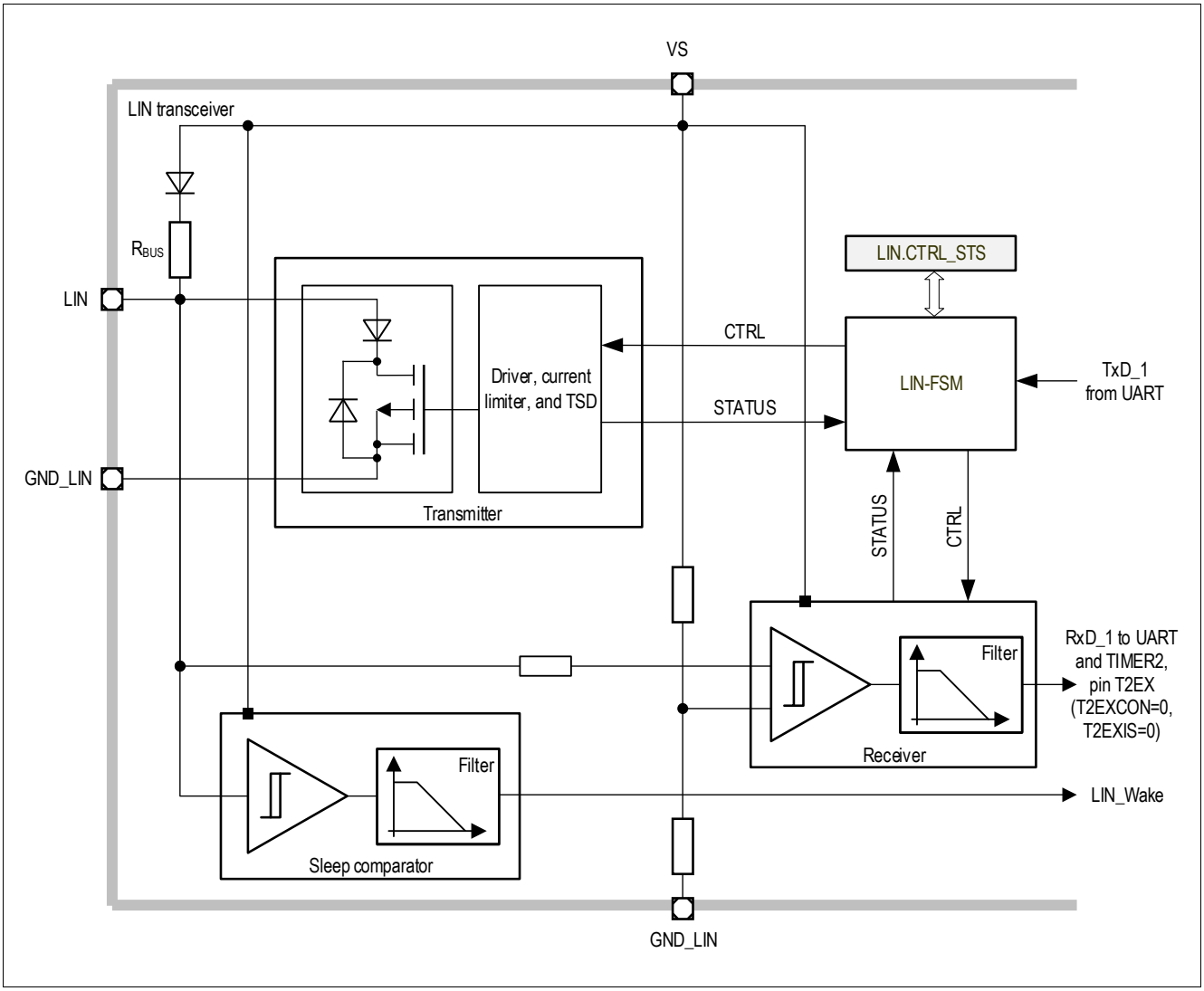


Figure 178 LIN transceiver block diagram

20.3 Functional Description

20.3.1 LIN Transceiver Modes

The LIN Module is controlled by an internal state machine which determines the actual state of the transceiver. This state machine is controllable by the SFR interface.

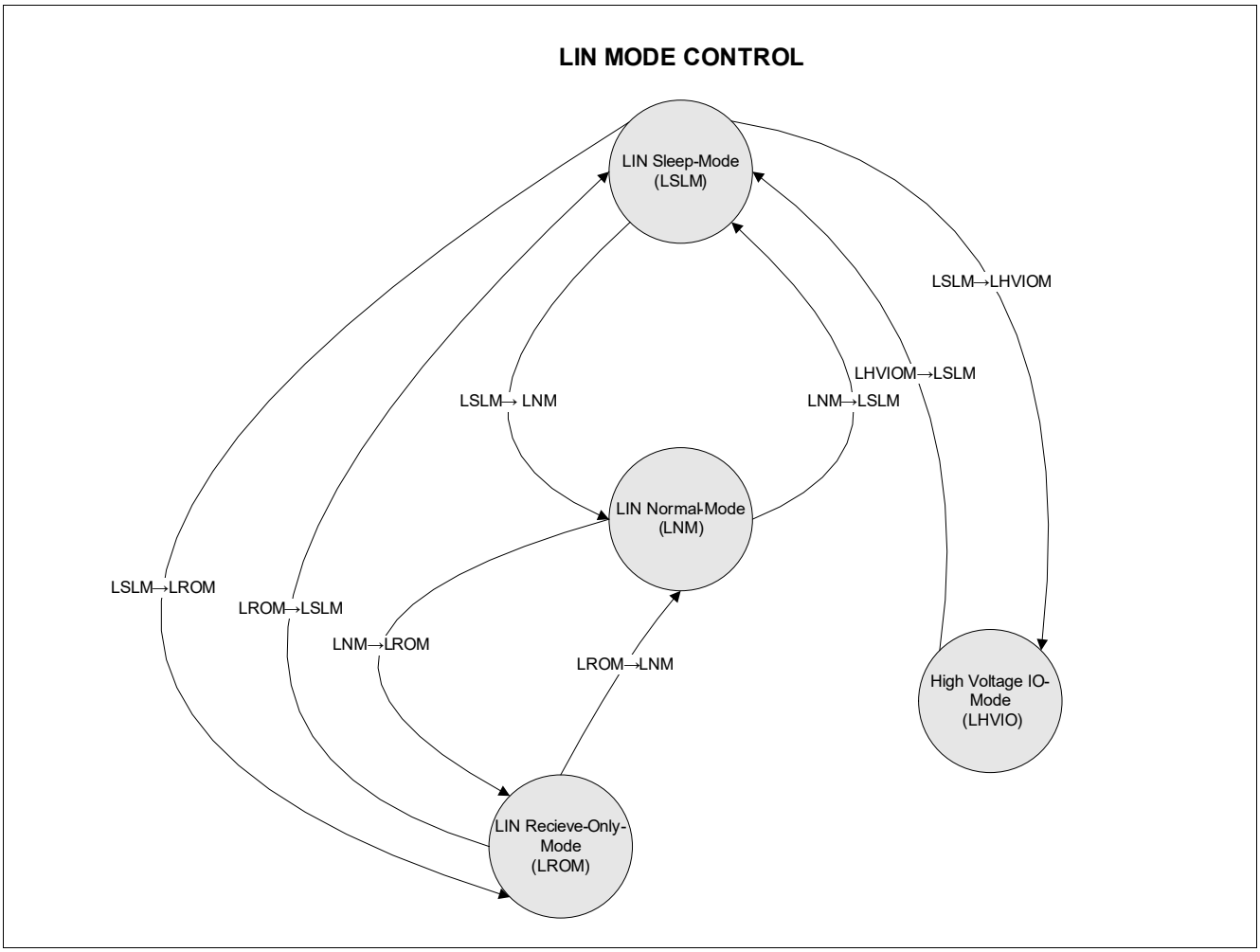


Figure 179 SFR controlled LIN Transceiver State machine

LIN Normal Mode (LNM)

In this mode it is possible to receive and transmit data with low slope, normal slope, fast slope or flash mode. Slope setting is locked during LIN Normal Mode to avoid destruction of communication process. This is blocked by hardware.

LIN Receive-Only Mode (LROM)

In LIN Receive-Only Mode the transmitter is disabled. The receiver is active. This mode can be directly selected by application software or is automatically set upon error detection.

LIN Sleep Mode (LSLM)

In this mode, the transmit and receive functions are disabled, the wake receiver is active. Minimum current consumption is achieved. Wake up via LIN is possible. To disable the wake capability via LIN, the LIN wake can be disabled within the PMU.

LIN High Voltage Input / Output (LHVIO)

This mode is dedicated for using the LIN transceiver as high voltage input/output (HV-GPIO), whereas high voltage refers to VS level. In LHVIO Mode the transceiver is controlled by **CTRL_STS**.TXD.
The Input status is controlled by **CTRL_STS**.TXD and can be read by **CTRL_STS**.RXD.

The transitions between the described states can only be executed when corresponding conditions are fulfilled. The detailed description of the transitions can be found below.

LIN Sleep Mode (LSLM) - LIN Receive-Only Mode (LROM) Transition Description

- LSLM - LROM transition is executed when:
  - **CTRL_STS**.MODE is configured to LIN Receive-Only Mode and
  - Feedback signals of Mode and Slope Mode are ok and
  - **CTRL_STS**.HV_Mode bit is not set
- LROM - LSLM transition is executed when:
  - **CTRL_STS**.MODE is configured to LIN Sleep Mode

LIN Sleep Mode (LSLM) - LIN Normal Mode (LNM) Transition Description

- LSLM - LNM transition is executed when:
  - **CTRL_STS**.MODE is configured to LIN Normal Mode and
  - Feedback signals of Mode and Slope Mode are ok and
  - **CTRL_STS**.HV_Mode bit is not set and
  - VS undervoltage flag (**SYS_SUPPLY_IRQ_STS**.VS_UV_STS) is not set
  - LIN transceiver OT_STS and OC_STS are not set and
  - no **CTRL_STS**.TXD_TMOUT is set
- LNM - LSLM transition is executed when:
  - **CTRL_STS**.MODE is configured LIN Sleep Mode

LIN Normal Mode (LNM) - LIN Receive-Only Mode (LROM) Transition Description

- LNM - LROM transition is executed when:
  - **CTRL_STS**.MODE is configured to LIN Receive-Only Mode or
  - Feedback signals of Mode and Slope Mode are not ok or
  - VS undervoltage flag (**SYS_SUPPLY_IRQ_STS**.VS_UV_STS) is set or
  - LIN transceiver OT_STS or **CTRL_STS**.OC_STS are set or
  - **CTRL_STS**.TXD_TMOUT is set
- LROM - LNM transition is executed when:
  - **CTRL_STS**.MODE is configured to LIN Normal Mode and
  - Feedback signals of Mode and Slope Mode are ok (see **Chapter 20.3.2**) and
  - VS undervoltage flag (**SYS_SUPPLY_IRQ_STS**.VS_UV_STS) is not set and
  - LIN transceiver **CTRL_STS**.OT_STS and **CTRL_STS**.OC_STS are not set and

- no **CTRL_STS**.TXD_TMOUT is set

LIN Sleep Mode (LSLM) - LIN High Voltage Input / Output Mode (LHVIO) Transition Description

- LSLM - LHVIO transition is executed when:
  - **CTRL_STS**.HV_MODE flag is set and
  - **CTRL_STS**.MODE is configured to LIN Normal Mode after **CTRL_STS**.HV_MODE flag was set and
  - Feedback signals of Mode and Slope Mode are ok and
  - LIN transceiver **CTRL_STS**.OT_STS and **CTRL_STS**.OC_STS are not set
- LHVIO - LSLM transition is executed when:
  - **CTRL_STS**.MODE is configured to LIN Sleep Mode and
  - **CTRL_STS**.HV_MODE flag is set or
  - Feedback signals of Mode and Slope Mode are not ok or
  - LIN transceiver **CTRL_STS**.OT_STS or **CTRL_STS**.OC_STS are set

LIN Specifications 1.3 and 2.0, 2.1

The LIN specification 2.0 is a superset of the 1.3 version offering some additional features. However, it is possible to use the LIN 1.3 slave node in a 2.0 node cluster, as long as the new features are not used. Vice versa it is possible to use a LIN 2.0 node in the 1.3 cluster without using the new features.

The latest version of the LIN specification 2.1 has no changes regarding the physical layer specification of LIN 2.0.

20.3.2 LIN Transceiver Status for Mode Selection

The LIN transceiver provides the possibility to monitor the on chip status through internally generated feedback signals. This provides additional protection functionality for the application to avoid wrong configuration of the transceiver, which may lead to a blocking of communication on the LIN Bus. The table below shows the decoding of feedback signals to check the current status of the transceiver.

Table 147 Decoding of Feedback Signals for LIN Transmitter Mode Settings

MODE_FB[2]	MODE_FB[1]	MODE_FB[0]	Remarks
0	0	0	Mode Error
0	0	1	LIN Sleep Mode
0	1	0	Mode Error
0	1	1	Mode Error
1	0	0	Mode Error
1	0	1	LIN Receive-Only Mode
1	1	0	Mode Error
1	1	1	LIN Normal Mode

A Mode Error indicates a problem in the LIN configuration. If that applies, check the LIN software configuration, and whenever this does not improve the mode feedback it is recommended to enter Sleep Mode.

20.3.3 LIN Transceiver Error Handling

The LIN Module provides error handling for three different cases:

LIN Transceiver TxD Timeout

If the internal UART TxD signal is dominant for the time $t > t_{timeout}$, the TxD timeout function deactivates the LIN transmitter output stage temporarily by entering the LIN Receive-Only Mode. The transceiver remains in recessive state. The TxD timeout function prevents the LIN bus from being blocked by a permanent low signal on the TxD pin, caused by a failure. The failure is stored in the TXD_TMOUT flag. The transmitter stage is activated again after the dominant timeout condition is removed and after the TXD_TMOUT flag is cleared by software.

Note: the TxD Timeout feature is automatically deactivated when the transceiver is set to LHVIO Mode.

LIN Transceiver Overcurrent

If the LIN transmitter detects an overcurrent condition $I > I_{BUS,sc}$, the LIN transceiver stays in LIN Normal Mode and the overcurrent status will be stored in the OC_STS flag. The short circuit current is limited to $I_{BUS,sc}$. The OC_STS flag can be cleared by software and will be set again as long as the above condition remains.

To generate an interrupt in case of LIN overcurrent detection, the corresponding interrupt can be enabled by setting the OC_IE in the SYS_IRQ_CTRL_1 register. This interrupt is routed to INTISR[10].

LIN Transceiver Overtemperature

If the LIN transmitter detects an overtemperature condition the transmitter will be deactivated temporarily by entering the LIN Receive-Only Mode. The transceiver remains in recessive state. The failure is stored in the LIN_OT_STS flag. The transmitter stage is activated again after the overtemperature condition is gone and after the LIN_OT_STS flag is cleared by software.

To generate an interrupt in case of LIN overtemperature detection, the corresponding interrupt can be enabled by setting the LIN_OT_IE in the SYS_IRQ_CTRL_1 register. This interrupt is routed to INTISR[10].

20.3.4 Slope Modes

The LIN Module provides some additional slope mode features which can be used for EoL (End of Line) programming or to reduce emission in case of usage of lower baudrates. The configurable slope modes are:

Normal Slope Mode

This mode is usually used to transmit and receive messages on the bus. The selected slew rate setting allows a transmission rate of up to 20 kilobaud.

Low Slope Mode

The usage of this mode is linked to a communication with lower baudrate. With this setting the emission of the transmitter can be reduced. The selected slew rate setting allows a transmission rate of up to 10.4 kilobaud.

Fast Slope Mode

In this mode it is also possible to transmit and receive messages on the bus. The selected slew rate setting allows a transmission rate of up to 40 kilobaud.

Flash Mode

In this mode it is possible to transmit and receive messages on the bus. Transmission rates of up to 115.2 kilobaud are allowed due the internal slew rate control. This mode can be used for EoL programming.

Changing Slope Modes

It is not possible to change slope modes when the module is operating in LIN Normal Mode because this would cause transmission errors. For example, to change from Normal Slope Mode to Flash Mode, proceed as follows:

- change to LIN Receive-Only Mode or LIN Sleep Mode;
- configure the desired slope mode; and
- go back to LIN Normal Mode.

20.3.5 LIN Transceiver Slope Mode Status

The LIN transceiver provides the possibility to monitor the on chip status of the slope control through internally generated feedback signals. The table shows the decoding of the feedback signals.

Table 148 Slope Mode Status

FB_SM3	FB_SM2	FB_SM1	Remarks
0	0	0	LIN module not enabled
0	0	1	Low Slope Mode
0	1	0	Normal Slope Mode
0	1	1	Fast Slope mode
1	0	0	Flash Mode
1	0	1	Slope Mode Error

Table 148 Slope Mode Status (cont'd)

FB_SM3	FB_SM2	FB_SM1	Remarks
1	1	0	Slope Mode Error
1	1	1	Slope Mode Error

20.4 Register Definition

Table 149 shows the module base and end addresses.

Table 149 Register Address Space

Module	Base Address	End Address	Note
LIN	$4801E000_H$	$4801FFFF_H$	

Table 150 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
CTRL_STS	LIN Transceiver Control and Status	00_H	0000 0000 0001 1xxx xxx0 0x10 $0000\ 0111_B$

The registers are addressed wordwise.

The LIN transceiver and the controlling finite state machine can be fully controlled by the following SFR Register.

LIN Transceiver Control and Status

The register is reset by RESET_TYPE_3.

This register is being initialized by the BootROM during the boot-up process to a value of 0058_9E07_H

CTRL_STS

LIN Transceiver Control and Status **(00_H)** **Reset Value: $0000\ 0000\ 0001\ 1xxx\ xxx0\ 0x10\ 0000\ 0111_B$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES							M_SM_ERR_CLR	RES	RES	HV_MODE	RES		MODE_FB		
r							w	r	r	rw	r		r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB_SM3	FB_SM2	FB_SM1	SM		RXD	TXD	RES	RES	TXD_TMOUT_STS	OC_STS	OT_STS	M_SM_ERR	MODE		RES
r	r	r	rw		r	rw	r	r	r	r	r	r	rw		r

Field	Bits	Type	Description
RES	31:25	r	**Reserved** Always read as 0
M_SM_ERR_CLR	24	w	**LIN Transceiver Mode or Slope Mode Error Clear** 0_B **Mode or Slope Mode Error**, Status not clear 1_B **Mode or Slope Mode Error**, Status clear
RES	23	r	**Reserved** Always read as 0
RES	22	r	**Reserved** Always read as 0
HV_MODE	21	rw	**LIN Transceiver High Voltage Input - Output Mode** 0_B **DISABLE**, High Voltage Mode Entry is disabled 1_B **ENABLE**, High Voltage Mode Entry is enabled
RES	20:19	r	**Reserved** Always read as “11”
MODE_FB	18:16	r	**Feedback Signals for LIN Transmitter Mode Settings** Coding see **Table 147**
FB_SM3	15	r	**Feedback Signal 3 for Slope Mode Setting** Coding see **Table 148**
FB_SM2	14	r	**Feedback Signal 2 for Slope Mode Setting** Coding see **Table 148**
FB_SM1	13	r	**Feedback Signal 1 for Slope Mode Setting** Coding see **Table 148**

Field	Bits	Type	Description
SM	12:11	rw	**LIN Transmitter Slope mode control** 00_B **Normal Slope Mode**, for max. 20 kilobaud 01_B **Fast Slope Mode**, for max. 40 kilobaud 10_B **Low Slope Mode**, for max. 10.4 kilobaud 11_B **Flash Mode**, for max. 150 kilobaud *Note: Slope Mode can not be changed in Normal Mode*
RXD	10	r	**Output Signal of Receiver** Can be used to monitor the Receiver Output
TXD	9	rw	**LIN Transmitter switch on (only used when HV_MODE is set)** 0_B **Pull Down LIN Line**, Transmitter is switched on 1_B **Pull Up Resistor is active**, Transmitter is switched off
RES	8	r	**Reserved** Always read as 0
RES	7	r	**Reserved** Always read as 0
TXD_TMOUT_STS	6	r	**LIN TXD time-out status** 0_B **NO_TIMEOUT**, no time-out occurred 1_B **TIMEOUT**, time-out occurred
OC_STS	5	r	**LIN Receiver Overcurrent Status** 0_B **no Overcurrent**, overcurrent status occurred 1_B **Overcurrent**, overcurrent status occurred
OT_STS	4	r	**LIN Receiver Overtemperature Status** 0_B **no Overtemperature**, overtemperature occurred 1_B **Overtemperature**, overtemperature occurred
M_SM_ERR	3	r	**LIN Transceiver Mode or Slope Mode Error** 0_B **no Mode or Slope Mode Error**, Status (see corresponding feedback bits) 1_B **Mode or Slope Mode Error**, Status (see corresponding feedback bits)
MODE	2:1	rw	**LIN transceiver power mode control** 00_B **LIN Sleep Mode**, LIN module switched to LIN Sleep Mode 01_B **LIN Receive-Only Mode**, LIN module switched to LIN Receive Only Mode 10_B **n.u.**, not used 11_B **LIN Normal Mode**, LIN module switched to LIN Normal Mode
RES	0	r	**Reserved** Always read as 1

20.5 Register Access Hints

The register **CTRL_STS**, in the LIN module is clocked with the MI_CLK. Write operations will be delayed by one MI_CLK cycle.

For handling the clearing if interrupt status flags special care might be necessary. Under certain coding constellations an extra write access to the interrupt clear flag will be necessary to add the extra MI_CLK cycle.

In case of interrupt status clear, two use cases in the software can be considered:

1. the writing to the interrupt clear flag and the reading of the corresponding interrupt status flag are having some other code in between
2. the writing to the interrupt clear flag and the reading of the corresponding interrupt status flag are placed in code right behind each other, with no code in between

Use Case 1 - Example, no extra write attempt to the interrupt clear register required

```
/* clearing of LIN Trx. Mode/Slope Mode Error status flag */
LIN->CTRL_STS.bit.M_SM_ERR_CLR = 1;
/* some other code is placed here */
...
/* reading of LIN Trx. Mode/Slope Mode Error status flag */
if ( LIN->CTRL_STS.bit.M_SM_ERR == 1 )
{
 /* some user code here */
}
```

In this example, between the interrupt clear register write and the corresponding status register read, sufficient time is ensured, as some C code (at least one C instruction) is placed in between.

Use Case 2- Example, one extra write attempt to the interrupt clear register required

```
/* clearing of LIN Trx. Mode/Slope Mode Error status flag */
LIN->CTRL_STS.bit.M_SM_ERR_CLR = 1;
/* place another interrupt status flag clear attempt to add an extra MI_CLK cycle*/
LIN->CTRL_STS.bit.M_SM_ERR_CLR = 0;
/* reading of LIN Trx. Mode/Slope Mode Error status flag */
if ( LIN->CTRL_STS.bit.M_SM_ERR == 1 )
{
 /* some user code here */
}
```

In this example, between the interrupt clear register write and the corresponding status register read, another interrupt clear register write is placed. In order to add another MI_CLK cycle and ensure that the first interrupt status flag write attempt has reach the interrupt status register before it will be read again in the if clause right after it. Without this extra clear attempt the following interrupt status flag read might still reflect the interrupt status set.

An extra instruction gives the first clear attempt at least one MI_CLK cycle of time. Any other C instruction might be sufficient as well. Using the same interrupt clear write again is proposed. This access cannot be eliminated by the compiler by optimization.

Executing the second interrupt clear write with a ‘0’ is recommended, as so, no functional effect to the register takes place.

21 High-speed synchronous serial interface (SSC1/SSC2)

21.1 Features

- Master and Slave mode operation
  - Full-duplex or half-duplex operation
- Transmit- and receive-buffered
- Flexible data format
  - Programmable number of data bits: 2 to 16 bits
  - Programmable shift direction: least significant bit (LSB) or most significant bit (MSB) shift first
  - Programmable clock polarity: idle low or high state for the shift clock
  - Programmable clock/data phase: data shift with leading or trailing edge of the shift clock
- Variable baud rate
- Compatible with Serial Peripheral Interface (SPI)
- Interrupt generation
  - On a “transmitter empty” condition
  - On a “receiver full” condition
  - On an error condition (receive, phase, baud rate, or transmission error)

21.2 Introduction

The high-speed synchronous serial interface (SSC) supports both full-duplex and half-duplex serial synchronous communication. The serial clock signal can be generated by the SSC internally (Master mode), using its own 16-bit baud rate generator, or can be received from an external master (Slave mode). Data width, shift direction, clock polarity, and phase are programmable. This allows communication with SPI-compatible devices as well as devices using other synchronous serial interfaces.

Data is transmitted or received on the TXD and RXD lines, which are normally connected to the MTSR (Master Transmit, Slave Receive) and MRST (Master Receive, Slave Transmit) pins. The clock signal is output via the MS_CLK (Master Serial Shift Clock) line or input via the SS_CLK (Slave Serial Shift Clock) line. Both lines are normally connected to the SCLK pin. Transmission and reception of data are double-buffered.

21.2.1 Block diagram

Figure 180 shows all functionally relevant interfaces associated with the SSC kernel.

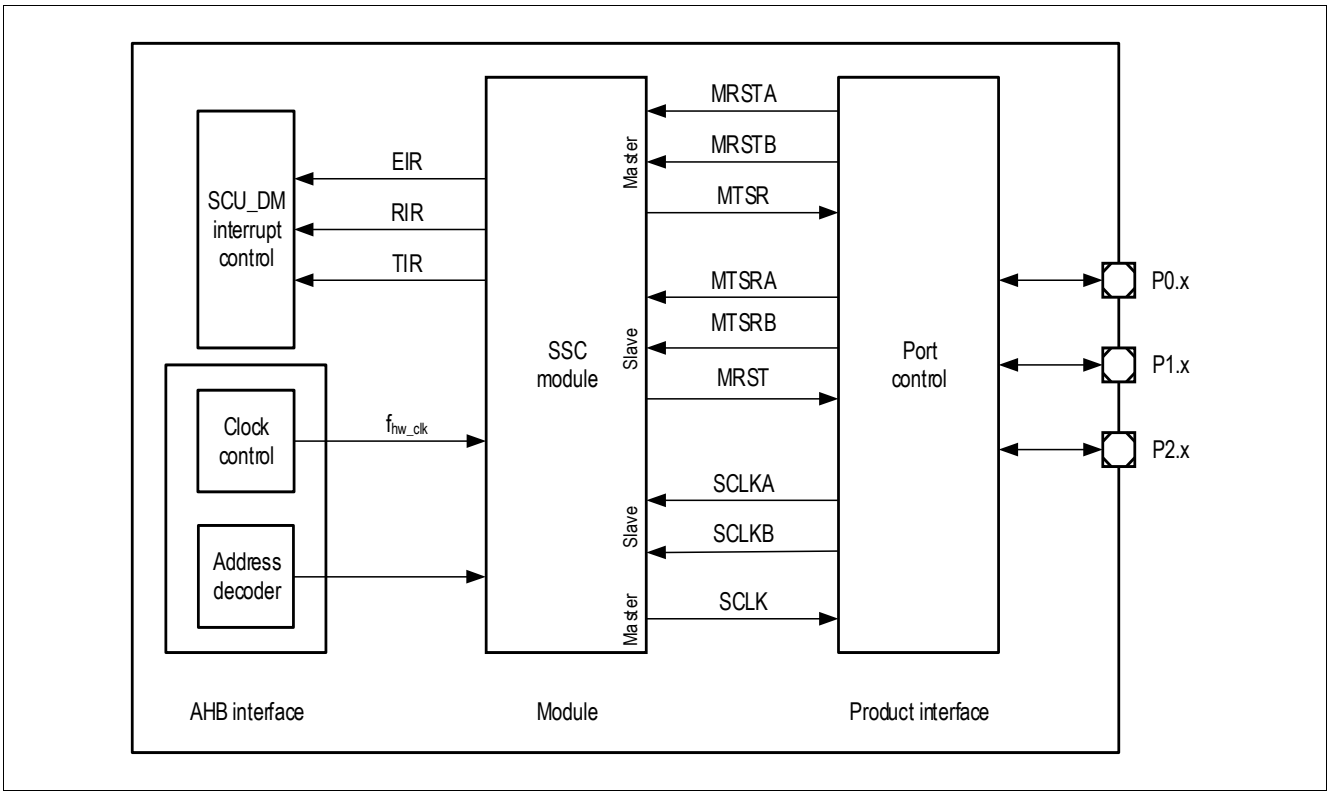


Figure 180 SSC interface diagram

21.3 Functional Description

21.3.1 SSC1 and SSC2 Modes Overview

The SSC supports full-duplex and half-duplex synchronous communication up to 20 MBaud (@ 40 MHz module clock). The serial clock signal can be generated by the SSC itself (Master Mode) or can be received from an external master (Slave Mode). Data width, shift direction, clock polarity, and phase are programmable. This allows communication with SPI-compatible devices. Transmission and reception of data is double-buffered. A 16-bit baud-rate generator provides the SSC with a separate serial clock signal.

The SSC can be configured in a very flexible way, so it can be used with other synchronous serial interfaces, can serve for master/slave or multimaster interconnections or can operate compatible with the popular SPI interface. Thus, the SSC can be used to communicate with shift registers (I/O expansion), peripherals (e.g. EEPROMs, etc.) or other controllers (networking). The SSC supports half-duplex and full-duplex communication. Data is transmitted or received on lines TXD and RXD, normally connected with pins MTSR (Master Transmit/Slave Receive) and MRST (Master Receive/Slave Transmit). The clock signal is output via line MS_CLK (Master Serial Shift Clock) or input via line SS_CLK (Slave Serial Shift Clock). Both lines are normally connected to pin SCLK.

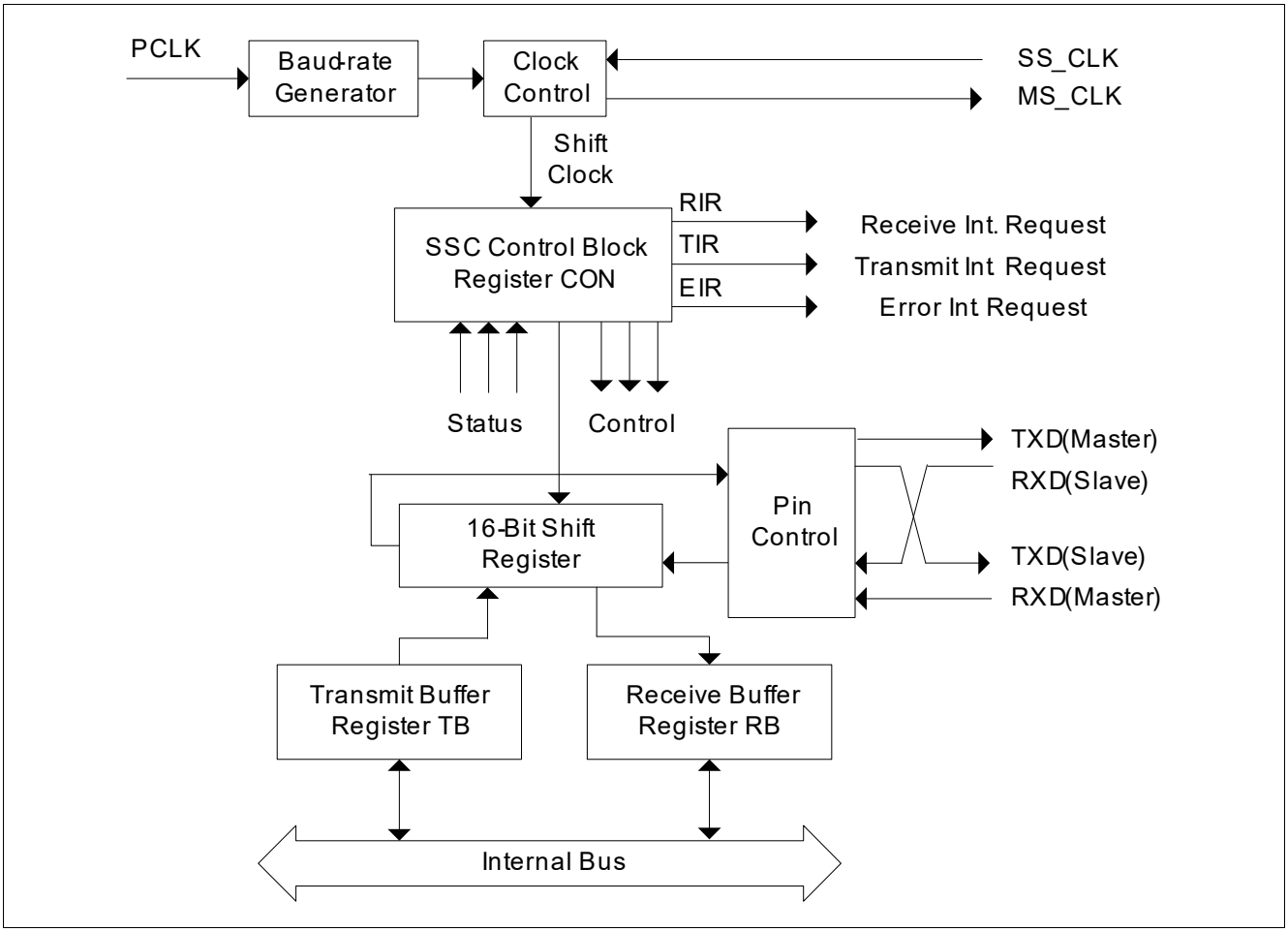


Figure 181 Synchronous Serial Channel SSC Block Diagram

21.3.2 Operating Mode Selection

The operating mode of the serial channel SSC is controlled by its control register CON. This register serves two purposes:

- During programming (SSC disabled by CON.EN = 0), it provides access to a set of control bits
- During operation (SSC enabled by CON.EN = 1), it provides access to a set of status flags.

The shift register of the SSC is connected to both the transmit lines and the receive lines via the pin control logic (see block diagram in **Figure 181**). Transmission and reception of serial data are synchronized and take place at the same time, i.e. the same number of transmitted bits is also received. Transmit data is written into the Transmit Buffer (TB) and is moved to the shift register as soon as this is empty. An SSC master (CON.MS = 1) immediately begins transmitting, while an SSC slave (CON.MS = 0) will wait for an active shift clock. When the transfer starts, the busy flag CON.BSY is set and the Transmit Interrupt Request line TIR will be activated to indicate that register TB may be reloaded again. When the programmed number of bits (2 … 16) has been transferred, the contents of the shift register are moved to the Receive Buffer RB and the Receive Interrupt Request line RIR will be activated. If no further transfer is to take place (TB is empty), CON.BSY will be cleared at the same time. Software should not modify CON.BSY, as this flag is hardware controlled.

Note: The SSC starts transmission and sets CON.BSY minimum two clock cycles after transmit data is written into TB. Therefore, it is not recommended to poll CON.BSY to indicate the start and end of a single transmission. Instead, interrupt service routine should be used if interrupts are enabled, or the interrupt flags IRCON1.TIR and IRCON1.RIR should be polled if interrupts are disabled.

Note: Only one SSC (etc.) can be master at a given time.

The transfer of serial data bits can be programmed in many respects:

- The data width can be specified from 2 bits to 16 bits
- A transfer may start with either the LSB or the MSB
- The shift clock may be idle low or idle high
- The data bits may be shifted with the leading edge or the trailing edge of the shift clock signal
- The baud rate may be set from 305.18 Baud up to 20 MBaud (@ 40 MHz module clock)
- The shift clock can be generated (MS_CLK) or can be received (SS_CLK)

These features allow the adaptation of the SSC to a wide range of applications requiring serial data transfer.

The Data Width Selection supports the transfer of frames of any data length, from 2-bit “characters” up to 16-bit “characters”. Starting with the LSB (CON.HB = 0) allows communication with SSC devices in Synchronous Mode or with 8051 like serial interfaces for example. Starting with the MSB (CON.HB = 1) allows operation compatible with the SPI interface.

Regardless of the data width selected and whether the MSB or the LSB is transmitted first, the transfer data is always right-aligned in registers TB and RB, with the LSB of the transfer data in bit 0 of these registers. The data bits are rearranged for transfer by the internal shift register logic. The unselected bits of TB are ignored; the unselected bits of RB will not be valid and should be ignored by the receiver service routine.

The Clock Control allows the adaptation of transmit and receive behavior of the SSC to a variety of serial interfaces. A specific shift clock edge (rising or falling) is used to shift out transmit data, while the other shift clock edge is used to latch in receive data. Bit CON.PH selects the leading edge or the trailing edge for each function. Bit CON.PO selects the level of the shift clock line in the idle state. Thus, for an idle-high clock, the leading edge is a falling one, a 1-to-0 transition (see **Figure 182**).

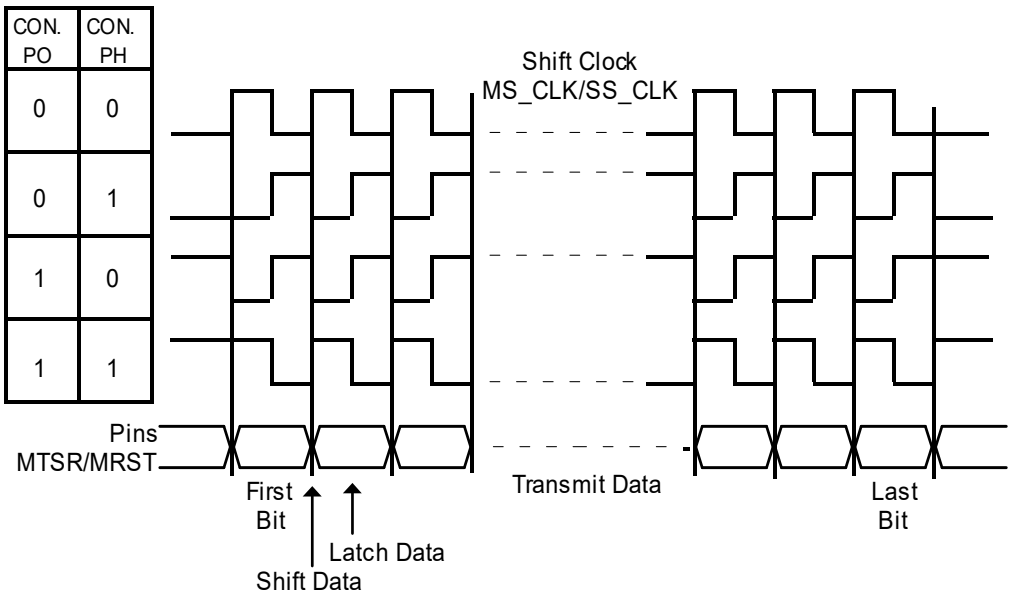


Figure 182 Serial Clock Phase and Polarity Options

21.3.3 Full-Duplex Operation

The various devices are connected through three lines. The definition of these lines is always determined by the master: the line connected to the master's data output line TXD is the transmit line; the receive line is connected to its data input line RXD; the shift clock line is either MS_CLK or SS_CLK. Only the device selected for master operation generates and outputs the shift clock on line MS_CLK. Since all slaves receive this clock, their pin SCLK must be switched to input mode. The output of the master's shift register is connected to the external transmit line, which in turn is connected to the slaves' shift register input. The output of the slaves' shift register is connected to the external receive line in order to enable the master to receive the data shifted out of the slave. The external connections are hard-wired, the function and direction of these pins is determined by the master or slave operation of the individual device.

Note: The shift direction shown in the figure applies for MSB-first operation as well as for LSB-first operation.

When initializing the devices in this configuration, one device must be selected for master operation while all other devices must be programmed for slave operation. Initialization includes the operating mode of the device's SSC and also the function of the respective port lines.

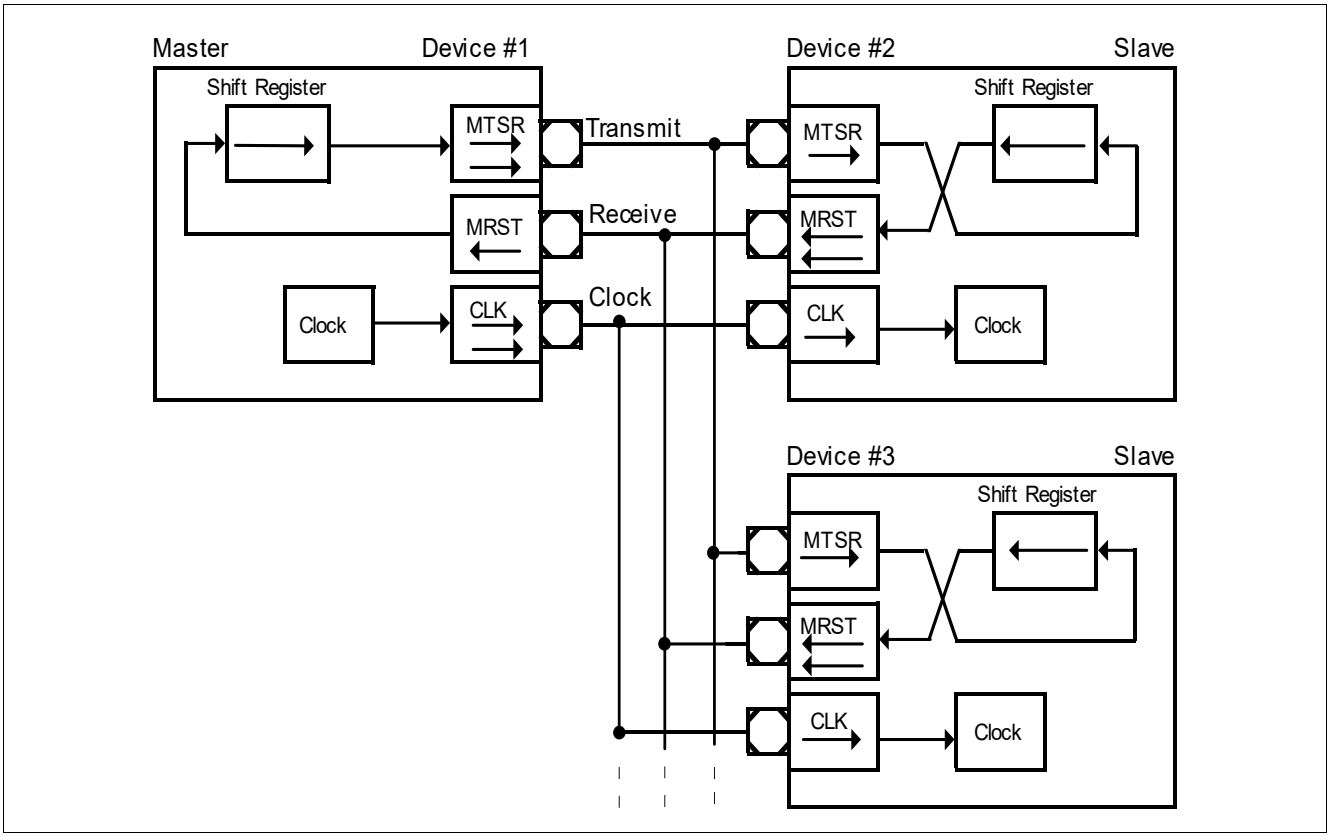


Figure 183 SSC Full-Duplex Configuration

The data output pins MRST of all slave devices are connected together onto the one receive line in the configuration shown in **Figure 183**. During a transfer, each slave shifts out data from its shift register. There are two ways to avoid collisions on the receive line due to different slave data:

- Only one slave drives the line, i.e. enables the driver of its MRST pin. All the other slaves must have their MRST pins programmed as input so only one slave can put its data onto the master's receive line. Only receiving data from the master is possible. The master selects the slave device from which it expects data either by separate select lines, or by sending a special command to this slave. The selected slave then switches its MRST line to output until it gets a de-selection signal or command.
- The slaves use open drain output on MRST. This forms a wired-AND connection. The receive line needs an external pull-up in this case. Corruption of the data on the receive line sent by the selected slave is avoided when all slaves not selected for transmission to the master only send ones (1s). Because this high level is not actively driven onto the line, but only held through the pull-up device, the selected slave can pull this line actively to a low-level when transmitting a zero bit. The master selects the slave device from which it expects data either by separate select lines or by sending a special command to this slave.

After performing the necessary initialization of the SSC, the serial interfaces can be enabled. For a master device, the alternate clock line will now go to its programmed polarity. The alternate data line will go to either 0 or 1 until the first transfer starts. After a transfer, the alternate data line will always remain at the logic level of the last transmitted data bit.

When the serial interfaces are enabled, the master device can initiate the first data transfer by writing the transmit data into register TB. This value is copied into the shift register (assumed to be empty at this time), and the selected first bit of the transmit data will be placed onto the TXD line on the next clock from the baud-rate generator (transmission starts only if CON.EN = 1). Depending on the selected clock phase, a clock pulse will also be generated on the MS_CLK line. At the same time, with the opposite clock edge, the master latches and shifts in the data detected at its input line RXD. This “exchanges” the transmit data with the receive data.

Because the clock line is connected to all slaves, their shift registers will be shifted synchronously with the master's shift register — shifting out the data contained in the registers, and shifting in the data detected at the input line. After the preprogrammed number of clock pulses (via the data width selection), the data transmitted by the master is contained in all the slaves' shift registers, while the master's shift register holds the data of the selected slave. In the master and all slaves, the contents of the shift register are copied into the receive buffer RB and the receive interrupt line RIR is activated.

A slave device will immediately output the selected first bit (MSB or LSB of the transfer data) at line RXD when the contents of the transmit buffer are copied into the slave's shift register. Bit CON.BSY is not set until the first clock edge at SS_CLK appears. The slave device will not wait for the next clock from the baud-rate generator, as the master does. The reason for this is that, depending on the selected clock phase, the first clock edge generated by the master may already be used to clock in the first data bit. Thus, the slave's first data bit must already be valid at this time.

Note: On the SSC, a transmission **and** *a reception takes place at the same time, regardless of whether valid data has been transmitted or received.*

Note: The initialization of the CLK pin on the master requires some attention in order to avoid undesired clock transitions, which may disturb the other devices. Before the clock pin is switched to output via the related direction control register, the clock output level will be selected in the control register CON and the alternate output be prepared via the related ALTSEL register, or the output latch must be loaded with the clock idle level.

21.3.4 Half-Duplex Operation

In a Half-Duplex Mode, only one data line is necessary for both receiving **and** transmitting of data. The data exchange line is connected to both the MTSR and MRST pins of each device, the shift clock line is connected to the SCLK pin.

The master device controls the data transfer by generating the shift clock, while the slave devices receive it. Due to the fact that all transmit and receive pins are connected to the one data exchange line, serial data may be moved between arbitrary stations.

Similar to Full-Duplex Mode, there are two ways to avoid collisions on the data exchange line:

- Only the transmitting device may enable its transmit pin driver
- The non-transmitting devices use open drain output and send only ones.

Because the data inputs and outputs are connected together, a transmitting device will clock in its own data at the input pin (MRST for a master device, MTSR for a slave). By this method, any corruptions on the common data exchange line are detected if the received data is not equal to the transmitted data.

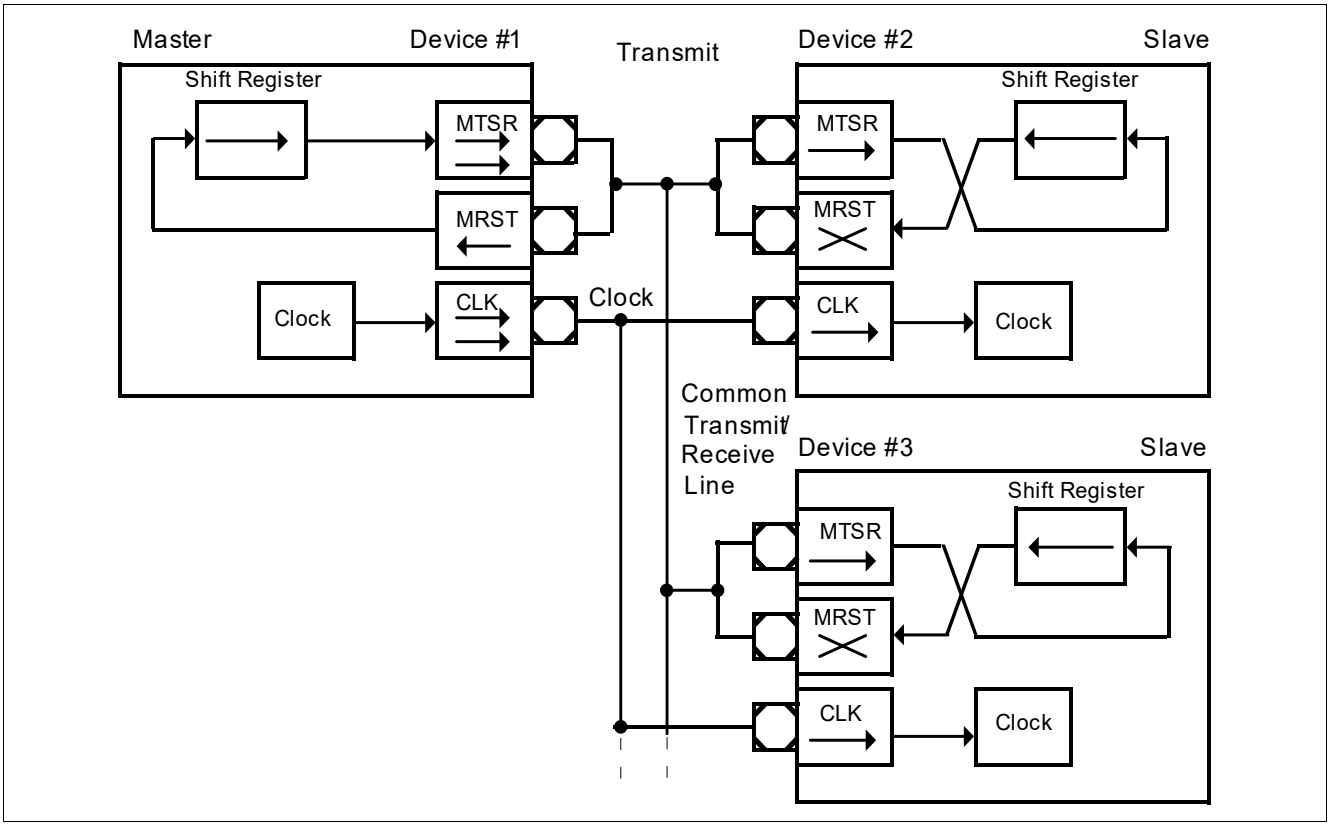


Figure 184 SSC Half-Duplex Configuration

21.3.5 Continuous Transfers

When the transmit interrupt request flag is set, it indicates that the transmit buffer TB is empty and ready to be loaded with the next transmit data. If TB has been reloaded by the time the current transmission is finished, the data is immediately transferred to the shift register and the next transmission will start without any additional delay. On the data line, there is no gap between the two successive frames. For example, two byte transfers would look the same as one word transfer. This feature can be used to interface with devices that can operate with or require more than 8 data bits per transfer. It is just a matter of software, how long a total data frame length can be. This option can also be used to interface to byte-wide and word-wide devices on the same serial bus, for instance.

Note: Of course, this can happen only in multiples of the selected basic data width, because it would require disabling/enabling of the SSC to reprogram the basic data width on-the-fly.

21.3.6 Baud Rate Generation

The serial channel SSC has its own dedicated 16-bit baud-rate generator with 16-bit reload capability, allowing baud rate generation independent of the timers. In addition to **Figure 181**, **Figure 185** shows the baud-rate generator of the SSC in more detail.

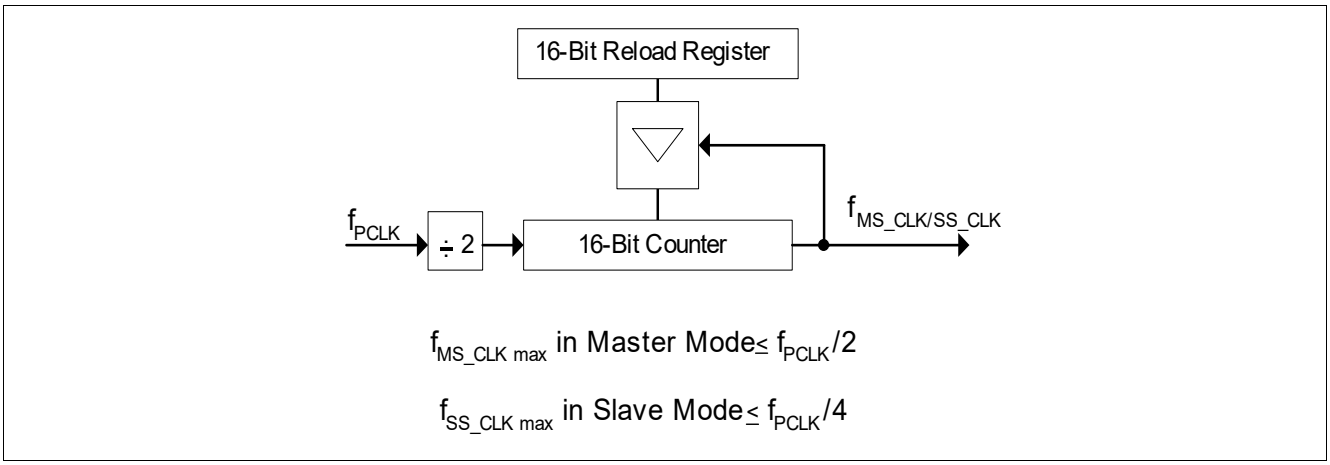


Figure 185 SSC Baud-rate Generator

The baud-rate generator is clocked with the module clock f_{hw_clk}. The timer counts downwards. Register BR is the dual-function Baud-rate Generator/Reload register. Reading BR, while the SSC is enabled, returns the contents of the timer. Reading BR, while the SSC is disabled, returns the programmed reload value. In this mode, the desired reload value can be written to BR.

Note: Never write to BR while the SSC is enabled.

The formulas below calculate either the resulting baud rate for a given reload value, or the required reload value for a given baud rate:

$$\text{Baud rate} = \frac{f_{hw_clk}}{2 * (\langle BR \rangle + 1)} \qquad BR = \frac{f_{hw_clk}}{2 * \text{Baud rate}} - 1 \tag{21.1}$$

 represents the contents of the reload register, taken as an unsigned 16-bit integer, while baud rate is equal to f_{MS_CLK/SS_CLK} as shown in **Figure 185**.

The maximum baud rate that can be achieved when using a module clock of 40 MHz is 20 MBaud in Master Mode (with
 = 0000_H) or 10 MBaud in Slave Mode (with
 = 0001_H).

Table 151 lists some possible baud rates together with the required reload values and the resulting bit times, assuming a module clock of 40 MHz.

Table 151 Typical Baud Rates of the SSC (f_{hw_clk} = 40 MHz)

Reload Value	Baud Rate (= f_{MS_CLK/SS_CLK})	Deviation
0000_H	20 MBaud (only in Master Mode)	0.0%
0001_H	10 MBaud	0.0%
0013_H	1 MBaud	0.0%
0027_H	500 kBaud	0.0%
$00C7_H$	100 kBaud	0.0%
$07CF_H$	10 kBaud	0.0%
$4E1F_H$	1 kBaud	0.0%
$FFFF_H$	305.18 Baud	0.0%

21.3.7 Error Detection Mechanisms

The SSC is able to detect four different error conditions. Receive Error and Phase Error are detected in all modes; Transmit Error and Baud Rate Error apply only to Slave Mode. When an error is detected, the respective error flag is/can be set and an error interrupt request will be generated by activating the EIR line (see **Figure 186**) if enabled. The error interrupt handler may then check the error flags to determine the cause of the error interrupt. The error flags are not reset automatically but must be cleared by software after servicing. This allows servicing of some error conditions via interrupt, while the others may be polled by software.

Note: The error interrupt handler must clear the associated (enabled) error flag(s) to prevent repeated interrupt requests.

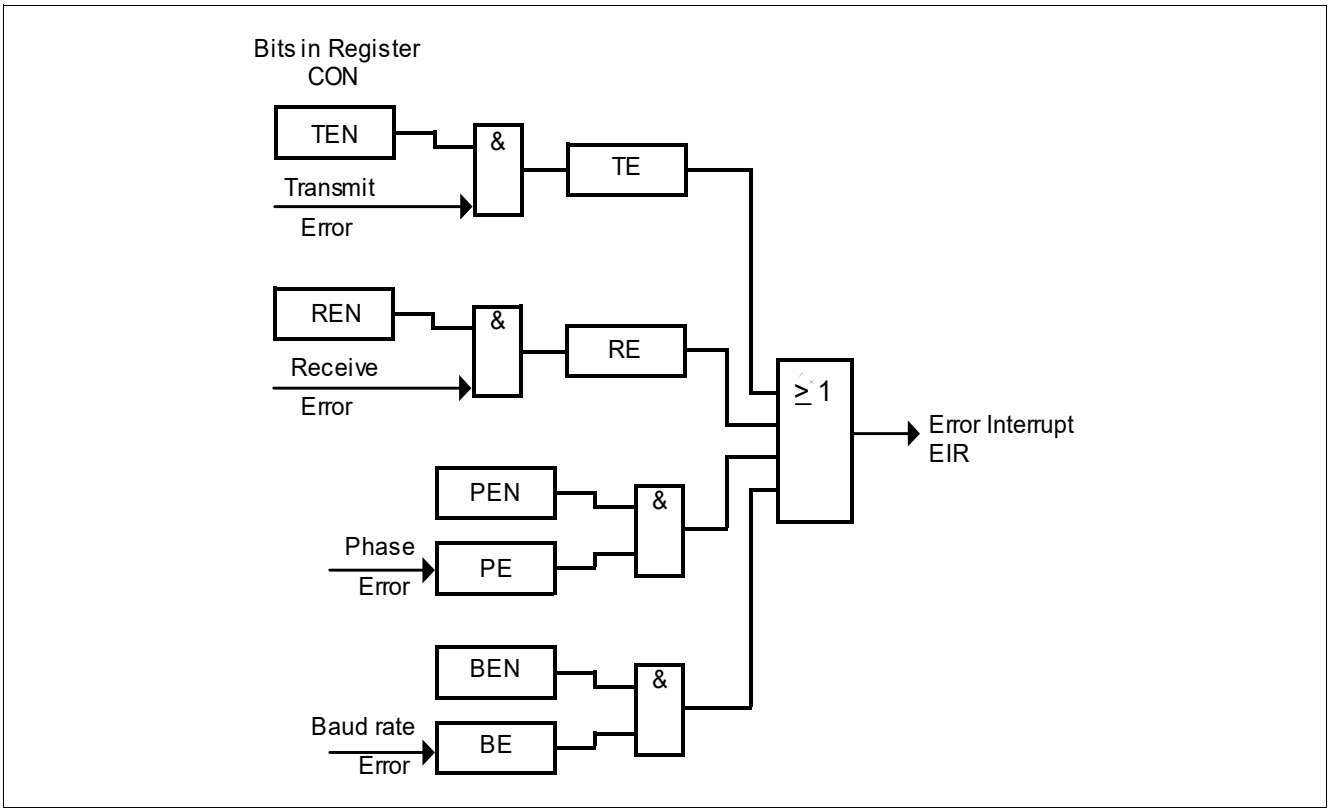


Figure 186 SSC Error Interrupt Control

A **Receive Error** (Master or Slave Mode) is detected when a new data frame is completely received but the previous data was not read out of the receive buffer register RB. This condition sets the error flag CON.RE and the error interrupt request line EIR, when enabled via CON.REN. The old data in the receive buffer RB will be overwritten with the new value and is irretrievably lost.

A **Phase Error** (Master or Slave Mode) is detected when the incoming data at pin MRST (Master Mode) or MTSR (Slave Mode), sampled with the same frequency as the module clock, changes between one cycle before and two cycles after the latching edge of the shift clock signal SCLK. This condition sets the error flag CON.PE and, when enabled via CON.PEN, the error interrupt request line EIR.

Note: When receiving and transmitting data in parallel, phase errors occur if the baud rate is configured to $f_{hw_clk}/2$.

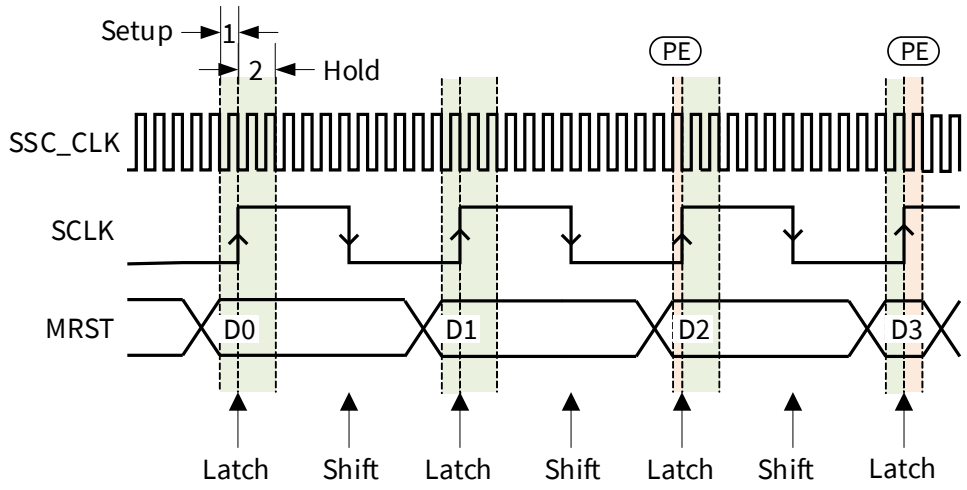


Figure 187 Example of an SSC phase error

A **Baud Rate Error** (Slave Mode) is detected when the incoming clock signal deviates from the programmed baud rate by more than 100%, i.e. it is either more than double or less than half the expected baud rate. This condition sets the error flag CON.BE and, when enabled via CON.BEN, the error interrupt request line EIR. Using this error detection capability requires that the slave's baud-rate generator is programmed to the same baud rate as the master device. This feature detects false additional, or missing pulses on the clock line (within a certain frame).

Note: If this error condition occurs and bit CON.REN = 1, an automatic reset of the SSC will be performed in case of this error. This is done to re-initialize the SSC if too few or too many clock pulses have been detected.

Note: This error can occur after any transfer if the communication is stopped. This is the case due to the fact that the SSC module supports back-to-back transfers for multiple transfers. In order to handle this, the baud rate detector expects after a finished transfer immediately a next clock cycle for a new transfer.

A **Transmit Error** (Slave Mode) is detected when a transfer was initiated by the master (SS_CLK gets active) but the transmit buffer TB of the slave was not updated since the last transfer. This condition sets the error flag CON.TE and the error interrupt request line EIR, when enabled via CON.TEN. If a transfer starts while the

transmit buffer is not updated, the slave will shift out the 'old' contents of the shift register, which normally is the data received during the last transfer. This may lead to corruption of the data on the transmit/receive line in half-duplex mode (open drain configuration) if this slave is not selected for transmission. This mode requires that slaves not selected for transmission only shift out ones; that is, their transmit buffers must be loaded with '$FFFF_H$' prior to any transfer.

Note: A slave with push/pull output drivers not selected for transmission, will normally have its output drivers switched. However, in order to avoid possible conflicts or misinterpretations, it is recommended to always load the slave's transmit buffer prior to any transfer.

The cause of an error interrupt request (receive, phase, baud rate, transmit error) can be identified by the error status flags in control register CON.

Note: In contrast to the error interrupt request line EIR, the error status flags CON.TE, CON.RE, CON.PE, and CON.BE, are not reset automatically upon entry into the error interrupt service routine, but must be cleared by software.

21.3.7.1 Port Control

The SSC uses three lines to communicate with the external world. Pin SCLK serves as the clock line, while pins MRST (Master Receive/Slave Transmit) and MTSR (Master Transmit/Slave Receive) serve as the serial data input/output lines. As shown in **Figure 180** these three lines (SCLK as input, Master Receive, Slave Receive) have all two inputs at the SSC Module kernel. Three bits in register PISEL define which of the two kernel inputs (A or B) are connected. This feature allows for each of the three SSC communication lines to be connected to two inputs coming from different port pins.

Operation of the SSC I/O lines depends on the selected operating mode (master or slave). The direction of the port lines depends on the operating mode. The SSC will automatically use the correct kernel output or kernel input line of the ports when switching modes. Port pins assigned as SSC I/O lines can be controlled in two ways:

- By hardware
- By software

When the SSC I/O lines are connected with dedicated pins typically hardware I/O control should be used. In this case, the two output signals reflect directly the state of the CON.EN and CON.MS bits (the M/S select line is inverted to the CON.MS bit definition).

When the SSC I/O lines are connected with bidirectional lines of general purpose I/O ports, typically software I/O control should be used. In this case port registers must be programmed for alternate output and input selection. When switching between master and slave mode, port registers must be reprogrammed.

21.4 Module Interfaces

This section describes:

- The SSC module related interfaces such as port connections and interrupt control
- All SSC module related registers with its addresses

21.4.1 Interfaces of the SSC Module

An overview of the SSC kernel I/O interface is shown in **Figure 188** (SSC1) and **Figure 189** (SSC2).

The interrupt requests of the SSC are not connected directly to the Interrupt Controller, but via the System Control Unit (SCU). The interrupt request signals of the SSC are mapped to the Interrupt Controller by the SCU.

*Note: Please refer to **Chapter 6** for the SCU description.*

The General Purpose IO (GPIO) Port provides the interface from the SSC to the external world. There are two SSC kernels in the TLE987x, namely SSC1 and SSC2.

*Note: Please refer to **Chapter 14** for the Ports description.*

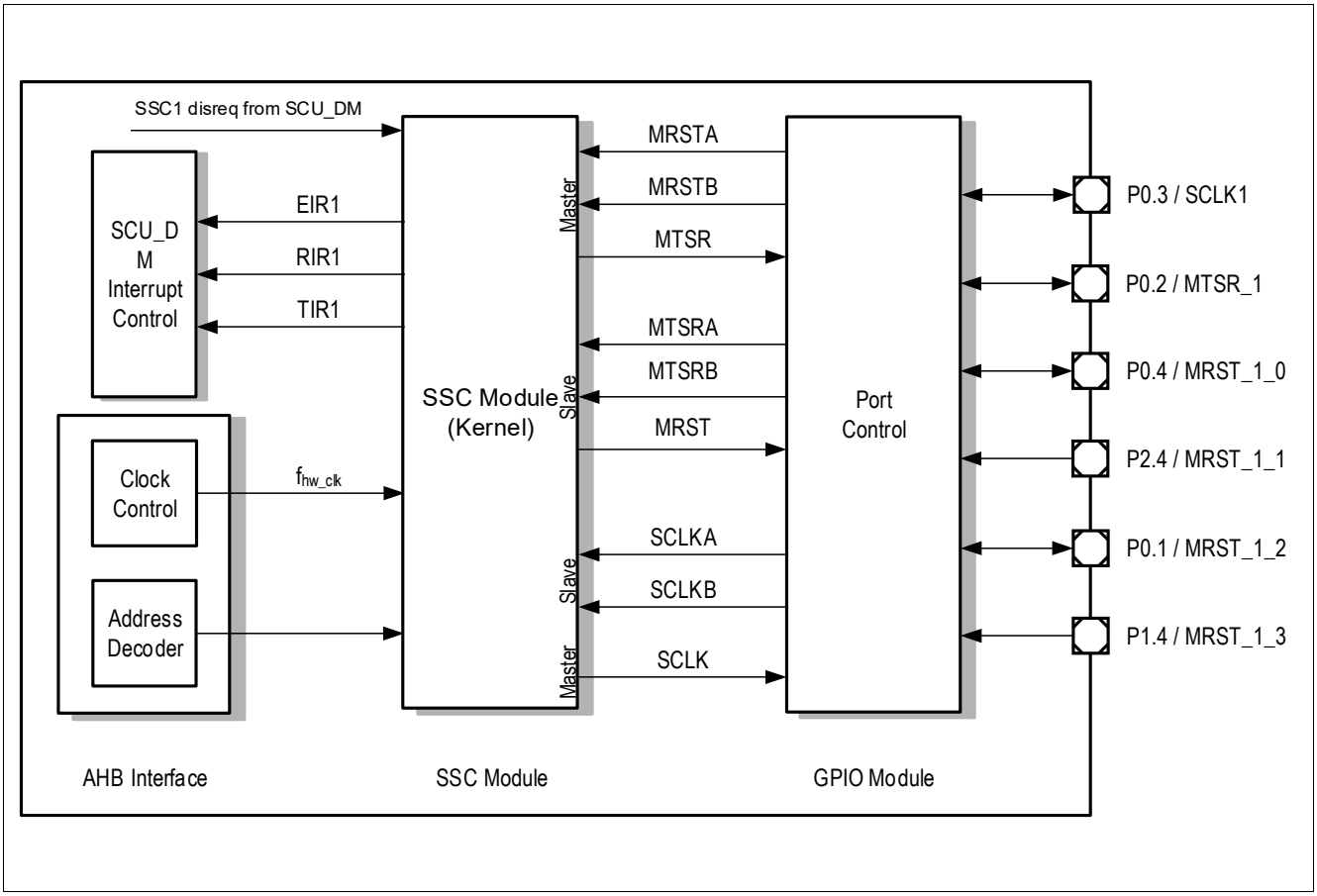


Figure 188 SSC1 Module I/O Interface

Table 152 SSC1 Interface

MIS_1	MIS_0	Description
0	0	P0.4 (MRST_1_0)
0	1	P2.4 (MRST_1_1)
1	0	P0.1 (MRST_1_2)
1	1	P1.4 (MRST_1_3)

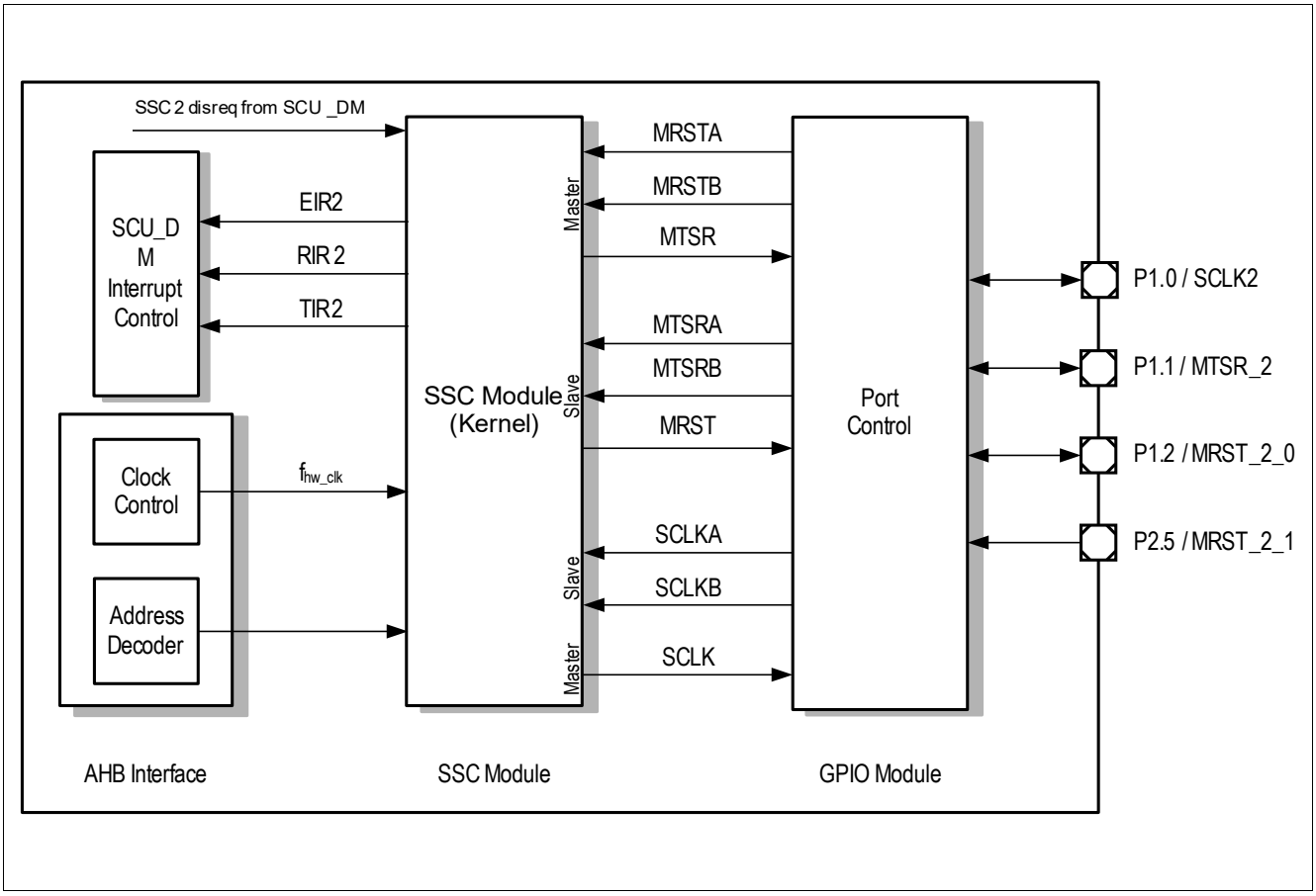


Figure 189 SSC2 Module I/O Interface

21.5 Interrupts

The three SSC interrupts can be separately enabled or disabled by setting or clearing their corresponding enable bits in SFR SCU_MODIEN.

For a detailed description of the various interrupts see **Section 21.4**. An overview is given in **Table 153**.

Table 153 SSC Interrupt Sources

Interrupt	Signal	Description
Transmission starts	TIR	Indicates that the transmit buffer can be reloaded with new data.
Transmission ends	RIR	The configured number of bits have been transmitted and shifted to the receive buffer.
Receive Error	EIR	This interrupt occurs if a new data frame is completely received and the last data in the receive buffer was not read.
Phase Error	EIR	This interrupt is generated if the incoming data changes between one cycle before and two cycles after the latching edge of the shift clock signal SCLK.
Baud Rate Error (Slave Mode only)	EIR	This interrupt is generated when the incoming clock signal deviates from the programmed baud rate by more than 100%.
Transmit Error (Slave Mode only)	EIR	This interrupt is generated when TB was not updated since the last transfer if a transfer is initiated by a master.

21.6 Register Definition

21.6.1 SSC1 and SSC2 Register Definition

21.6.1.1 Port Input Select Register

The PISEL register controls the receiver input selection of the SSC module.

PISEL
Port Input Select Register, RESET_TYPE_3 (00_H) **Reset Value: 00_H**

7	6	5	4	3	2	1	0
RES				MIS_1	CIS	SIS	MIS_0
r				rw	rw	rw	rw

Field	Bits	Type	Description
MIS_0	0	rw	**Master Mode Receiver Input Select** 0_B see **Table 152 "SSC1 Interface" on Page 670** (SSC1). 1_B see **Table 152 "SSC1 Interface" on Page 670** (SSC1). 0_B Receiver input (Port A: P1.2) is selected (SSC2). 1_B Receiver input (Port B: P2.5) is selected (SSC2).
SIS	1	rw	**Slave Mode Receiver Input Select** 0_B Receiver input (Port A: P0.2) is selected (SSC1). 1_B Receiver input (Port B: P0.2) is selected (SSC1). 0_B Receiver input (Port A: P1.1) is selected (SSC2). 1_B Receiver input (Port B: P1.1) is selected (SSC2).
CIS	2	rw	**Slave Mode Clock Input Select** 0_B Clock input (Port A: P0.3) is selected (SSC1). 1_B Clock input (Port B: P0.3) is selected (SSC1). 0_B Clock input (Port A: P1.0) is selected (SSC2). 1_B Clock input (Port B: P1.0) is selected (SSC2).
MIS_1	3	rw	**Master Mode Receiver Input Select** 0_B see **Table 152 "SSC1 Interface" on Page 670** (SSC1). 1_B see **Table 152 "SSC1 Interface" on Page 670** (SSC1). 0_B n/a (SSC2). 1_B n/a (SSC2).
RES	7:4	r	**Reserved** Returns 0 if read; should be written with 0.

Note: Port A and Port B inputs of the SSC kernel are connected to the external pins of Port 0 and Port 1 respectively.

21.6.1.2 Configuration Register

The operating mode of the serial channel SSC is controlled by the control register CON. This register contains control bits for mode and error check selection, and status flags for error identification. Depending on bit EN, either control functions or status flags and master/slave control are enabled.

CON.EN = 0: Programming Mode

CON
Control Register **(04_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	**MS**	**RES**	**AREN**	**BEN**	**PEN**	**REN**	**TEN**	**LB**	**PO**	**PH**	**HB**	**BM**			
rw	rw	r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

Field	Bits	Type	Description
BM	3:0	rw	**Data Width Selection** 0000_BReserved. Do not use this combination. 0001_B- 1111_BTransfer Data Width is 2 … 16 bits (BM+1).
HB	4	rw	**Heading Control** 0_B Transmit/Receive LSB First. 1_B Transmit/Receive MSB First.
PH	5	rw	**Clock Phase Control** 0_B Shift transmit data on the leading clock edge, latch on trailing edge. 1_B Latch receive data on leading clock edge, shift on trailing edge.
PO	6	rw	**Clock Polarity Control** 0_B Idle clock line is low, leading clock edge is low-to-high transition. 1_B Idle clock line is high, leading clock edge is high-to-low transition.
LB	7	rw	**Loop Back Control** 0_B Normal output. 1_B Receive input is connected with transmit output (half-duplex mode).
TEN	8	rw	**Transmit Error Enable** 0_B Ignore transmit errors. 1_B Check transmit errors.
REN	9	rw	**Receive Error Enable** 0_B Ignore receive errors. 1_B Check receive errors.
PEN	10	rw	**Phase Error Enable** 0_B Ignore phase errors. 1_B Check phase errors.

Field	Bits	Type	Description
BEN	11	rw	**Baud Rate Error Enable** 0_B Ignore baud rate errors. 1_B Check baud rate errors.
AREN	12	rw	**Automatic Reset Enable** 0_B No additional action upon a baud rate error. 1_B The SSC is automatically reset upon a baud rate error.
RES	13	r	**Reserved** Returns 0 if read; should be written with 0.
MS	14	rw	**Master Select** 0_B Slave Mode. Operate on shift clock received via SCLK. 1_B Master Mode. Generate shift clock and output it via SCLK.
EN	15	rw	**Enable Bit = 0** Transmission and reception disabled. Access to control bits. *Note: The effect of EN bit becomes visible on the next write to the CON register.*

CON.EN = 1: Operating Mode

CON
Control Register **(04_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	**MS**	**RES**	**BSY**	**BE**	**PE**	**RE**	**TE**	**RES**				**BC**			
rw	rw	r	rh	r	r	r	r	r				rh			

Field	Bits	Type	Description
BC	3:0	rh	**Bit Count Field** Shift counter is updated with every shift bit. *Note: This bit field is not to be written to.*
RES	7:4	r	**Reserved** Returns 0 if read; should be written with 0.
TE	8	r	**Transmit Error Flag** 0_B No error. 1_B Transfer starts with the slave's transmit buffer not being updated.
RE	9	r	**Receive Error Flag** 0_B No error. 1_B Reception completed before the receive buffer was read.
PE	10	r	**Phase Error Flag** 0_B No error. 1_B Received data changes around sampling clock edge.
BE	11	r	**Baud Rate Error Flag** 0_B No error. 1_B More than factor 2 or 0.5 between slave's actual and expected baud rate.
BSY	12	rh	**Busy Flag** Set while a transfer is in progress. *Note: This bit is not to be written to.*
RES	13	r	**Reserved** Returns 0 if read; should be written with 0.
MS	14	rw	**Master Select Bit** 0_B Slave Mode. Operate on shift clock received via SCLK. 1_B Master Mode. Generate shift clock and output it via SCLK.
EN	15	rw	**Enable Bit = 1** Transmission and reception enabled. Access to status flags and M/S control. *Note: The effect of EN bit becomes visible on the next write to the CON register.*

ISRCLR
Interrupt Status Register Clear **(14_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				BECLR	PECLR	RECLR	TECLR	RES							
r				w	w	w	w	r							

Field	Bits	Type	Description
RES	7:0	r	**Reserved** Returns 0 if read; should be written with 0.
TECLR	8	w	**Transmit Error Flag Clear** 0_B No error clear. 1_B Error clear.
RECLR	9	w	**Receive Error Flag Clear** 0_B No error clear. 1_B Error clear.
PECLR	10	w	**Phase Error Flag Clear** 0_B No error clear. 1_B Error clear.
BECLR	11	w	**Baud Rate Error Flag Clear** 0_B No error clear. 1_B Error clear.
RES	15:12	r	**Reserved** Returns 0 if read; should be written with 0.

21.6.1.3 Baud Rate Timer Reload Register

The SSC baud rate timer reload register BR contains the 16-bit reload value for the baud rate timer.

BR
Baud Rate Timer Reload Register **(10_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR_VALUE															
rw															

Field	Bits	Type	Description
BR_VALUE	15:0	rw	**Baud Rate Timer/Reload Register Value** Reading BR returns the 16-bit contents of the baud rate timer. Writing BR loads the baud rate timer reload register with BR_VALUE. Never write to BR while the SSC is enabled via CON.EN = 1.

21.6.1.4 Transmitter Buffer Register

The SSC transmitter buffer register TB contains the transmit data value.

TB
Transmitter Buffer Register **(08_H)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TB_VALUE															
rw															

Field	Bits	Type	Description
TB_VALUE	15:0	rw	**Transmit Data Register Value** TB_VALUE is the data value to be transmitted. Unselected bits of TB are ignored during transmission.

21.6.1.5 Receiver Buffer Register

The SSC receiver buffer register RB contains the receive data value.

RB
Receiver Buffer Register **($0C_H$)** **Reset Value: 0000_H**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB_VALUE															
rh															

Field	Bits	Type	Description
RB_VALUE	15:0	rh	**Receive Data Register Value** RB contains the received data value RB_VALUE. Unselected bits of RB will be not valid and should be ignored.

21.6.2 Register Map

There are two SSC kernels in the TLE987x, namely SSC1 and SSC2. **Table 154** shows the SSC module base addresses.

Table 155 lists the addresses of the SSC SFRs.

Table 154 Register Address Space

Module	Base Address	End Address	Note
SSC1	48024000_H	$48025FFF_H$	
SSC2	48026000_H	$48027FFF_H$	

Table 155 Register Overview SSC

Register Short Name	Register Long Name	Offset Address	Page Number
PISEL	Port Input Select Register	00_H	**673**
CON	Control Register	04_H	**674**
TB	Transmitter Buffer Register	08_H	**678**
RB	Receiver Buffer Register	$0C_H$	**679**
BR	Baud Rate Timer Reload Register	10_H	**678**
ISRCLR	Interrupt Status Clear Register	14_H	**677**

22 Measurement unit

22.1 Features

- 1 x 8-bit ADC with 10 inputs. Attenuators allow measuring high-voltage input signals.
- Supply voltage attenuators for attenuating **VS**, **VDDP** and **VDDC**.
- VBG monitoring of the 8-bit ADC to guarantee functional safety requirements.
- Bridge driver diagnosis measurement (VDH, VCP).
- Temperature sensor for monitoring the chip temperature and PMU regulator temperature.
- BEMF comparators for triggering commutation in BLDC applications.
- Supplement block with reference voltage generation, bias current generation, voltage buffer for NVM reference voltage, voltage buffer for analog module reference voltage, and a test interface.

22.2 Introduction

The measurement unit is a functional unit that comprises the following submodules:

Table 156 Measurement functions and associated modules

Module name	Module	Functions
Central function unit	Bandgap reference circuit	The bandgap reference submodule provides two reference voltages: 1. A trimmable reference voltage for the 8-bit ADC. A local dedicated bandgap circuit ensures that the reference voltage does not drop, e.g., because of crosstalk or ground voltage shift. 2. The reference voltage for the NVM module.
Reference voltage (V_{AREF})	10-bit ADC reference voltage circuit	Generates the 5-V reference voltage for the 10-bit ADC and can be switched off when an external reference voltage source is used.
8-bit ADC (ADC2)	8-bit ADC module with 10 multiplexed inputs and including high-voltage input attenuators	• 5 high-voltage inputs supporting the full supply range (2.5 V...30.7 V(FS)) • 2 medium-voltage inputs (0..5 V/7 V FS). • 3 low-voltage inputs (0..1.2 V/1.6 V FS) (See the following figure for the allocation of the inputs).
10-bit ADC (ADC1)	10-bit ADC module with 8 multiplexed inputs	Five (5 V) analog inputs from port 2.x.
14 Bit ADCs[1)] (ADC3, ADC4)	14 Bit Sigma Delta ADC module	Two differential analog inputs from Port 2.x
VDH input voltage attenuator	VDH input voltage attenuator	Scales down V(VDH) to the input voltage range of ADC1.CH6.

Table 156 Measurement functions and associated modules (cont'd)

Module name	Module	Functions
Temperature sensor	Temperature sensor with two multiplexed sensing elements: • Sensor located on the PMU • Sensor located on the central chip	Generates an output voltage that is a linear function of the local chip (junction) temperature.
BEMF comparators	Back electromotive force comparators	Comparators are used to detect the back electromotive force (zero-crossing event), which can be used as a commutation trigger for BLDC applications.
Core measurement module	Digital signal processing and ADC2 control unit	1. Generates the control signal for the 8-bit ADC2 and the synchronous clock for the switched capacitor circuits. 2. Performs digital signal processing functions and provides status outputs for interrupt generation.

1) Available only for products with SDADC. See the datasheet for details.

22.2.1 Block diagram

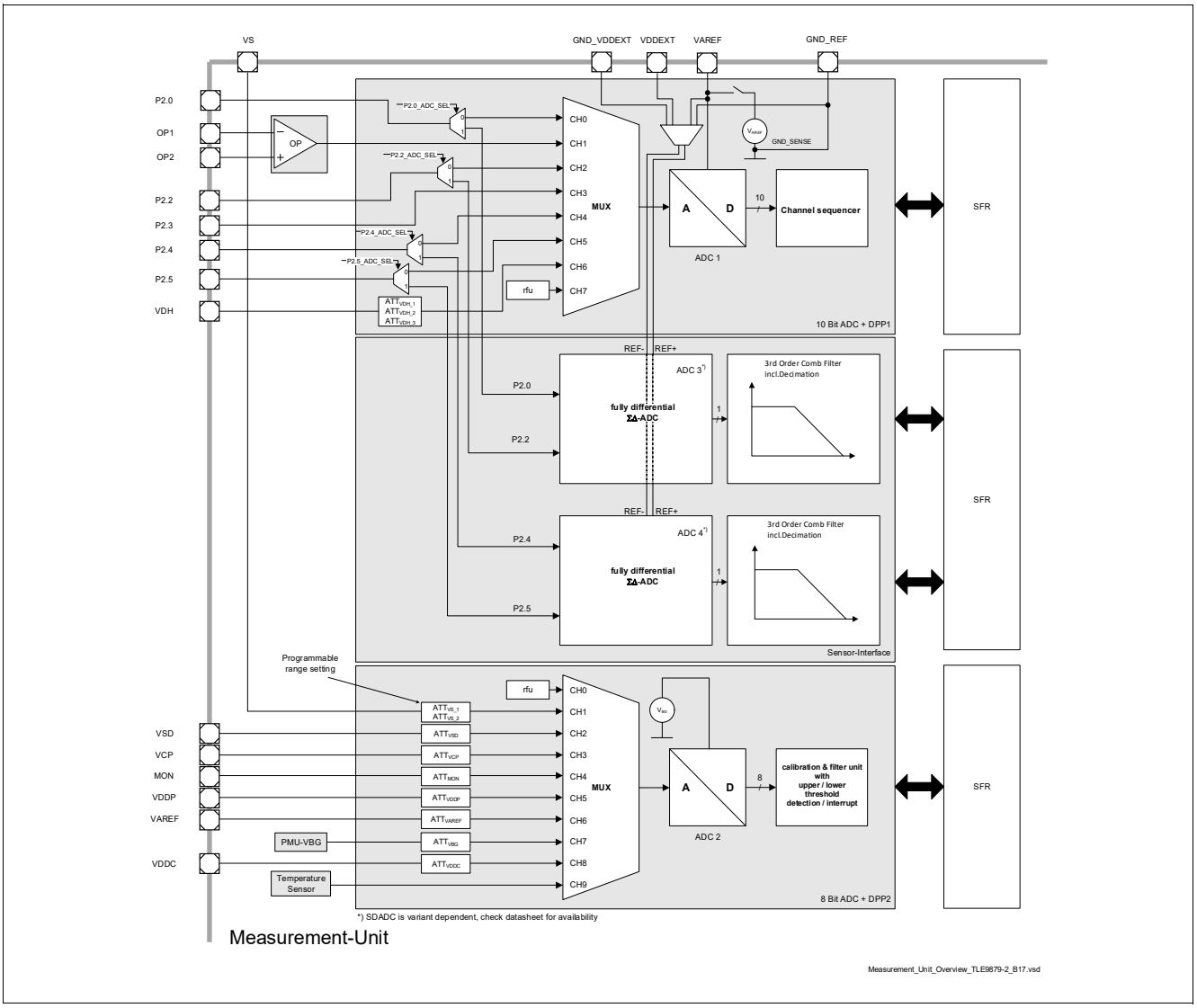


Figure 190 Measurement Unit-Overview

22.2.1.1 BEMF comparator block diagram

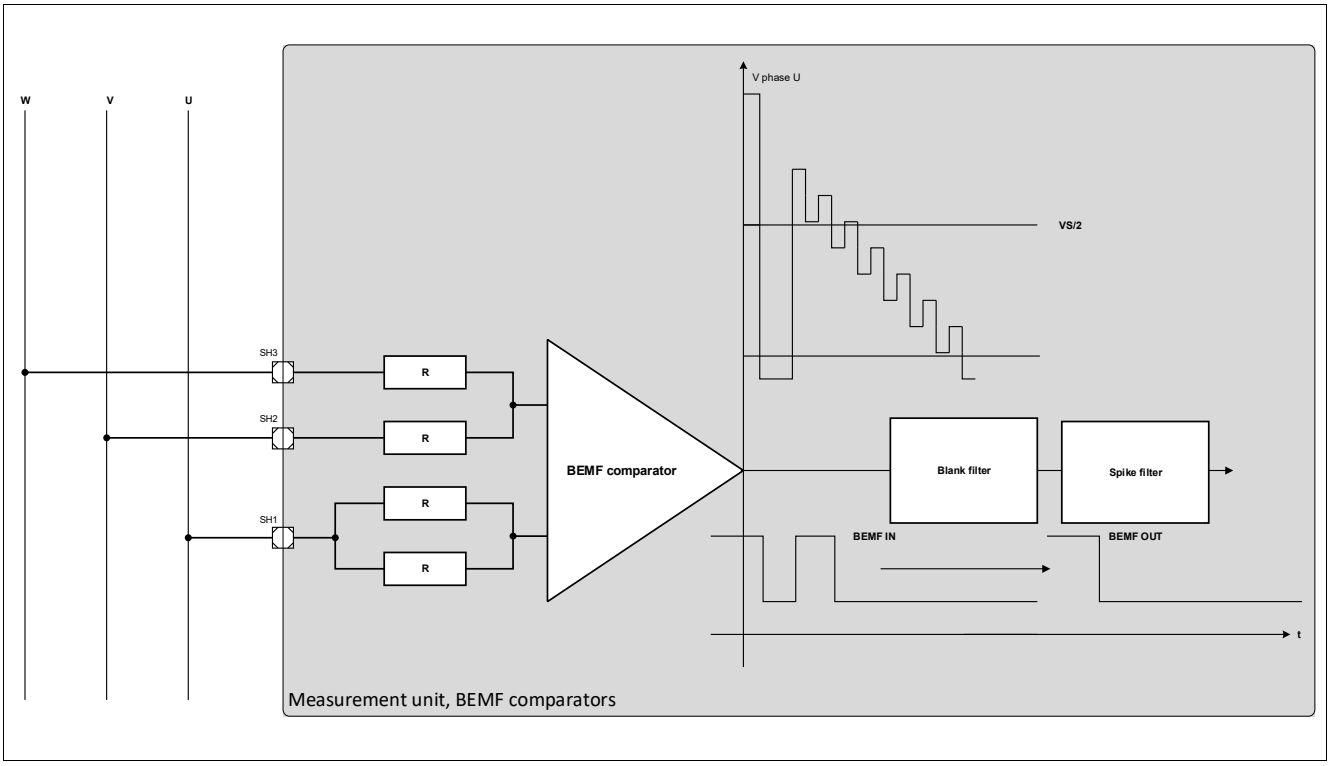


Figure 191 BEMF comparator (applies to each of the three comparators)

22.2.2 Measurement Unit Register Overview

All registers of blocks that belong to the measurement unit are mapped to base address see below.

Table 157 Register Address Space for Measurement Unit Registers

Module	Base Address	End Address	Note
MF	48018000_H	$4801BFFF_H$	Measurement Unit

Table 158 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Port 2 ADC Selection Register			
P2_ADCSEL_CTRL	Port 2 ADC Selection Control Register	00_H	$0000\ 0400_H$
VDH Input Attenuator Control Register			
VMON_SEN_CTRL	Monitoring Supply Sense Control Register	04_H	$0000\ 0000_H$
BEMF Comparators Control and Status Register			
BEMFC_CTRL_STS	BEMF Comparator Control Status Register	08_H	$0000\ 0000_H$
TRIM_BEMFx	Trimming of BEMF comparators	20_H	$0000\ 0000_H$
Temperature Sensor Control Register			
TEMPSENSE_CTRL	Temperature Sensor Control Register	10_H	$0000\ 0003_H$
Supplement Modules Control and Status Register			
REF1_STS	Reference 1 Status Register	14_H	$0000\ 00C1_H$
REF2_CTRL	Reference 2 Control Register	18_H	$0000\ 0001_H$

The registers are addressed wordwise.

22.3 8-bit - 10 Channel ADC Core (ADC2)

The 8-bit ADC Core operates at the VDDC Supply Voltage. This enables the user to operate the measurement system down to VS reset threshold. The ADC2 can also be operated independently from the DPP (Data Post Processing) unit. This enables the user to build up a software controlled measurement cycle. The main features of the 8-bit ADC core are listed below.

Module Features

- Conversion time = 15 system clock cycles.
- programmable sampling time (4 to 22 MI_CLK cycles, default: 18 MI_CLK cycles)
- Scalable clock frequency from 10 - 30 MHz.

The ADC2 is controlled in two different ways by the control module DPP2:

- Fully controlled by the sequencer inside DPP2.
- Partly controlled by the user within defined timeslots of Sequencer (EIM, Exceptional Interrupt Measurement)

22.3.1 8-bit ADC Channel Allocation

The allocation of the 10 Channels of ADC2 is sketched below:

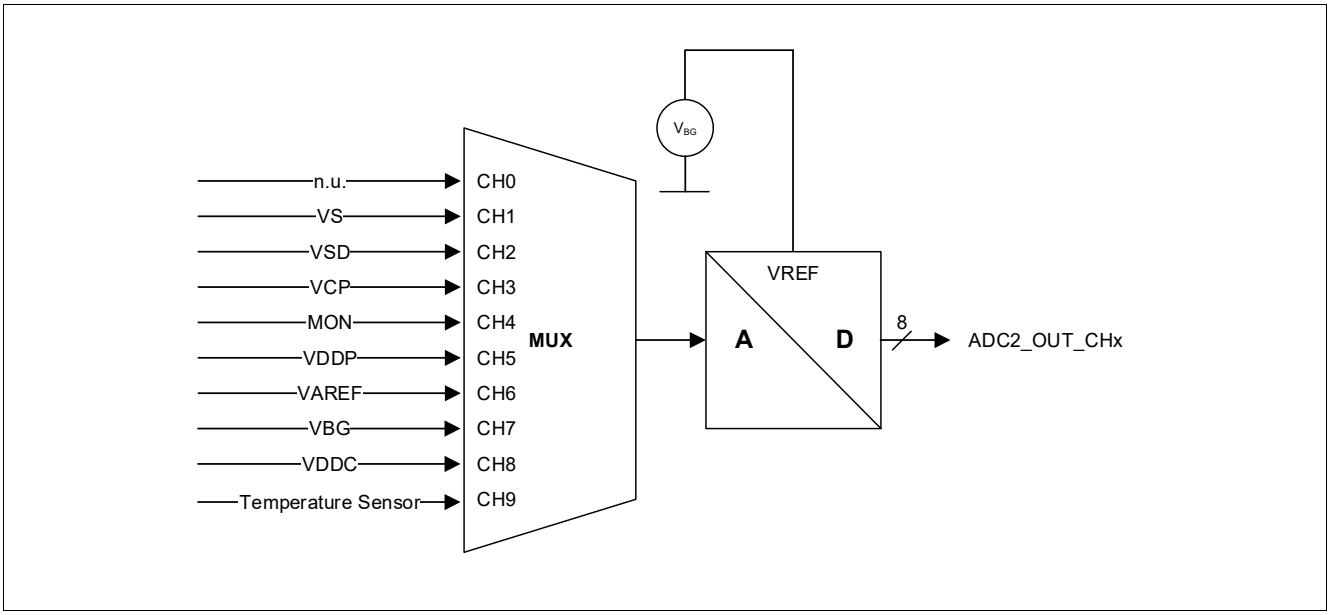


Figure 192 8-bit ADC (ADC2) Channel Allocation

ADC2 Channel Assignment:

- VS Pin Voltage Measurement.
- VSD Pin Voltage Measurement.
- Charge Pump Output Voltage (VCP).
- MON Pin Voltage Measurement
- VDDP Pin Voltage Measurement.
- ADC1 Reference Voltage Check (VAREF).
- ADC2 Reference Voltage Check (VBG).
- VDDC Pin Voltage Measurement.
- Device Temperature Measurement (T_j).

22.3.2 Transfer Characteristics of ADC2

The transfer function of ADC2 can be expressed by the equation below:

$$\text{ADC2out} = \text{floor}\left(\frac{V_{inCHx} * Gain_{CHx}}{V_{LSB}} + 1\right) \tag{22.1}$$

where Vin is the input voltage and $Gain_{CHx}$ the individual Channel Gain. The LSB Voltage is calculated:

$$V_{LSB} = \frac{V_{BG}}{2^8 - 1} = \frac{1.211V}{255} = 4.749mV \tag{22.2}$$

where ***typ.*** $\boldsymbol{V_{BG}}$ is **1.211 V @ 27 °C.** *Typ.* V_{BG} = 1.211V results in $\boldsymbol{V_{LSB}}$ **= 4.749 mV**:

A detailed specification of both A/D-converters is given in Chapter **Electrical Characteristics in the Data Sheet**.

22.3.3 8-bit - 10 Channel Control Registers

The ADC2 control registers are located in the **Measurement Core Module** Block (see **Chapter 23**). Also the configurable attenuator factors are located there (see register CTRL_STS).

22.4 ADC Input Selection

22.4.1 Functional Description

The 10-bit ADC (ADC1) and the SDADC[1)] are sharing Port 2.x inputs. The configuration is shown in Chapter 22.2.1

22.4.2 Port 2 ADC Selection Register

Table 159 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Port 2 ADC Selection Register			
P2_ADCSEL_CTRL	Port 2 ADC Selection Control Register	00_H	$0000\ 0400_H$

The registers are addressed bytewise.

Port 2 ADC Selection Control Register

The register is reset by RESET_TYPE_3.

This register is being initialized by the BootROM during the boot-up process to a value of $0000\ 0400_H$ for products with CSA. In the product without CSA the reset value is $0000\ 0000_H$.

P2_ADCSEL_CTRL

Port 2 ADC Selection Control Register **(00_H)** **Reset Value: $0000\ 0400_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					**ADC1_CH1_SEL**	**ADC3_INN_SEL**	**ADC3_INP_SEL**	**RES**		**RES**	**P2_5_ADC_SEL**	**P2_4_ADC_SEL**	**P2_3_ADC_SEL**	**P2_2_ADC_SEL**	**P2_0_ADC_SEL**
r					rw	rw	rw	r		r	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:11	r	**Reserved** Always read as 0
ADC1_CH1_SEL	10	rw	**ADC1 Channel 1 Input Selection** 0_B **none** , Current Sense Amplifier disconnected from ADC1.Ch1 1_B **CS_AMP**, Current Sense Amplifier is connected to ADC1.Ch1

1) SDADC is not available in all products of the TLE987x family. See the datasheet for details.

Field	Bits	Type	Description
ADC3_INN_SEL	9	rw	**ADC3 Negative Input Selection**[1)] 0_B **none**, ADC3.N input disconnected 1_B **Mux-Port 2.2**, Port 2.2 multiplexer is connected to ADC3.N
ADC3_INP_SEL	8	rw	**ADC3 Positive Input Selection**[1)] 0_B **none**, ADC3.P input disconnected 1_B **Mux-Port 2.0**, Port 2.0 multiplexer is connected to ADC3.P
RES	7:6	r	**Reserved** Always read as 0x
RES	5	r	**Reserved** Always read as 0x
P2_5_ADC_SEL	4	rw	**Port 2.5 Input Selection** 0_B **ADC1.Ch5**, Port 2.5 is connected to ADC1.Ch5 1_B **ADC4.N**, Port 2.5 is connected to ADC4.N[1)]
P2_4_ADC_SEL	3	rw	**Port 2.4 Input Selection** 0_B **ADC1.Ch4**, Port 2.4 is connected to ADC1.Ch4 1_B **ADC4.P**, Port 2.4 is connected to ADC4.P[1)]
P2_3_ADC_SEL	2	rw	**Port 2.3 Input Selection** 0_B **ADC1.Ch3**, Port 2.3 is connected to ADC1.Ch3 1_B **none**, ADC1.Ch3 disconnected from P2.3
P2_2_ADC_SEL	1	rw	**Port 2.2 Input Selection** 0_B **ADC1.Ch2**, Port 2.2 is connected to ADC1.Ch2 1_B **ADC3.N**, Port 2.2 is connected to ADC3.N[1)], ADC3_INN_SEL must be set to ‘1’
P2_0_ADC_SEL	0	rw	**Port 2.0 Input Selection** 0_B **ADC1.Ch0**, Port 2.0 is connected to ADC1.Ch0 1_B **ADC3.P**, Port 2.0 is connected to ADC3.P[1)], ADC3_INP_SEL must be set to ‘1’

1) Available only for products with SDADC. See the datasheet for details.

22.5 VDH Input Voltage Attenuator

22.5.1 Functional Description

The function of the VDH voltage attenuator is to scale down the applied voltage on the VDH pin for the on-chip 10-Bit ADC. The voltage attenuator has the following features:

Features

- Two selectable attenuation factors: ATT_{VDH_1}, ATT_{VDH_2} (with ATT_{VDH_3} for a limited temperature range), please refer to the data sheet for the values
- Analog Mux switching time < 30 µs
- Input resistance: $R_{in_VDH,measure}$ (see data sheet for value)

Note: External series resistor affects measurement resolution accuracy by changing the internal attenuation factor.

The next chapter lists the user configuration possibilities of the VDH input voltage attenuators.

Note: **It is important to disable the VDH attenuator before entering Stop Mode. Otherwise an increased system current consumption can be observed.**

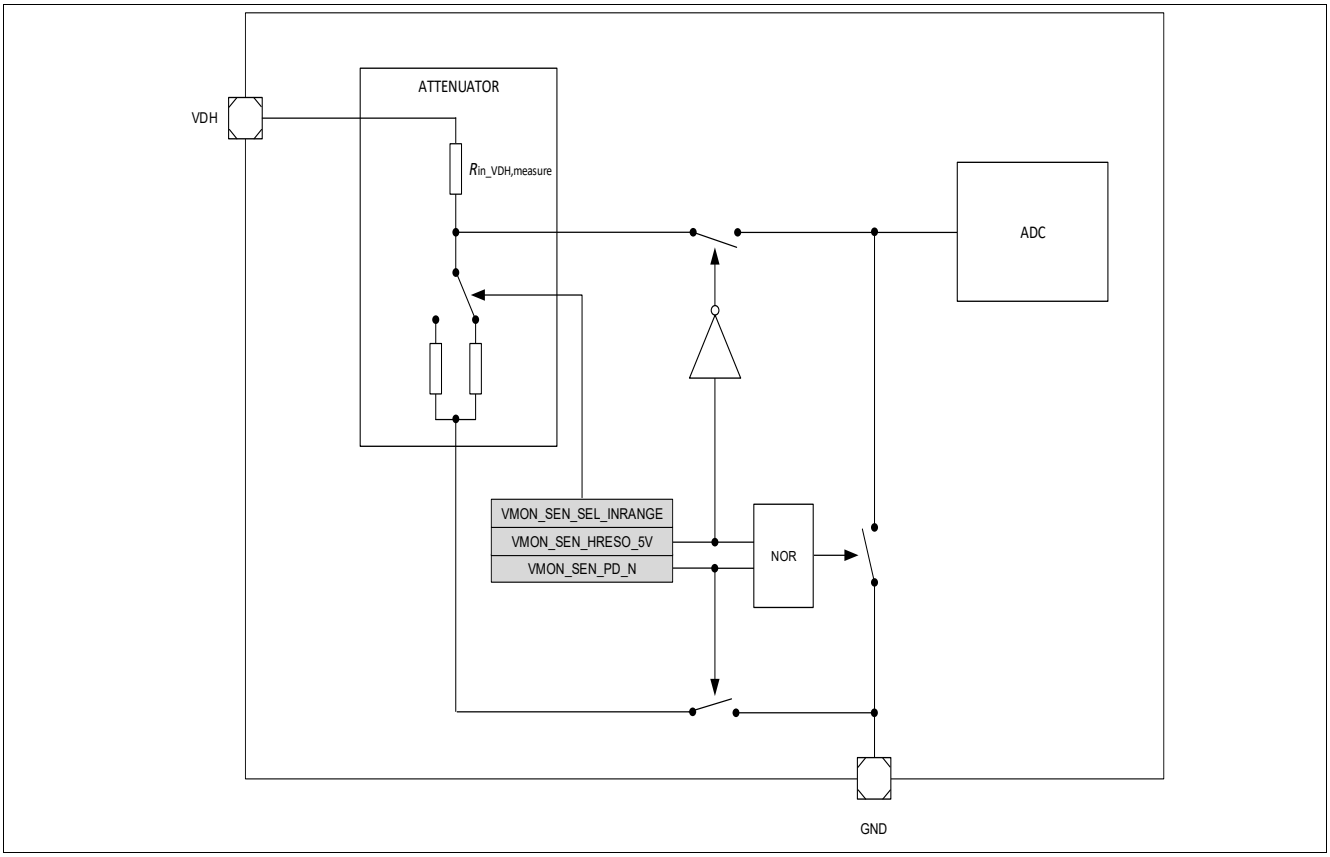


Figure 193 Input Structure of HV-VDH Attenuator

Table 160 VDH attenuator input structure truth table

VMON_SEN_PD_N	VMON_SEN_HRES0_5V	VMON_SEN_SEL_INRANGE	Function
0	0	0	ADC1.Ch6 input pulled down
0	0	1	ADC1.Ch6 input pulled down
0	1	0	Attenuator deactivated, ADC1.Ch6 input HiZ
0	1	1	Attenuator deactivated, ADC1.Ch6 input HiZ
1	0	0	measurement with attenuation for 0-20V range
1	0	1	measurement with attenuation for 0-30V range
1	1	0	reserved
1	1	1	reserved

22.5.2 VDH Input Attenuator Control Register

Table 161 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
VDH Input Attenuator Control Register			
VMON_SEN_CTRL	Monitoring Supply Sense Control Register	04_H	$0000\ 0000_H$

The registers are addressed bytewise.

VDH Monitoring Input Sense Control Register

The register is reset by RESET_TYPE_3.

VMON_SEN_CTRL
Monitoring Sense Control Register **(04_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES										VMON_SEN_SEL_INRANGE	VMON_SEN_HRESO_5V	RES			VMON_SEN_PD_N
r										rw	rw	r			rw

Field	Bits	Type	Description
RES	31:6	r	**Reserved** Always read as 0x
VMON_SEN_SEL_INRANGE	5	rw	**VDH Monitoring Input Attenuator Select Input Range** 0_B **0 - 20V Range**, is selected (ATT_{VDH_2}, ATT_{VDH_3}) 1_B **0 - 30V Range**, is selected (ATT_{VDH_1})
VMON_SEN_HRESO_5V	4	rw	**VDH Monitoring Input Attenuator High Impedance Output Control** 0_B **Attenuator Output Enable**, attenuator connected to ADC1.Ch6 1_B **Attenuator Output Disable**, attenuator disconnected from ADC1.Ch6
RES	3:1	r	**Reserved** Always read as 0
VMON_SEN_PD_N	0	rw	**VDH Monitoring Input Attenuator enable** 0_B **DISABLE**, Attenuator switched off 1_B **ENABLE**, Attenuator switched on

22.6 BEMF Comparators

22.6.1 Functional Description

For the detection of the position of a motor the BEMF information can be used. This BEMF information is always sensed in the phase which is currently not active. As an example, the figure below shows the BEMF comparator of phase U and typical phase voltage and BEMF comparator output signals.

Module Features

- Threshold derived from remaining active phases.
- Low settling time < 1 µs.
- Comparators can be switched off if not needed to avoid additional power consumption and undesired input currents in power down modes.

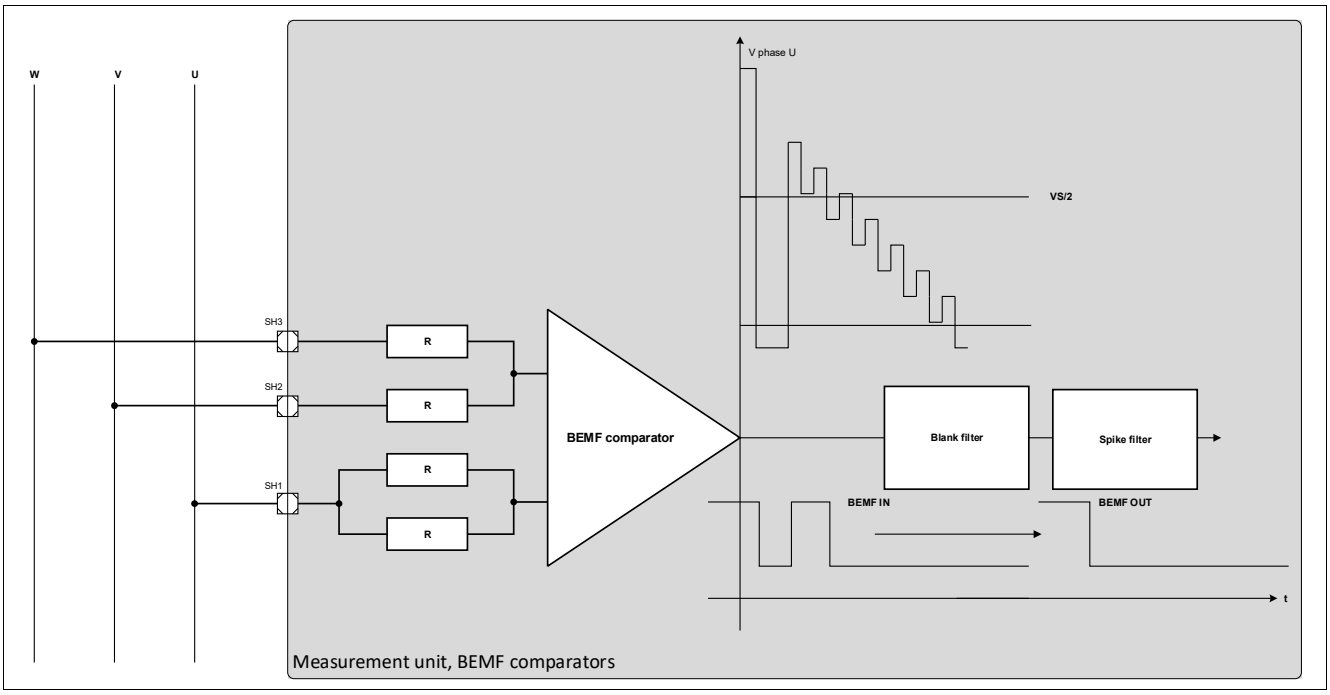


Figure 194 Connection of BEMF Comparator of Phase U

With SH1, SH2, and SH3 being the motor phase voltages, the input voltages of the BEMF comparator are:

$$V_{+} = V_{SH1} \tag{22.3}$$

$$V_{-} = \frac{V_{SH2} + V_{SH3}}{2}$$

The output V_{OUT} of the BEMF comparator is:

$$V_{OUT} = \begin{cases} low & if \quad V_{+} < V_{-} \\ high & if \quad V_{+} > V_{-} \end{cases} \tag{22.4}$$

For the BE, BF, and A design steps, the differential input voltage is limited to a maximum of 12 V.

22.6.1.1 Postprocessing of BEMF Comparator Signals

In order to provide a suitable signal for the system PWM engine (CCU6) to trigger commutation events the outputs of the on-chip comparators need to be postprocessed. This postprocessing is shown in the figure below:

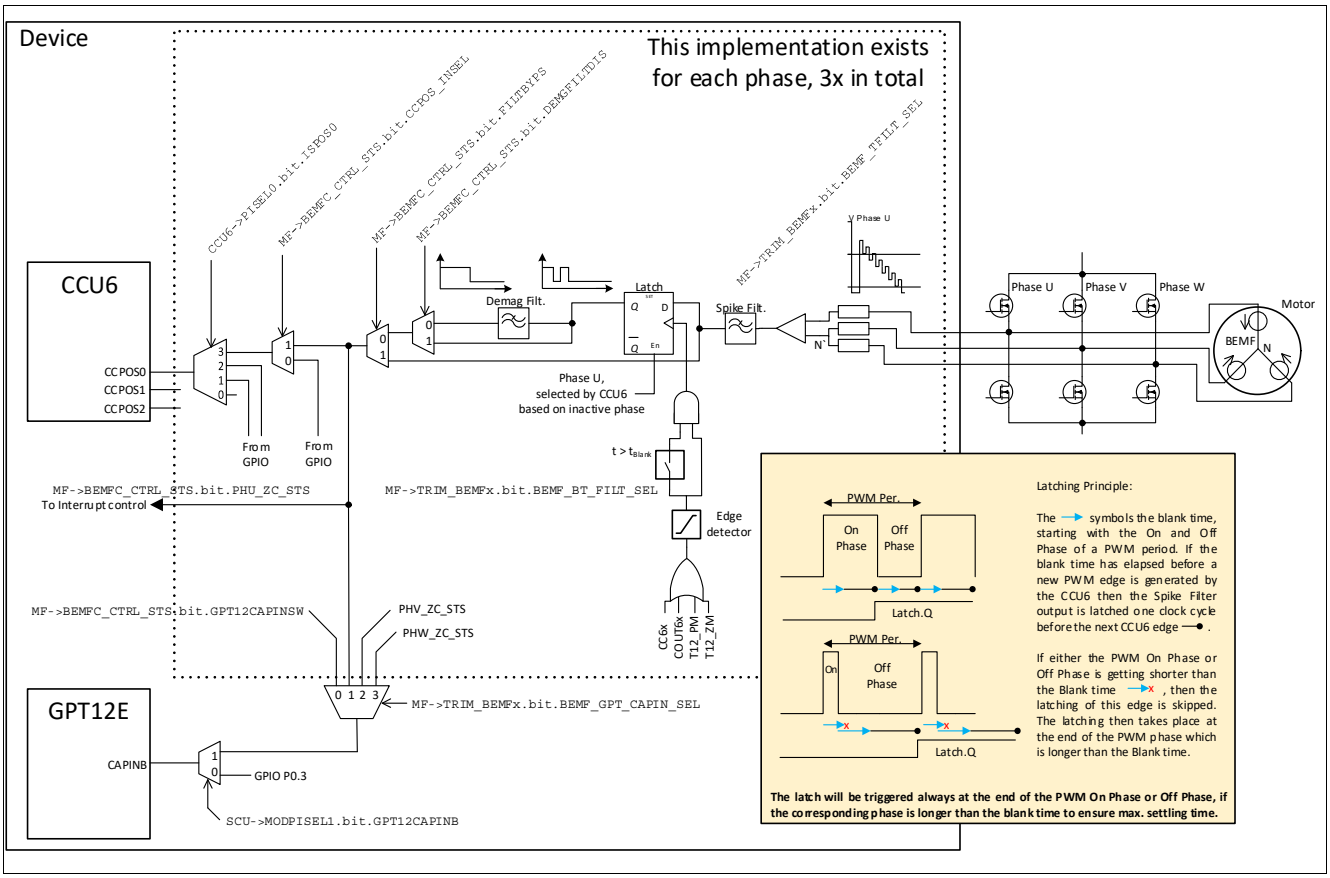


Figure 195 BEMF Comparator post-processing

The post-processing consists of a spike filter providing a low pass filtered signal at the input of the latch, the spike filter can be configured by **TRIM_BEMFx**.

The latch is triggered by the CCU6, either by the channel signals, or by the Timer12 zero match or period match for 0% and 100% duty cycles. As highlighted in the yellow box, the latch will be triggered one clock cycle before the end of the current PWM phase, On Phase, or Off Phase. The blank filter ensures that the latching only takes place if the corresponding PWM phase has a width greater than the blank time. This provides the maximum possible settling time for the BEMF signal.

The output of the latch can either be selected directly, or being fed through the demagnetization filter. The option to bypass the demag. filter provides more flexibility to the application to handle the start up of the motor, where due to improper phasing of the electric field inside the motor to its rotor no zero-crossing of the generated BEMF signal is yet present. The existence of the demag. pulse can then be used to toggle the CCPOSx signal.

If the motor is running in closed loop, the demag. filter shall be turned on, because then the undesired demag. pulse is removed and the CCPOSx signal toggles only at the zero-crossing of the BEMF.

Multiplexers at the output allow to feed out the BEMF signal either right after the spike filter, which is useful to check whether a motor is already, or still, rotating without commutation (windmilling), or before or after the demag. filter.

The status of the BEMF comparators can be routed to the CAPINB of the GPT12 module. As an example, if the GPT2.T5 is running freely in timer mode, a selected edge on the desired BEMF status captures the current timer value into the CAPREL register. If needed the GPT12.T5 can be reset by the capture event. The CAPREL register then holds the time (in GPT12.T5 clock ticks) between the selected edges of the BEMF status, which is an equivalent for the rotating speed of the motor.

As displayed in **Figure 195** the BEMF status signals can be routed to the CCPOSx inputs of the CCU6. Usually a HALL sensor is connected to the CCPOSx inputs. The three BEMF status signals together do count like a 3-times HALL pattern, therefore the CCU6 internal HALL pattern evaluation unit can be used for BEMF evaluation as well, see also **Chapter 18.7**.

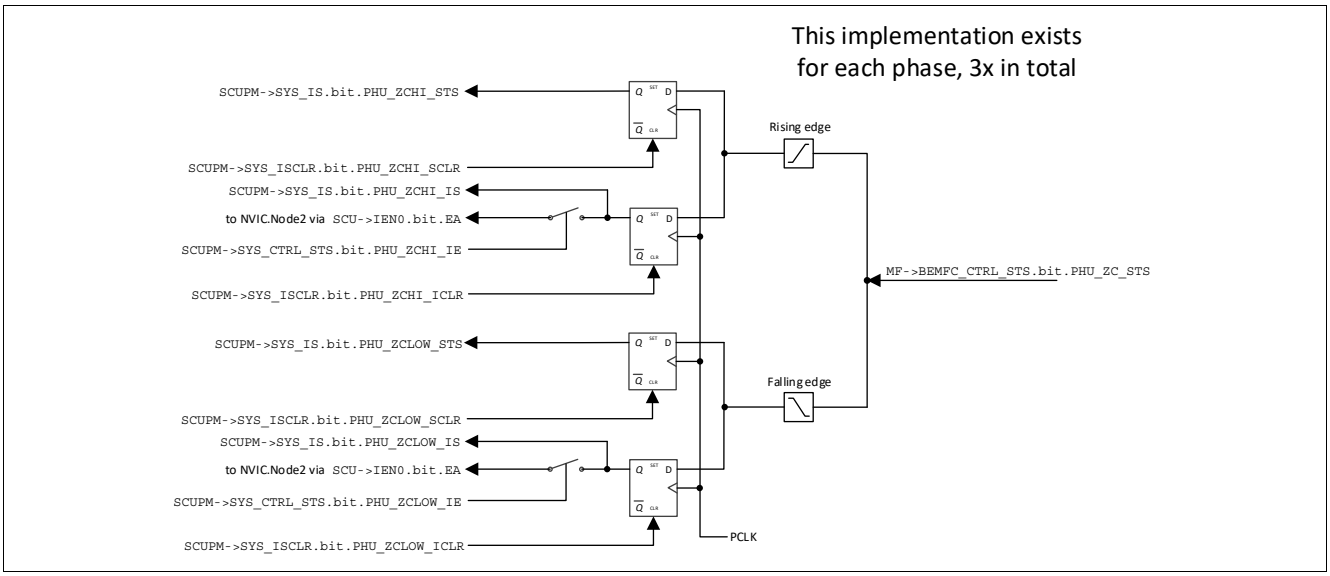


Figure 196 BEMF Comparator Interrupt Structure

Figure 196 lists the interrupt structure for the Phase U, the implementation for the other two phases is analog.

22.6.2 BEMF Comparators Control and Status Register

The next chapter lists the configuration and status readback possibilities of the BEMF comparators used for zero crossing detection.

Table 162 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
BEMF Comparators Control and Status Register			
BEMFC_CTRL_STS	BEMF Comparator Control Status Register	08_H	$0000\ 0000_H$
TRIM_BEMFx	Trimming of BEMF comparators	20_H	$0000\ 0000_H$

The registers are addressed wordwise.

BEMF Comparator Control and Status Register

The register is reset by RESET_TYPE_3.

BEMFC_CTRL_STS
BEMF Comparator Control Status Register **(08_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													PHW_ZC_STS	PHV_ZC_STS	PHU_ZC_STS
						r							r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			CCPOS_INSEL	RES	PHWCOMP_ON	PHVCOMP_ON	PHUCOMP_ON	RES		GPT12CAPINSW	FILTBYPS	DEMGFILTDIS	PHWCOMP_EN	PHVCOMP_EN	PHUCOMP_EN
	r		rw	r	rw	rw	rw	r		rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
PHW_ZC_STS	18	r	**Phase W Comparator zero crossing status** BEMF comparator outputs '1' when it is switched off 0_B **Below Threshold**, zero crossing comparator input is below threshold 1_B **Above Threshold**, zero crossing comparator input is above threshold
PHV_ZC_STS	17	r	**Phase V Comparator zero crossing status** BEMF comparator outputs '1' when it is switched off 0_B **Below Threshold**, zero crossing comparator input is below threshold 1_B **Above Threshold**, zero crossing comparator input is above threshold

Field	Bits	Type	Description
PHU_ZC_STS	16	r	**Phase U Comparator zero crossing status** BEMF comparator outputs '1' when it is switched off 0_B **Below Threshold**, zero crossing comparator input is below threshold 1_B **Above Threshold**, zero crossing comparator input is above threshold
RES	15:13	r	**Reserved** Always read as 0
CCPOS_INSEL	12	rw	**CCPOSx_3 ISPOS select, x = 0,1,2** 0_B **ENABLE**, GPIO selected 1_B **ENABLE**, BEMF selected, set CCU6.**PISEL0**.ISPOSx = 3
RES	11	r	**Reserved** Always read as 0
PHWCOMP_ON	10	rw	**Phase W Comparator on** *Note: for a proper operation of BEMF functionality all three comparators need to be switched on* 0_B **DISABLE**, Comparator is switched off 1_B **ENABLE**, Comparator is switched on
PHVCOMP_ON	9	rw	**Phase V Comparator on** *Note: for a proper operation of BEMF functionality all three comparators need to be switched on* 0_B **DISABLE**, Comparator is switched off 1_B **ENABLE**, Comparator is switched on
PHUCOMP_ON	8	rw	**Phase U Comparator on** *Note: for a proper operation of BEMF functionality all three comparators need to be switched on* 0_B **DISABLE**, Comparator is switched off 1_B **ENABLE**, Comparator is switched on
RES	7:6	r	**Reserved** Always read as 0
GPT12CAPINSW	5	rw	**GPT12 CAPIN software trigger** set **TRIM_BEMFx**.BEMF_GPT_CAPIN_SEL = 0 0_B **Value 0**, software trigger reset 1_B **Value 1**, software trigger set
FILTBYPS	4	rw	**BEMF Comparator Output Filter Bypass** 0_B **ENABLE**, BEMF Comparator Output Filter enabled 1_B **DISABLE**, BEMF Comparator Output Filter disabled

Field	Bits	Type	Description
DEMGFILTDIS	3	rw	**BEMF Comparator Demagnetization (Demag) Filter Disable** 0_B **ENABLE**, BEMF Comparator Demag Filter enabled 1_B **DISABLE**, BEMF Comparator Demag FIlter bypass
PHWCOMP_EN	2	rw	**Phase W Comparator enable** *Note: for a proper operation of BEMF functionality all three comparators need to be enabled* 0_B **DISABLE**, Comparator is disabled 1_B **ENABLE**, Comparator is enabled
PHVCOMP_EN	1	rw	**Phase V Comparator enable** *Note: for a proper operation of BEMF functionality all three comparators need to be enabled* 0_B **DISABLE**, Comparator is disabled 1_B **ENABLE**, Comparator is enabled
PHUCOMP_EN	0	rw	**Phase U Comparator enable** *Note: for a proper operation of BEMF functionality all three comparators need to be enabled* 0_B **DISABLE**, Comparator is disabled 1_B **ENABLE**, Comparator is enabled

Trimming of BEMF comparators

The register is reset by RESET_TYPE_4.

TRIM_BEMFx
Trimming of BEMF comparators **(20_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						BEMF_TFILT_SEL		RES		BEMF_GPT_CAPIN_SEL		RES	BEMF_BT_TFILT_SEL		
r						rw		rw		rw		rw	rw		

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
BEMF_TFILT_SEL	9:8	rw	**Filter Time for BEMF Comparator Output Signal** 00_B **1_ µs**, 1 µs filter time 01_B **2_ µs**, 2 µs filter time 10_B **4_ µs**, 4 µs filter time 11_B **8_ µs**, 8 µs filter time
RES	7:6	rw	**Reserved**
BEMF_GPT_CAPIN_SEL	5:4	rw	**GPT12 CAPIN input selector** set **MODPISEL1**.GPT12CAPINB = 1 00_B **SWTRIG**, trigger from **BEMFC_CTRL_STS**.GPT12CAPINSW 01_B **Phase U**, trigger from **BEMFC_CTRL_STS**.PHU_ZC_STS 10_B **Phase V**, trigger from **BEMFC_CTRL_STS**.PHV_ZC_STS 11_B **Phase W**, trigger from **BEMFC_CTRL_STS**.PHW_ZC_STS
RES	3:3	rw	**Reserved**
BEMF_BT_TFILT_SEL	2:0	rw	**Blanking Time for BEMF Comparator Output Signal** 000_B **6_ µs**, 6 µs filter time 001_B **8_ µs**, 8 µs filter time 010_B **12_ µs**, 12 µs filter time 011_B **16_ µs**, 16 µs filter time $1xx_B$ **3_ µs**, 3 µs filter time

22.7 Central and PMU Regulator Temperature Sensor

22.7.1 Functional Description

This module is a quasi-combination of a main on-chip temperature sensor and a PMU Regulator temperature sensor.

Modules Features

- Temperature range -40 …175°C
- Temperature range corresponds to single ended output voltage range 0.6 …1.2V, resolution approx. 1.8°C
- The combined system temperature sensor plus ADC can be calibrated in software using calibration figures that are stored in the NVM at the production test.
- A dedicated calibration transistor facilitates on-chip reference temperature measurement. Hardware trimming is not implemented.

This temperature sensor, including two sensing elements, monitors the chip temperature (T_j) and PMU Regulator temperature. One sensing element is placed in the center of the device to get the average device temperature status and the other sensing element is close to the PMU Regulator.

The ADC output value is given by:

$$\mathrm{ADC2out} = \mathrm{floor}\left(\frac{V_{Temp}}{V_{LSB}} + 1\right) \tag{22.5}$$

The LSB Voltage is calculated:

$$V_{LSB} = \frac{V_{BG}}{2^8 - 1} = \frac{1.211V}{255} = 4.749mV \tag{22.6}$$

Vtemp is the direct proportional to temperature input voltage and is calculated by:

$$V_{\mathrm{TEMP}}(T) = \mathrm{a} + \mathrm{b} \bullet (T - T_0) \tag{22.7}$$

where **the coefficient a** is **666 mV, b is 2,31 mV/K and T_0 is 273 K, T has to be given in Kelvin**.

The next chapter lists the available registers to configure temperature sensors.

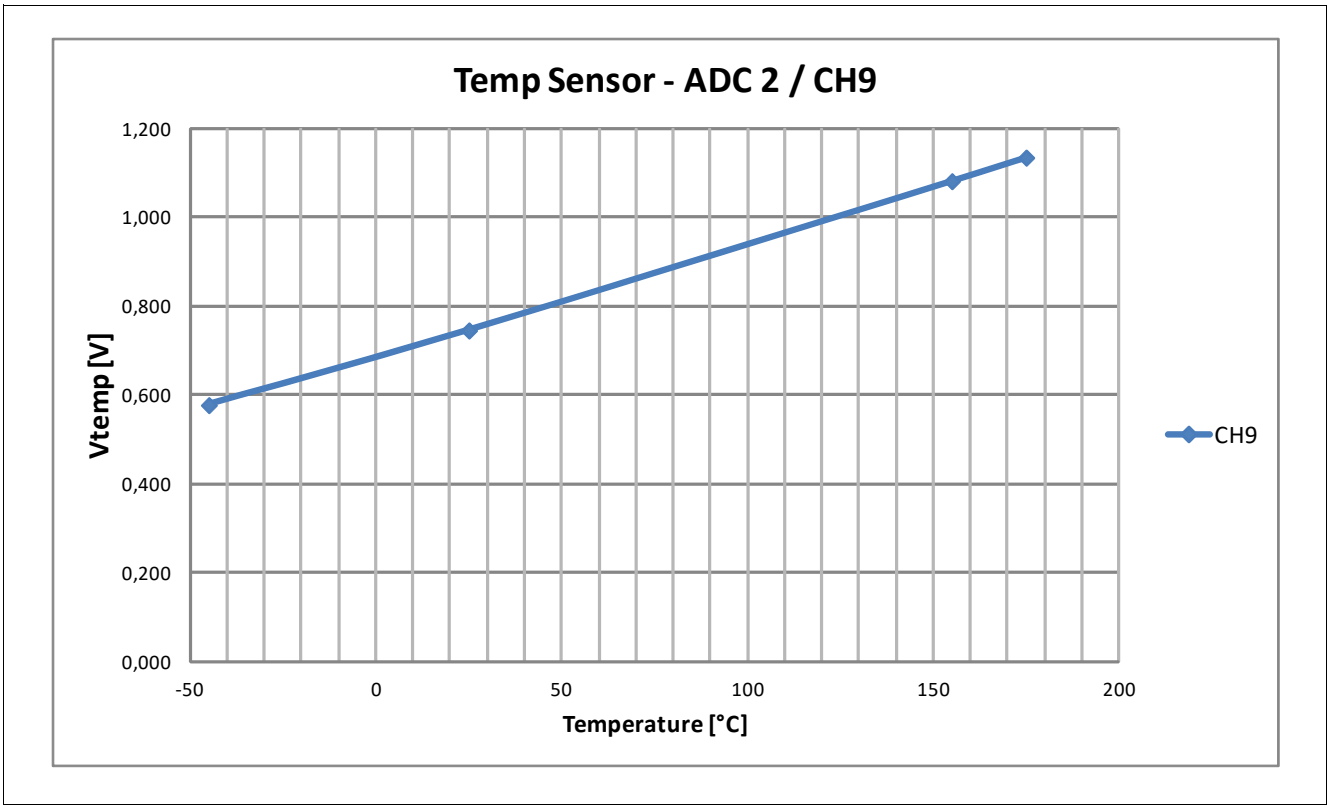


Figure 197 Measured Characteristic of On Chip Temperature Sensor

22.7.2 Temperature Sensor Control Register

The Temperature Sensor is fully controllable by the below listed SFR Register.

Table 163 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Temperature Sensor Control Register			
TEMPSENSE_CTRL	Temperature Sensor Control Register	10_H	$0000\ 0003_H$

The registers are addressed bytewise.

Temperature Sensor Control Register

The register is reset by RESET_TYPE_3.

TEMPSENSE_CTRL
Temperature Sensor Control Register **(10_H)** **Reset Value: $0000\ 0003_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								SYS_OT_STS	SYS_OTWARN_STS	PMU_OT_STS	PMU_OTWARN_STS	RES			
r								r	r	r	r	r			

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
SYS_OT_STS	7	r	**System Overtemperature (ADC2-Channel 9) Status** 0_B **INACTIVE**, write clears status 1_B **ACTIVE**, interrupt status set
SYS_OTWARN_STS	6	r	**System Overtemperature Warning (ADC2-Channel 9) Status** 0_B **INACTIVE**, write clears status 1_B **ACTIVE**, interrupt status set
PMU_OT_STS	5	r	**PMU Regulator Overtemperature (ADC2-Channel 9) Status** 0_B **INACTIVE**, write clears status 1_B **ACTIVE**, interrupt status set
PMU_OTWARN_STS	4	r	**PMU Regulator Overtemperature Warning (ADC2-Channel 9) Status** 0_B **INACTIVE**, write clears status 1_B **ACTIVE**, interrupt status set
RES	3:0	r	**Reserved** Always read as 0

22.8 Supplement Modules

22.8.1 Functional Description

The purpose of the supplement modules is to enable a certain infrastructure on the device to guarantee a fail safe operation:

Module Features

- Bandgap Reference Voltage with accuracy ± 1.5% over complete temperature range.
- Bandgap is monitored by an independent reference voltage.
- ADC1 Reference with accuracy ± 1%.
- ADC1 Reference has overload detection.

22.8.1.1 ADC2 Functional Safety Feature

ADC2 offers a functional safety feature, which allows to monitor the consistency of ADC2 results. The channel 7 of ADC2 monitors a voltage which is generated by a resistor of 26.66k and a 40 µA current. This 40 µA current is generated by a similar second resistor and a voltage, which is a second independent reference voltage in the system.

The root cause for a code change of channel 7 of ADC2 can only be explained by a ADC2 reference voltage variation. This feature can be used to validate the reference voltage of ADC2 and the results of the conversion.

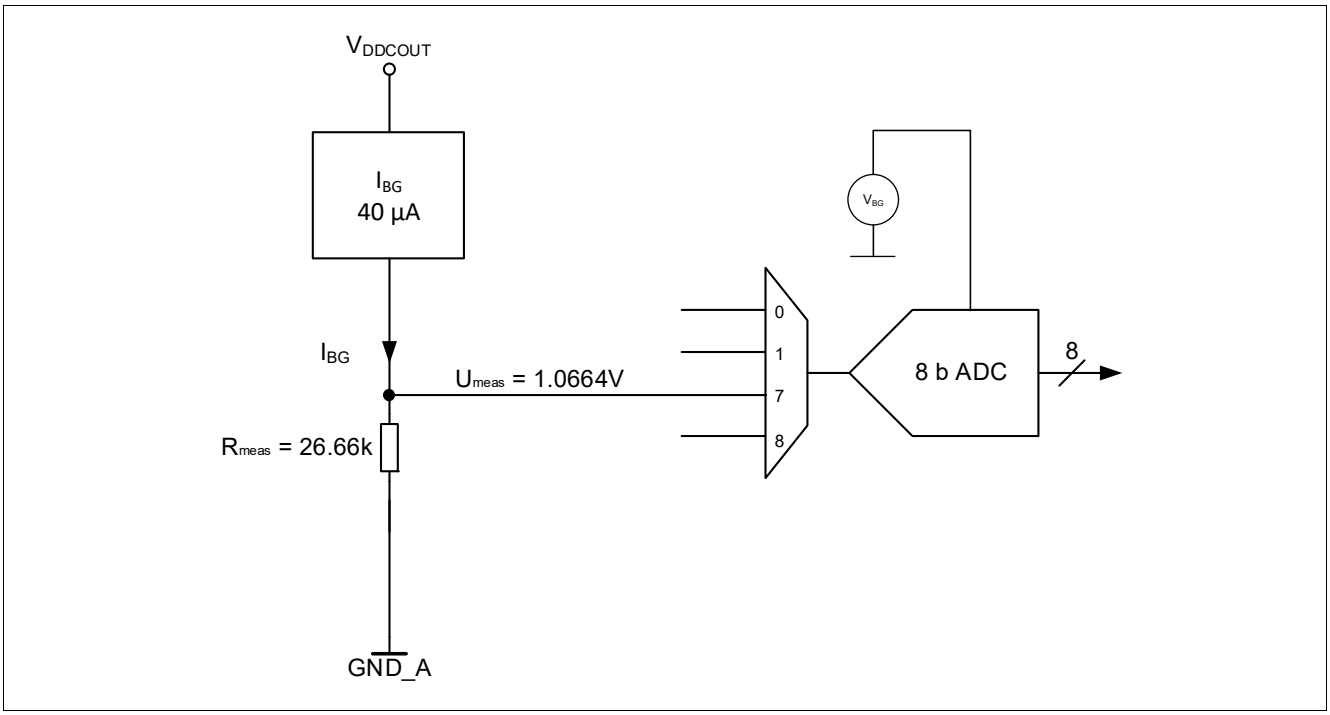


Figure 198 Principle of ADC2 functional safety feature

22.8.1.2 ADC1 reference voltage (V_{AREF})

An internal reference voltage circuit generates the 5-V output (V_{AREF}) which is used as reference for the 10-bit ADC. The ADC1 reference voltage (V_{AREF}) can either be provided from the internal source (V_{AREF}) or it can also be supplied externally over the VAREF pin. If this reference is provided externally, set the **REF2_CTRL.VREF5V_PD_N** bit to 0 to disable the internal voltage reference circuit.

22.8.2 Supplement Modules Control and Status Register

The next chapter lists the diagnosis and configuration possibilities of the supplement modules.

Table 164 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Supplement Modules Control and Status Register			
REF1_STS	Reference 1 Status Register	14_H	$0000\ 00C1_H$
REF2_CTRL	Reference 2 Control Register	18_H	$0000\ 0001_H$

The registers are addressed bytewise.

Reference 1 Status Register

The register is reset by RESET_TYPE_3.

This register is being initialized by the BootROM during the boot-up process to a value of $0000\ 08C1_H$

REF1_STS

Reference 1 Status Register **(14_H)** **Reset Value: $0000\ 00C1_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES										REFBG_UPTHWARN_ST	REFBG_LOTHWARN_STS	RES	RES		
r										r	r	r	r		

Field	Bits	Type	Description
RES	31:6	r	**Reserved** Always read as 0000003_H
REFBG_UPTHWARN_STS	5	r	**Status for Overvoltage Threshold Measurement of internal bandgap reference VBG** 0_B **UPPER_TRIG_RESET**, write clears status 1_B **UPPER_TRIG_SET**, trigger status set
REFBG_LOTHWARN_STS	4	r	**Status for Undervoltage Threshold Measurement of internal bandgap reference VBG** 0_B **UPPER_TRIG_RESET**, write clears status 1_B **UPPER_TRIG_SET**, trigger status set
RES	3	r	**Reserved** Always read as 0
RES	2:0	r	**Reserved** Always read as 001_B

Reference 2 Control Register

The register is reset by RESET_TYPE_3.

REF2_CTRL
Reference 2 Control Register **18_H** **Reset Value: $0000\ 0001_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES												VREF5V_OV_STS	VREF5V_UV_STS	VREF5V_OVL_STS	VREF5V_PD_N
r												r	r	r	rw

Field	Bits	Type	Description
RES	31:4	r	**Reserved** Always read as 1
VREF5V_OV_STS	3	r	**ADC1 Bit Reference Voltage Generation Overvoltage Bit** 0_B **no Overvoltage**, no Overvoltage detected 1_B **Overvoltage**, Overvoltage detected
VREF5V_UV_STS	2	r	**ADC1 Bit Reference Voltage Generation Undervoltage Bit** 0_B **no Undervoltage**, no Undervoltage detected 1_B **Undervoltage**, Undervoltage detected
VREF5V_OVL_STS	1	r	**ADC1 Bit Reference Voltage Generation Over Load Bit** 0_B **no OVERLOAD**, no OVERLOAD detected 1_B **OVERLOAD**, OVERLOAD detected
VREF5V_PD_N	0	rw	**ADC1 Bit Reference Voltage Generation Power Down Bit** 0_B **DISABLED**, Power Down 1_B **ACTIVE**, no Power Down

Re-enabling of VAREF

In order to re-enable VAREF, the VAREF enable flag hast to be cleared, MF->**REF2_CTRL**.VREF5V_PD_N = 0. Then the status flags should be cleared, VREF5V_LOWTH_ICLR, VREF5V_UPTH_ICLR and VREF5V_OVL_ICLR in the register SCUPM->**SYS_ISCLR**. After that re-enable VAREF by setting MF->**REF2_CTRL**.VREF5V_PD_N = 1.

23 Measurement core module (incl. ADC2)

23.1 Features

- 10 individually programmable channels, split into two groups of user-configurable and non-configurable channels, respectively.
- Individually programmable channel prioritization scheme for the measurement unit.
- Two independent filter stages with programmable low-pass and time filter characteristics for each channel.
- Two channel configurations:
  - Programmable upper- and lower trigger thresholds comprising a fully programmable hysteresis.
  - Two individually programmable trigger thresholds with limit hysteresis settings.
- Individually programmable interrupts and statuses for all channel thresholds.

23.2 Introduction

The basic function of this block is the digital postprocessing of several digitized analog measurement signals by filtering, level comparison, and interrupt generation. The measurement postprocessing block consists of ten identical channel units attached to the outputs of the 10-channel 8-bit ADC (ADC2). It processes ten channels, where the channel sequence and prioritization is programmable within a wide range.

23.2.1 Block diagram

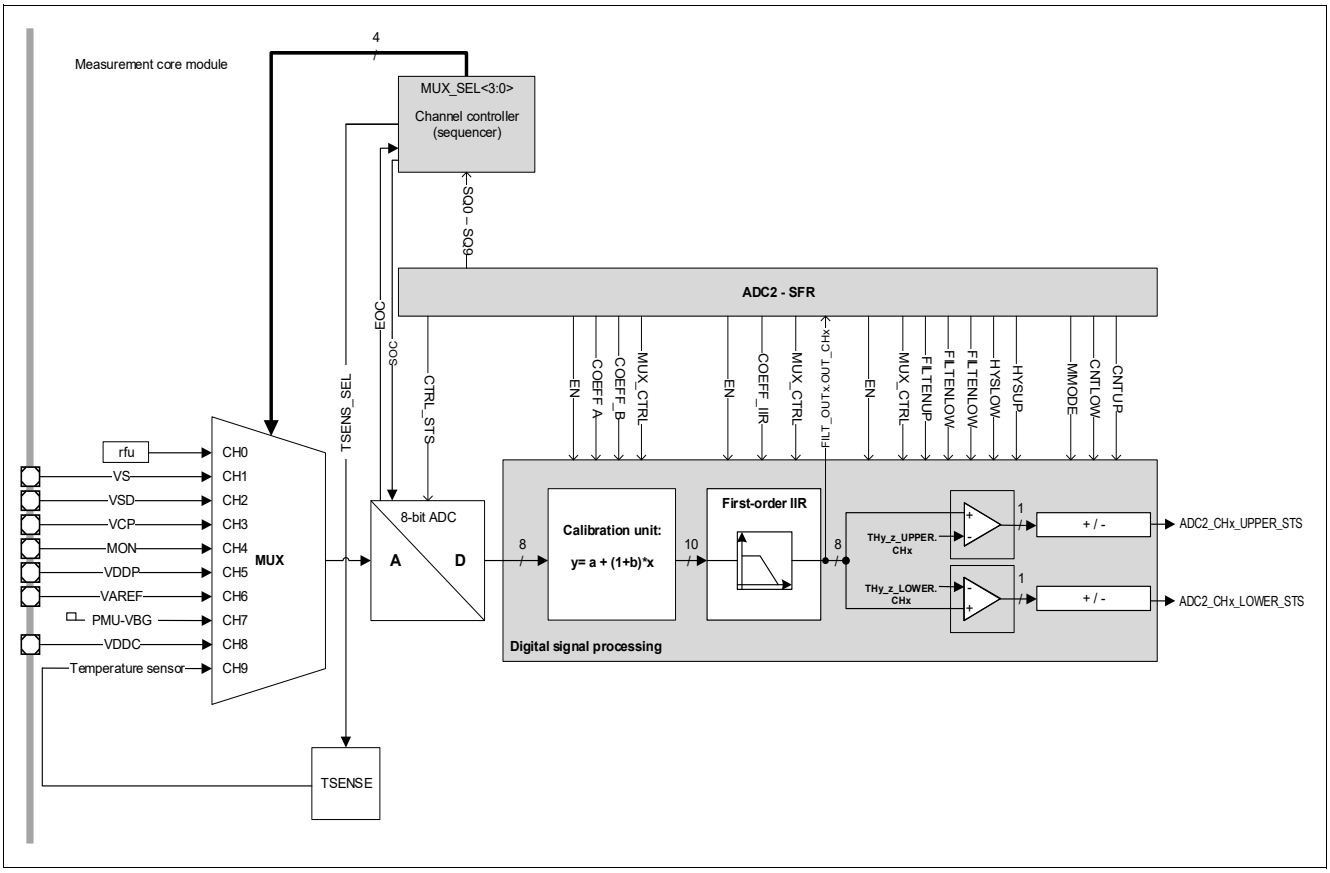


Figure 199 Module block diagram

23.2.2 Measurement core module mode overview

The basic function of this unit is the digital signal processing of several digitized analog measurement signals by filtering, level comparison, and interrupt generation. The measurement core module processes ten channels in a quasi-parallel process.

As shown in the figure above, the ADC2 postprocessing unit consists of a channel controller (sequencer), a 10-channel demultiplexer, and the signal-processing block, which filters and compares the sampled ADC2 values for each channel individually. The channel control block controls the multiplexer sequencing on the analog side, before the ADC2, and in the digital domain, behind the ADC2. The channel sequence can be controlled in a flexible way, which allows a certain degree of channel prioritization.

This capability can be used, e.g., to give supply voltage channels a higher priority than the other channel measurements. In addition, the measurement core module offers two different postprocessing measurement modes for over- and undervoltage detection and for two-level threshold detection.

Usually the external register settings should only be changed during the start-up phase of the post-processing module. Otherwise the signal processing might be disturbed.

The channel controller (sequencer) runs in one of the following modes:

- Normal Sequencer: Channels are selected according to the 10 sequence registers which contain individual enablers for each of the 10 channels.
- Exceptional Interrupt Measurement: Following a hardware event, a high-priority channel is inserted into the current sequence. The current actual measurement is not destroyed.
- Exceptional Sequence Measurement: Following a hardware event, a complete sequence is inserted after the current measurement is finished. The current sequence is interrupted by the exception sequence.

The threshold counter can be bypassed **FILT_UP_CTRL** and **FILT_LO_CTRL**.

23.3 ADC2 - Core (8-bit ADC)

23.3.1 Functional Description

The different sequencer modes are controlled by SFR Register:

- “Normal Sequencer Mode” described in the Chapter **Channel Controller**.
- “Exceptional Interrupt Measurement” (EIM), upon hardware event , the channel programmed in **CHx_EIM** is inserted after the current measurement is finished. Afterwards the current sequence will be continued with the next measurement from the current sequence.
- “Exceptional Sequence Measurement“ (ESM), upon hardware event , the sequence programmed in **CHx_ESM** is inserted after the current measurement is finished. After the sequence (up to 10 measurements) exception is finished the next measurement from the interrupted sequence is selected. After the Exceptional Sequence Measurement is finished an interrupt is issued.

Debug Suspend Mode:

During Debug Suspend Mode the Sequencer is stopped once the current measurement is finished (after the next EOC event). As long as the Debug Suspend Mode is active no measurements are performed by the Sequencer. Once the Debug Suspend Mode is left, the Sequencer continues immediately with the next pending measurement.
Measurements can be still triggered in Debug Suspend Mode. EIM and ESM events are ignored during Debug Suspend Mode.

The ADC2 timing is controlled by SFR Register (Special Function Register)

- Sample time adjustment described in the register **CTRL2**.

23.3.2 ADC2 Control Registers

The ADC2 is fully controllable by the below listed SFR Registers. The control must be enabled by setting all sequencer bits to zero. To enable the sequencer again this corresponding bits in the sequencer register must be set to one again.

Table 165 shows the module base addresses.

Table 165 Register Address Space

Module	Base Address	End Address	Note
ADC2	$4801C000_H$	$4801DFFF_H$	ADC2 - ADC-SAR8B

Table 166 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
ADC2 Control Registers			
CTRL_STS	ADC2 Control and Status Register	00_H	$0000\ 0001_H$
HV_STS	ADC2 HV Status Register	BC_H	$0000\ 0000_H$

The registers are addressed wordwise.

ADC2 Control and Status Register

The register is reset by RESET_TYPE_3.

CTRL_STS
ADC2 Control and Status Register **(00_H)** **Reset Value: $0000\ 0001_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													RES	VS_RANGE	RES
						r							r	rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			RES					RES				RES		RES	RES
r			r					r				r		r	r

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
RES	18	r	**Reserved** Always read as 0
VS_RANGE	17	rw	**ADC2 Channel 1 Range Selection** *Note: changing this bit during ADC conversion might change the conversion results to a wrong value* 0_B **Range 1**, Range from 3 to 22V is selected (ATT_{VS_1}) 1_B **Range 2**, Range from 3 to 31V is selected (ATT_{VS_2})
RES	16	r	**Reserved** Always read as 0
RES	15:13	r	**Reserved** Always read as 0
RES	12:8	r	**Reserved** Always read as 0
RES	7:4	r	**Reserved** Always read as 0
RES	3:2	r	**Reserved** Always read as 0
RES	1	r	**Reserved** Always read as 0
RES	0	r	**Reserved** Always read as 1

ADC2 HV Status Register

The register is reset by RESET_TYPE_3.

HV_STS
ADC2 HV Status Register **(BC_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES														READY	RES
r														r	r

Field	Bits	Type	Description
RES	31:2	r	**Reserved** Always read as 0
READY	1	r	**HV ADC Ready bit** 0_B **Not ready**, Module in power down or in init phase 1_B **Ready**, set automatically 5 ADC clock cycles after module is enabled
RES	0	r	**Reserved** Always read as 0

23.4 Channel Controller

23.4.1 Functional Description

The task of channel controller is a prioritization of the individual measurement channels. The sequencing scheme is illustrated in the example of following table and can be programmed.

Table 167 Measurement channel sequence definition example (used as default sequence)

Measurement channel n $SQ'_z = \{SQ_{x_y}_int[3:0], SQ_{z}[5:0]\}$	MSB CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	LSB CH0
Registers $SQ'_1[9:0]$	1	0	1	1	1	1	0	1	1	1
Registers $SQ'_2[9:0]$	0	1	1	1	1	0	1	0	0	0
Registers $SQ'_3[9:0]$	1	0	1	1	1	1	0	1	1	0
Registers $SQ'_4[9:0]$	0	1	1	1	1	0	1	0	0	1
Registers $SQ'_5[9:0]$	1	0	1	1	1	1	0	1	1	0
Registers $SQ'_6[9:0]$	0	1	1	1	1	0	1	0	0	0
Registers $SQ'_7[9:0]$	1	0	1	1	1	1	0	1	1	1
Registers $SQ'_8[9:0]$	0	1	1	1	1	0	1	0	0	0
Registers $SQ'_9[9:0]$	1	0	1	1	1	1	0	1	1	0
Registers $SQ'_{10}[9:0]$	0	1	1	1	1	0	1	0	0	1

The sequence registers SQ_n and SQ_n_int define the time sequence of the measurement channels by the following rules:

- The sequence registers define the measurement sequence and are evaluated from register 1 to 10 and for each register from MSB to LSB, which defines a max. overall measurement periodicity of 100 sampling and conversion cycles.
- If the individual bit in the sequence register is set to ’1’, the corresponding channel is measured.
- If the individual bit in the sequence register is set to ’0’, the corresponding channel is skipped.

In the upper example, the resulting channel sequence is defined as:

CH9, CH7, CH6, CH5, CH4, CH2, CH1, CH0, CH8, CH7, CH6, CH5, CH3,....., CH8, CH7, CH6, CH5, CH3, CH0

In TLE987x Channels 0 - 5 can be fully programmed. Several Sequence registers, especially for channels 6-9, are protected to ensure a fast update of measurement results used for internal diagnosis. Hence the channels 6 and 7 are prioritized and are measured more often, the overall periodicity is mainly determined by these two channels. The channels 0-5 are measured depending on the amount of ’1’ bits, written in the sequence registers. The following equations can be used to calculate the periodicity of the required channel measurement.

Measurement core module (incl. ADC2)

The total number of measurement done by the sequencer in a complete run-through expressed in A/D conversion cycles is defined as:

(23.1)

$$N_{meas} = \sum_{m=1}^{10} \left(\sum_{n=1}^{10} SQ`_m[n] \right)$$

The average measurement periodicity of channel n in A/D conversion cycles is defined as:

(23.2)

$$\overline{T_{meas,\,n}} = \frac{\left(\sum_{m=1}^{10} SQ`_m[n] \right)}{T_{meas}}$$

Where T_{meas} is defined as:

(23.3)

$$T_{meas} = N_{meas} * (T_{samp,\,n} * T_{conv})$$

N_{meas} is the number of measurements programmed in the SQ' registers, $T_{samp,n}$ is the sampling time setting for channel n and T_{conv} is the ADC2 conversion time.

The timing of the analog MUX and the digital DEMUX is controlled by the channel controller accordingly. The analog MUX with sample and hold stage needs one clock cycle for channel switching and the ADC takes, as default setting, 18 clock cycles for the sampling of the input voltage. The conversion time for a single channel measurement value is 15 clock cycles.

As already mentioned above, the channel controller has a partly fixed sequence register setting which cannot be changed by the user. The fixed register setting is needed, to fulfill the sampling frequency requirements of the internal circuits, e.g. shutdown in case of overtemperature and protection for the bridge drivers (BDRV).

The minimum measurement periodicity, which can be achieved, by enabling only channel1 in the sequence registers, depends on the MI_CLK frequency and is given by (example):

(23.4)

$$\overline{t_{meas_ch1_min}} = \frac{ticks_{sampling} + ticks_{conv}}{f_{MI_CLK}} = \frac{18 + 15}{20MHz} = 1.65\mu s$$

The following calculation includes the sampling time as well as the conversion time of ADC2 and the switching time of the channel multiplexer. If all programmable channels are enabled, the maximum periodicity of channel 1 is calculated by (example): (23.5)

$$\overline{t_{meas_ch1_max}} = \frac{n_{channels} \cdot (ticks_{sampling} + ticks_{conv} + ticks_{mux_switch})}{f_{MI_CLK}} = \frac{10 \cdot (18 + 15 + 1)}{20MHz} = 17\mu s$$

At a MI_CLK frequency of 20 MHz, the channel 1 is measured with min. 1.65 µs. The maximum update time of channel1 with 20 MHz clock frequency is 17 µs. This is calculated based on the assumption, that all channels are enabled and channel1 is enabled in every sequence register. As a prerequisite for this calculation **CTRL2**.SAMPLE_TIME_int = 7 (sample period = 18 MI_CLK clock cycles) is set.

23.4.2 Channel Controller Control Registers

The Channel Controller can be configured by the registers listed in **Table 168**. The registers which cannot be written by the user have the attribute **rwp**. Those registers are:

SQ1_8_int,

SQ9_10_int

Table 168 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Channel Controller Control Registers			
SQ_FB	Sequencer Feedback Register	04_H	0000 0000 000X XXXX 0XXX XXXX 0000 XXXX_B
CHx_EIM	Channel Setting Bits for Exceptional Interrupt Measurement	08_H	0000 0000_H
CHx_ESM	Channel Setting Bits for Exceptional Sequence Measurement	$0C_H$	0000 0000_H
CTRL1	Measurement Unit Control Register 1	14_H	0000 0000_H
CTRL2	Measurement Unit Control Register 2	18_H	000F 0805_H
CTRL4	Measurement Unit Control Register 4	$1C_H$	0000 $0F3F_H$
SQ1_4	Measurement Channel Enable Bits for Cycle 1-4	20_H	2936 2837_H
SQ5_8	Measurement Channel Enable Bits for Cycle 5-8	24_H	2837 2836_H
SQ9_10	Measurement Channel Enable Bits for Cycle 9-10	28_H	0000 2936_H
SQ1_8_int	Measurement Channel Enable Bits for Cycle 1-8	$2C_H$	7B7B $7B1B_H$
SQ9_10_int	Measurement Channel Enable Bits for Cycle 9 and 10	30_H	0000 $007B_H$

The registers are addressed wordwise.

Measurement Unit Control Register 1

This register is dedicated for controlling the calibration unit of the measurement core module. The respective channel calibration can be enabled or disabled by the bits listed below.

The register is reset by RESET_TYPE_4.

CTRL1

Measurement Unit Control Register 1 **(14_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				RES				RES		CALIB_EN					
r				r				r		rw					

Field	Bits	Type	Description
RES	31:12	r	**Reserved** Always read as 0
RES	11:8	r	**Reserved**
RES	7:6	r	**Reserved** Always read as 0
CALIB_EN	5:0	rw	**Calibration Enable for Channels 0 to 5** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 calibration enable $00\ 0010_B$ **CH1_EN**, Channel 1 calibration enable $00\ 0100_B$ **CH2_EN**, Channel 2 calibration enable $00\ 1000_B$ **CH3_EN**, Channel 3 calibration enable $01\ 0000_B$ **CH4_EN**, Channel 4 calibration enable $10\ 0000_B$ **CH5_EN**, Channel 5 calibration enable

Measurement Unit Control Register 2

This register contains the sample time adjustment for ADC2. The default value is 18 clock cycles. Values above 18 clock cycles are not recommended, because they increase the overall response time of the measurement system.

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of 0000 0703_H.

CTRL2

Measurement Unit Control Register 2 **(18_H)** **Reset Value: $000F\ 0805_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES												SEL_TS_COUNT			
r												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SAMPLE_TIME_int				MCM_RDY	RES				TSENSE_SD_SEL	TS_SD_SEL_CONF	MCM_PD_N
r				rw				r	r				rw	rw	rw

Field	Bits	Type	Description
RES	31:20	r	**Reserved** Always read as 0
SEL_TS_COUNT	19:16	rw	**Time for automatic multiplexing of temperature sensor** Defines how often all sequences SQ1-SQ10 shall be measured until the temperature sensor TSENSE_SD_SEL changes automatically (in case TS_SD_SEL_CONF is '1') 0_H **1**, Sequence 1_H **2**, Sequences 2_H **3**, Sequences 3_H **4**, Sequences 4_H **5**, Sequences 5_H **6**, Sequences 6_H **7**, Sequences 7_H **8**, Sequences 8_H **9**, Sequences 9_H **10**, Sequences A_H **11**, Sequences B_H **12.**, Sequences C_H **13**, Sequences D_H **14**, Sequences E_H **15**, Sequences F_H **16**, Sequences
RES	15:12	r	**Reserved** Always read as 0

Field	Bits	Type	Description
SAMPLE_TIME_int	11:8	rw	**Sample time of ADC2** 0_H **MICLK4**, 4 MI_CLK clock periods 1_H **MICLK6**, 6 MI_CLK clock periods 2_H **MICLK8**, 8 MI_CLK clock periods 3_H **MICLK10**, 10 MI_CLK clock periods 4_H **MICLK12**, 12 MI_CLK clock periods 5_H **MICLK14**, 14 MI_CLK clock periods 6_H **MICLK16**, 16 MI_CLK clock periods 7_H **MICLK18**, 18 MI_CLK clock periods 8_H **MICLK20**, 20 MI_CLK clock periods 9_H **MICLK22**, 22 MI_CLK clock periods A_H **n.u.**, not used B_H **n.u.**, not used C_H **n.u.**, not used D_H **n.u.**, not used E_H **n.u.**, not used F_H **n.u.**, not used
MCM_RDY	7	r	**Ready Signal for MCM[1] after Power On or Reset** 0_B **MCM Not Ready**, Measurement Core Module in startup phase 1_B **MCM Ready**, Measurement Core Module start-up phase finished
RES	6:3	r	**Reserved** Always read as 0
TSENSE_SD_SEL	2	rw	**TSENSE Sensor selection for ADC2 channel 9** 0_B **PMU_TSENSE**, PMU temperature sensor selected 1_B **Central_TSENSE**, Central Temperature Sensor selected
TS_SD_SEL_CONF	1	rw	**Temperature Sensor Control Configuration** 0_B **CONSTANT**, Temperature Sensor Selection done by bit field TSENSE_SD_SEL 1_B **NOT ACTIVE**, Temperature Sensor Selection automatically done by DPP
MCM_PD_N	0	rw	**Power Down Signal for MCM** 0_B **MCM Disabled**, Measurement Core Module Disabled 1_B **MCM Enabled**, Measurement Core Module Enabled

1) MCM = Measurement Core Module

Measurement Unit Control Register 4

The register is reset by RESET_TYPE_4.

CTRL4
Measurement Unit Control Register 4 **($1C_H$)** **Reset Value: $0000\ 0F3F_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_SEL_9_6				RES		FILT_OUT_SEL_5_0					
r				rw				r		rw					

Field	Bits	Type	Description
RES	31:12	r	**Reserved** Always read as 0
FILT_OUT_SEL_9_6	11:8	rw	**Output Filter Selection for Channels 6 to 9** 0000_B**ADC2 Unfiltered Data** , can be monitored in the corresponding FILT_OUTx Registers.[1)] 0001_B**Channel 6 IIR Data enabled for FILT_OUT6 Register**, . 0010_B**Channel 7 IIR Data enabled for FILT_OUT7 Register**, . 0100_B**Channel 8 IIR Data enabled for FILT_OUT8 Register**, . 1000_B**Channel 9 IIR Data enabled for FILT_OUT9 Register**, . 1111_B**For Channels 9-6 IIR Data is enabled for FILT_OUTx Registers**, .
RES	7:6	r	**Reserved** Always read as 0

Field	Bits	Type	Description
FILT_OUT_SEL_5_0	5:0	rw	**Output Filter Selection for Channels 0 to 5** $00\ 0000_B$**ADC2 Unfiltered Data can be monitored in the corresponding FILT_OUTx Registers**, [1)] $00\ 0001_B$**Channel 0 IIR Data enabled for FILT_OUT0 Register**, . $00\ 0010_B$**Channel 1 IIR Data enabled for FILT_OUT1 Register**, . $00\ 0100_B$**Channel 2 IIR Data enabled forFILT_OUT2 Register**, . $00\ 1000_B$**Channel 3 IIR Data enabled for FILT_OUT3 Register**, . $01\ 0000_B$**Channel 4 IIR Data enabled for FILT_OUT4 Register**, . $10\ 0000_B$**Channel 5 IIR Data enabled for FILT_OUT5 Register**, . $11\ 1111_B$**For Channels 5-0 IIR Data is enabled for FILT_OUTx Registers**, .

1) The unfiltered data can be either the calibrated data or the raw data, depending on calib_en.

Measurement Channel Enable Bits for Cycle 1-4

The register is reset by RESET_TYPE_4.

SQ1_4
Measurement Channel Enable Bits for Cycle 1-4 (20_H) **Reset Value: $2936\ 2837_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		SQ4						RES		SQ3					
r		rw						r		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		SQ2						RES		SQ1					
r		rw						r		rw					

Field	Bits	Type	Description
RES	31:30	r	**Reserved** Always read as 0
SQ4	29:24	rw	**Sequence 4 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable
RES	23:22	r	**Reserved** Always read as 0
SQ3	21:16	rw	**Sequence 3 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable
RES	15:14	r	**Reserved** Always read as 0
SQ2	13:8	rw	**Sequence 2 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Always read as 0
SQ1	5:0	rw	**Sequence 1 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable

Measurement Channel Enable Bits for Cycle 1-8

The register is reset by RESET_TYPE_4.

SQ1_8_int
Measurement Channel Enable Bits for Cycle 1-8 ($2C_H$) **Reset Value: $7B7B\ 7B1B_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SQ8_int				**SQ7_int**				**SQ6_int**				**SQ5_int**			
rwp				rwp				rwp				rwp			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ4_int				**SQ3_int**				**SQ2_int**				**SQ1_int**			
rwp				rwp				rwp				rwp			

Field	Bits	Type	Description
SQ8_int	31:28	rwp	**Sequence 8 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ7_int	27:24	rwp	**Sequence 7 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ6_int	23:20	rwp	**Sequence 6 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ5_int	19:16	rwp	**Sequence 5 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ4_int	15:12	rwp	**Sequence 4 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable

Field	Bits	Type	Description
SQ3_int	11:8	rwp	**Sequence 3 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ2_int	7:4	rwp	**Sequence 2 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ1_int	3:0	rwp	**Sequence 1 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable

Measurement Channel Enable Bits for Cycle 5-8

The register is reset by RESET_TYPE_4.

SQ5_8
Measurement Channel Enable Bits for Cycle 5-8 (24_H) **Reset Value: $2837\ 2836_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		SQ8						RES		SQ7					
r		rw						r		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		SQ6						RES		SQ5					
r		rw						r		rw					

Field	Bits	Type	Description
RES	31:30	r	**Reserved** Always read as 0
SQ8	29:24	rw	**Sequence 8 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable
RES	23:22	r	**Reserved** Always read as 0
SQ7	21:16	rw	**Sequence 7 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable
RES	15:14	r	**Reserved** Always read as 0
SQ6	13:8	rw	**Sequence 6 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable

Field	Bits	Type	Description
RES	7:6	r	**Reserved** Always read as 0
SQ5	5:0	rw	**Sequence 5 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable

Measurement Channel Enable Bits for Cycle 9-10

The register is reset by RESET_TYPE_4.

SQ9_10
Measurement Channel Enable Bits for Cycle 9-10 (28_H) **Reset Value: $0000\ 2936_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		SQ10						RES		SQ9					
r		rw						r		rw					

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
RES	15:14	r	**Reserved** Always read as 0
SQ10	13:8	rw	**Sequence 10 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable
RES	7:6	r	**Reserved** Always read as 0
SQ9	5:0	rw	**Sequence 9 channel enable** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable

Measurement Channel Enable Bits for Cycle 9 & 10

The register is reset by RESET_TYPE_4.

SQ9_10_int
Measurement Channel Enable Bits for Cycle 9 and 10 (30_H) **Reset Value: $0000\ 007B_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								SQ10_int				SQ9_int			
r								rwp				rwp			

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
SQ10_int	7:4	rwp	**Sequence 10 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
SQ9_int	3:0	rwp	**Sequence 9 channel enable** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable

Sequencer Feedback Register

The register is reset by RESET_TYPE_3.

SQ_FB
Sequencer Feedback Register **(04_H)** **Reset Value:**
$0000\ 0000\ 000X\ XXXX\ 0XXX\ XXXX\ 0000\ XXXX_B$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											CHx				
r											r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	SQx				ESM_ACTIVE	EIM_ACTIVE	SQ_STOP	RES				SQ_FB			
r	r				r	r	r	r				r			

Field	Bits	Type	Description
RES	31:21	r	**Reserved** Always read as 0
CHx	20:16	r	**Current ADC2 Channel** Other bit combinations are **reserved**, do not use. 00000_B **CH0**, Channel 0 enable 00001_B **CH1**, Channel 1 enable 00010_B **CH2**, Channel 2 enable 00011_B **CH3**, Channel 3 enable 00100_B **CH4**, Channel 4 enable 00101_B **CH5**, Channel 5 enable 00110_B **CH6**, Channel 6 enable 11101_B **CH7**, Channel 7 enable 11110_B **CH8**, Channel 8 enable 11111_B **CH9**, Channel 9 enable
RES	15	r	**Reserved** Always read as 0
SQx	14:11	r	**Current Active Sequencer** Other bit combinations are **reserved**, do not use. 0000_B **SQ0**, Sequence 0 enable 0001_B **SQ1**, Sequence 1 enable 0010_B **SQ2**, Sequence 2 enable 0011_B **SQ3**, Sequence 3 enable 0100_B **SQ4**, Sequence 4 enable 0101_B **SQ5**, Sequence 5 enable 0110_B **SQ6**, Sequence 6 enable 0111_B **SQ7**, Sequence 7 enable 1000_B **SQ8**, Sequence 8 enable 1001_B **SQ9**, Sequence 9 enable

Field	Bits	Type	Description
ESM_ACTIVE	10	r	**ADC2 ESM active** 0_B **not active**, ESM not active 1_B **active**, ESM active
EIM_ACTIVE	9	r	**ADC2 EIM active** 0_B **not active** , EIM not active 1_B **active**, EIM active
SQ_STOP	8	r	**ADC2 Sequencer Stop Signal for DPP** 0_B **DPP Running**, Post-processing Sequencer in running mode 1_B **DPP Stopped**, Post-processing Sequencer stopped
RES	7:4	r	**Reserved** Always read as 0
SQ_FB	3:0	r	**Current Sequence** Other bit combinations are **n.u.**, not used. 0000_B **SQ1**, Sequence 1 0001_B **SQ2**, Sequence 2 0010_B **SQ3**, Sequence 3 0011_B **SQ4**, Sequence 4 0100_B **SQ5**, Sequence 5 0101_B **SQ6**, Sequence 6 0110_B **SQ7**, Sequence 7 0111_B **SQ8**, Sequence 8 1000_B **SQ9**, Sequence 9 1001_B **SQ10**, Sequence 10 1010_B **ESM**, ESM 1011_B **CH_MASK**, Channel Mask = 0 1100_B **SUSPEND**, Debug Suspend Mode

Channel Setting Bits for Exceptional Interrupt Measurement

The register is reset by RESET_TYPE_3.

CHx_EIM

Channel Setting Bits for Exceptional Interrupt Measurement (08_H) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			SEL	EN	REP			RES			CHx				
r			rw	rw	rw			r			rw				

Field	Bits	Type	Description
RES	31:13	r	**Reserved** Always read as 0
SEL	12	rw	**Exceptional interrupt measurement (EIM) Trigger select** 0_B **CCU6_SEL**, ccu6_int triggers EIM 1_B **CP_SEL**, cp_clk triggers EIM
EN	11	rw	**Exceptional interrupt measurement (EIM) Trigger Event enable** 0_B **DISABLE**, start of EIM disabled 1_B **ENABLE**, start of EIM enabled
REP	10:8	rw	**Repeat count for exceptional interrupt measurement (EIM)** 000_B **1**, Measurements 001_B **2**, Measurements 010_B **4**, Measurements 011_B **8**, Measurements 100_B **16**, Measurements 101_B **32**, Measurements 110_B **64**, Measurements 111_B **128**, Measurements
RES	7:5	r	**Reserved** Always read as 0
CHx	4:0	rw	**Channel set for exceptional interrupt measurement (EIM)** Other bit combinations are **n.u.**, not used. 00000_B **CH0_EN**, Channel 0 enable 00001_B **CH1_EN**, Channel 1 enable 00010_B **CH2_EN**, Channel 2 enable 00011_B **CH3_EN**, Channel 3 enable 00100_B **CH4_EN**, Channel 4 enable 00101_B **CH5_EN**, Channel 5 enable 00110_B **CH6_EN**, Channel 6 enable 11101_B **CH7_EN**, Channel 7 enable 11110_B **CH8_EN**, Channel 8 enable 11111_B **CH9_EN**, Channel 9 enable

Channel Setting Bits for Exceptional Sequence Measurement

The register is reset by RESET_TYPE_3.

CHx_ESM

Channel Setting Bits for Exceptional Sequence Measurement ($0C_H$) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES														STS	EN
r														r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					SEL	ESM_1				ESM_0					
r					rw	rw				rw					

Field	Bits	Type	Description
RES	31:18	r	**Reserved** Always read as 0
STS	17	r	**Exceptional Sequence Measurement is finished** 0_B **Not Completed**, Exceptional Sequence Measurement not completed 1_B **Completed**, Exceptional Sequence Measurement completed
EN	16	rw	**Enable for Exceptional Sequence Measurement Trigger Event** 0_B **Disable**, start of ESM disabled 1_B **Enable**, start of ESM enabled
RES	15:11	r	**Reserved** Always read as 0
SEL	10	rw	**Exceptional Sequence Measurement Trigger Select** 0_B **CCU6_SEL**, ccu6_int starts ESM 1_B **CP_SEL**, cp_clk starts ESM
ESM_1	9:6	rw	**Channel Sequence for Exceptional Sequence Measurement (ESM)** The following values can be or'ed: 0001_B **CH6_EN**, Channel 6 enable 0010_B **CH7_EN**, Channel 7 enable 0100_B **CH8_EN**, Channel 8 enable 1000_B **CH9_EN**, Channel 9 enable
ESM_0	5:0	rw	**Channel Sequence for Exceptional Sequence Measurement (ESM)** The following values can be or'ed: $00\ 0001_B$ **CH0_EN**, Channel 0 enable $00\ 0010_B$ **CH1_EN**, Channel 1 enable $00\ 0100_B$ **CH2_EN**, Channel 2 enable $00\ 1000_B$ **CH3_EN**, Channel 3 enable $01\ 0000_B$ **CH4_EN**, Channel 4 enable $10\ 0000_B$ **CH5_EN**, Channel 5 enable

23.5 Calibration Unit

23.5.1 Functional Description

The calibration unit of the Measurement Core module is dedicated to cancel offset and gain errors out of the signal chain. The upcoming two chapters describe usage and setup of the calibration unit.

23.5.1.1 Method for determining the Calibration Parameters

As mentioned in the introduction of the calibration unit, the module can be used to correct gain and offset errors caused by non-idealities in the measurement chain. These non-idealities are caused by the corresponding measurement chain modules.

Those first order non-idealities are:

- Offset and Gain Error of ADC2.
- Offset and Gain Error of the Attenuator (especially voltage measurement).
- Offset and Gain Error of Reference Voltage caused by non-ideality of reference voltage.

All these factors are summed up in the overall Gain (factor **b**) and overall Offset (adder **a**) of the complete measurement chain. They are calculated from a two point test result and stored inside the NVM.

Note: **The calibration of the HV-Monitoring-Pins was done with an external 1 kΩ resistor. The usage of a resistor with a bigger value requires a recalibration of the signal path due to a finite resistor value of the attenuator.**

23.5.1.2 Setup of Calibration Unit

Each channel has its own calibration unit and thus also its dedicated Gain and Offset parameter. These parameters are stored in a 100TP page of the Flash Module. After each reset of RESET_TYPE_4 these coefficients are downloaded from NVM into the corresponding registers. The user may not take care about the configuration of these parameters. After this has been done, the values are used for the correction procedure. The figure below shows the formula performed by the calibration unit and the required SFR-Register to control its functionality in a generic way.

The parameters ADC2_CALOFFS_CHx and ADC2_CALGAIN_CHx are stored in an 8 bit, 2th complement format.

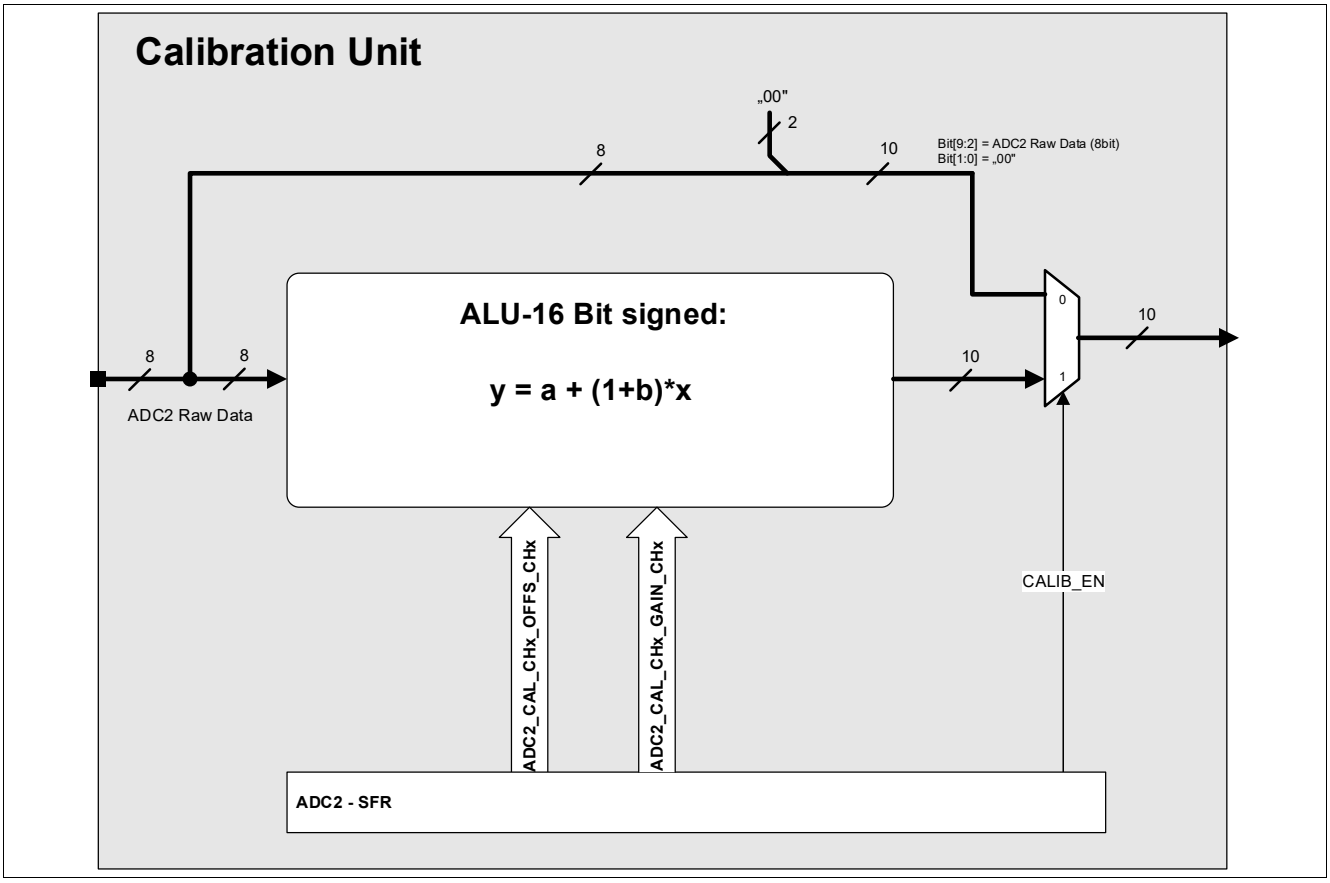


Figure 200 Structure of Calibration Unit

The function applied to calculate the calibrated ADC2 value is:

$$\text{ADC2_CHx_cal} = \left(1 + \frac{\text{ADC2_CALGAIN_CHx}}{256}\right) * \text{ADC2_CHx_uncal} + \frac{\text{ADC2_CALOFFS_CHx}}{2} \tag{23.6}$$

The calibration takes 12 MI_CLK cycles.

23.5.2 Calibration Unit Control Registers

The Calibration Unit can be configured by the **SFR** Register shown below. The registers which cannot be written by the user have the attribute **rwp**. Those registers are:

CAL_CH6_7,

CAL_CH8_9

Table 169 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Calibration Unit Control Registers			
CAL_CH0_1	ADC2 Calibration Value for Channel 0 & 1	34_H	$0000\ 0000_H$
CAL_CH2_3	ADC2 Calibration Value for Channel 2 & 3	38_H	$0000\ 0000_H$
CAL_CH4_5	ADC2 Calibration Value for Channel 4 & 5	$3C_H$	$0000\ 0000_H$
CAL_CH6_7	ADC2 Calibration Value for Channel 6 & 7	40_H	$0000\ 0000_H$
CAL_CH8_9	ADC2 Calibration Value for Channel 8 & 9	44_H	$0000\ 0000_H$

The registers are addressed wordwise.

ADC2 Calibration Value Channel 0 & 1

The register is reset by RESET_TYPE_4.

CAL_CH0_1
ADC2 Calibration Value for Channel 0 & 1 **(34_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH1								OFFS_CH1							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH0								OFFS_CH0							
rw								rw							

Field	Bits	Type	Description
GAIN_CH1	31:24	rw	**Gain Calibration for channel 1** For uncalibrated ADC output set CALIB_EN_1 = 0
OFFS_CH1	23:16	rw	**Offset Calibration for channel 1** For uncalibrated ADC output set CALIB_EN_1 = 0
GAIN_CH0	15:8	rw	**Gain Calibration for channel 0** For uncalibrated ADC output set CALIB_EN_0 = 0
OFFS_CH0	7:0	rw	**Offset Calibration for channel 0** For uncalibrated ADC output set CALIB_EN_0 = 0

ADC2 Calibration Value for Channel 2 & 3

The register is reset by RESET_TYPE_4.

CAL_CH2_3
ADC2 Calibration Value for Channel 2 & 3 **(38_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH3								OFFS_CH3							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH2								OFFS_CH2							
rw								rw							

Field	Bits	Type	Description
GAIN_CH3	31:24	rw	**Gain Calibration for channel 3** For ADC output set CALIB_EN_3 = 0
OFFS_CH3	23:16	rw	**Offset Calibration for channel 3** For ADC output set CALIB_EN_3 = 0
GAIN_CH2	15:8	rw	**Gain Calibration for channel 2** For ADC output set CALIB_EN_2 = 0
OFFS_CH2	7:0	rw	**Offset Calibration for channel 2** For ADC output set CALIB_EN_2 = 0

ADC2 Calibration Value for Channel 4 & 5

The register is reset by RESET_TYPE_4.

CAL_CH4_5
ADC2 Calibration Value for Channel 4 & 5 **($3C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH5								OFFS_CH5							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH4								OFFS_CH4							
rw								rw							

Field	Bits	Type	Description
GAIN_CH5	31:24	rw	**Gain Calibration for channel 5** For ADC output set CALIB_EN_5 = 0
OFFS_CH5	23:16	rw	**Offset Calibration for channel 5** For ADC output set CALIB_EN_5 = 0
GAIN_CH4	15:8	rw	**Gain Calibration for channel 4** For ADC output set CALIB_EN_4 = 0
OFFS_CH4	7:0	rw	**Offset Calibration for channel 4** For ADC output set CALIB_EN_4 = 0

ADC2 Calibration Value for Channel 6 & 7

The register is reset by RESET_TYPE_4.

CAL_CH6_7
ADC2 Calibration Value for Channel 6 & 7 **(40_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH7								OFFS_CH7							
rwp								rwp							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH6								OFFS_CH6							
rwp								rwp							

Field	Bits	Type	Description
GAIN_CH7	31:24	rwp	**Gain Calibration for channel 7** For ADC output set CALIB_EN_7 = 0
OFFS_CH7	23:16	rwp	**Offset Calibration for channel 7** For ADC output set CALIB_EN_7 = 0
GAIN_CH6	15:8	rwp	**Gain Calibration for channel 6** For ADC output set CALIB_EN_6 = 0
OFFS_CH6	7:0	rwp	**Offset Calibration for channel 6** For ADC output set CALIB_EN_6 = 0

ADC2 Calibration Value for Channel 8 & 9

The register is reset by RESET_TYPE_4.

CAL_CH8_9
ADC2 Calibration Value for Channel 8 & 9 **(44_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH9								OFFS_CH9							
rwp								rwp							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH8								OFFS_CH8							
rwp								rwp							

Field	Bits	Type	Description
GAIN_CH9	31:24	rwp	**Gain Calibration for channel 9** For ADC output set CALIB_EN_9 = 0
OFFS_CH9	23:16	rwp	**Offset Calibration for channel 9** For ADC output set CALIB_EN_9 = 0
GAIN_CH8	15:8	rwp	**Gain Calibration for channel 8** For ADC output set CALIB_EN_8 = 0
OFFS_CH8	7:0	rwp	**Offset Calibration for channel 8** For ADC output set CALIB_EN_8 = 0

23.6 IIR-Filter

23.6.1 Functional Description

To cancel low frequency noise out of the measured signal, every channel of the digital signal includes a first order IIR Filter. The structure of the IIR Filter is shown in the picture below.

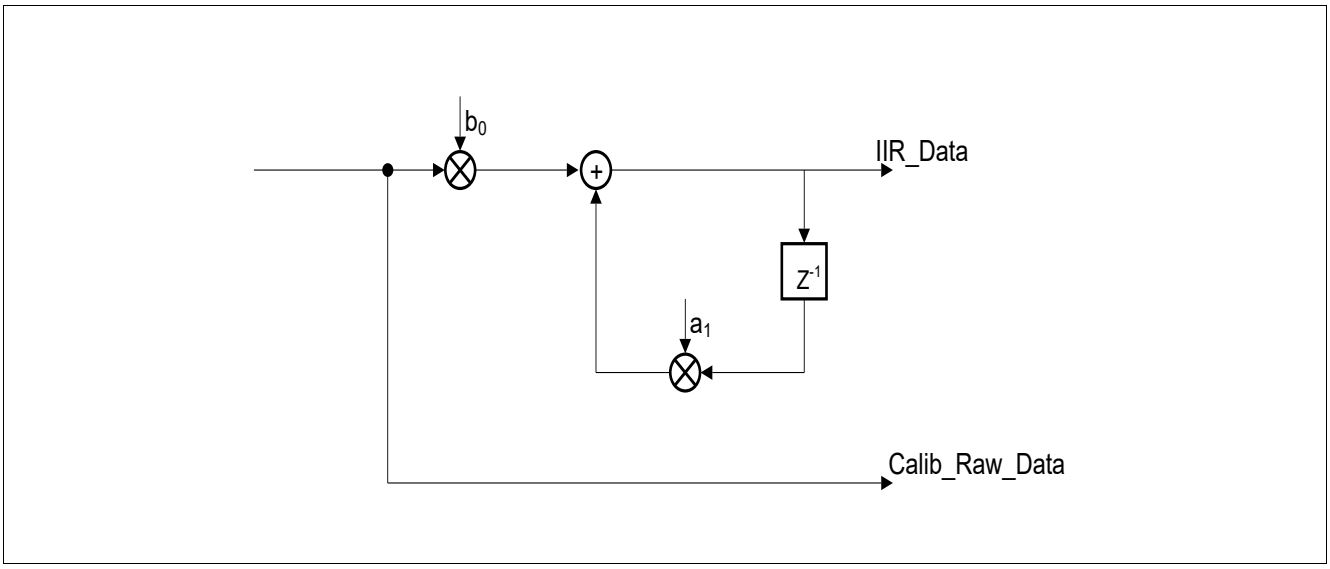


Figure 201 IIR-Filter Implementation Structure

$$H_{IIR}(z) = \frac{b_0}{(1 - a_1 * z^{-1})} \tag{23.7}$$

This filter allows an effective suppression of high-frequency components like noise or crosstalk caused by HF-components in order to avoid the generation of unwanted interrupts. The coefficient b can be expressed as:

$$a_1 = 1 - b_0 \tag{23.8}$$

With the coefficient b implemented in the IIR Filter transfer function, it looks like:

$$H_{IIR}(z) = \frac{b_0}{(1 - (1 - b_0) * z^{-1})} \tag{23.9}$$

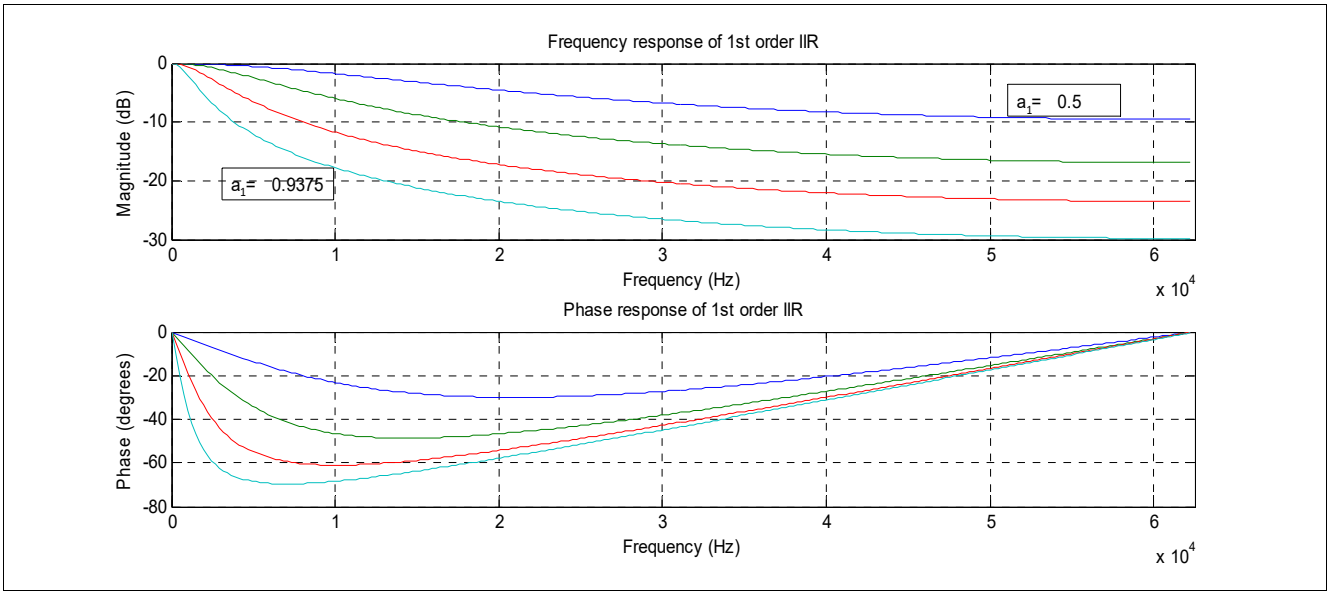


Figure 202 IIR Filter Transfer Function for different filter length fl (sampling frequency is assumed to be 125 kHz)

23.6.1.1 Step Response

The IIR filter's step response time is shown in the figure below:

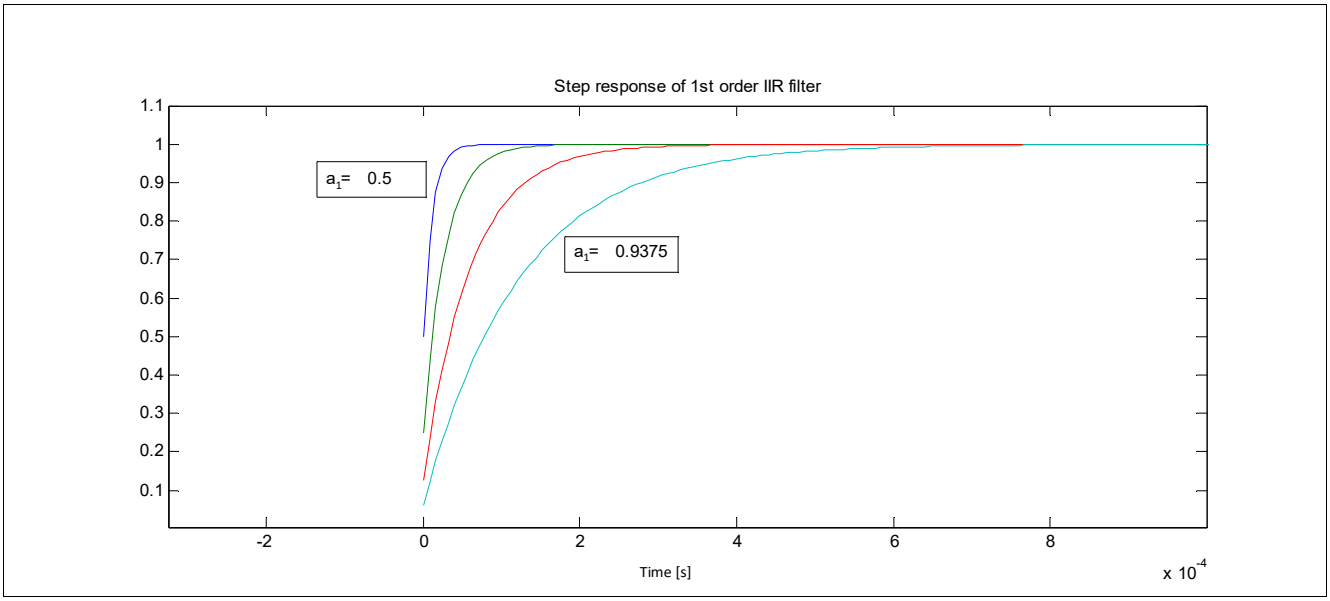


Figure 203 IIR Step Response Time

Table 170 summarizes the main filter characteristics.

Table 170 IIR filter characteristics

Filter coefficient	Group delay at=ω0
b_0	τ[samples]
2^{-1}	2
2^{-2}	4
2^{-3}	8
2^{-4}	16

The propagation delay of the IIR filter is 5 clock cycles, based on MI_CLK.

23.6.2 IIR Filter Control Registers

The IIR Filter can also be configured by the SFR Register shown below. The registers which cannot be written by the user have the attribute **rwp**. Those registers are:

FILTCOEFF6_9

The **FILT_OUT0** to **FILT_OUT9** registers are 10 bits wide, but the ADC delivers only a resolution of 8 bits. **Table 171** shows how the lower two bits are determined.

Table 171 ADC2_FILT_OUT register setting

CTRL1.CALIB_EN	CTRL4.FILT_OUT_SEL	FILT_OUT0.OUT_CH0
0	0	“00”
0	1	“filt_out(3:2)”
1	0	“calib_out(1:0)”
1	1	“filt_out(3:2)”

The result of the calibration unit is 10 bits, the output is feed into the IIR filter. The internal result of the IIR filter is 12 bits, the output is converted to 10 bit and fed into the post-processing. The user can monitor the calculated values in the **FILT_OUT0** to **FILT_OUT9** registers and gets access to 10 bit wide result information.

Table 172 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
IIR Filter Control Registers			
FILTCOEFF0_5	Filter Coefficients ADC Channel 0-5	48_H	$0000\ 0AAA_H$
FILTCOEFF6_9	Filter Coefficients ADC Channel 6-9	$4C_H$	$0000\ 00D5_H$
FILT_OUT0	ADC or Filter Output Channel 0	50_H	0000 0000 0000 0000 $0000\ 00XX\ XXXX\ XXXX_B$
FILT_OUT1	ADC or Filter Output Channel 1	54_H	0000 0000 0000 0000 $0000\ 00XX\ XXXX\ XXXX_B$
FILT_OUT2	ADC or Filter Output Channel 2	58_H	0000 0000 0000 0000 $0000\ 00XX\ XXXX\ XXXX_B$
FILT_OUT3	ADC or Filter Output Channel 3	$5C_H$	0000 0000 0000 0000 $0000\ 00XX\ XXXX\ XXXX_B$

Table 172 Register Overview (cont'd)

Register Short Name	Register Long Name	Offset Address	Reset Value
FILT_OUT4	ADC or Filter Output Channel 4	60_H	0000 0000 0000 0000 0000 00XX XXXX XXXX_B
FILT_OUT5	ADC or Filter Output Channel 5	64_H	0000 0000 0000 0000 0000 00XX XXXX XXXX_B
FILT_OUT6	ADC or Filter Output Channel 6	68_H	0000 0000 0000 0000 0000 00XX XXXX XXXX_B
FILT_OUT7	ADC or Filter Output Channel 7	$6C_H$	0000 0000 0000 0000 0000 00XX XXXX XXXX_B
FILT_OUT8	ADC or Filter Output Channel 8	70_H	0000 0000 0000 0000 0000 00XX XXXX XXXX_B
FILT_OUT9	ADC or Filter Output Channel 9	74_H	0000 0000 0000 0000 0000 00XX XXXX XXXX_B

The registers are addressed wordwise.

Filter Coefficients ADC Channel 0-5

The register is reset by RESET_TYPE_4.

FILTCOEFF0_5
Filter Coefficients ADC Channel 0-5 **(48_H)** **Reset Value: $0000\ 0AAA_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				CH5		CH4		CH3		CH2		CH1		CH0	
r				rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
RES	31:12	r	**Reserved** Always read as 0
CH5	11:10	rw	**Filter Coefficient b_0 for ADC channel 5** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH4	9:8	rw	**Filter Coefficient b_0 for ADC channel 4** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH3	7:6	rw	**Filter Coefficient b_0 for ADC channel 3** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH2	5:4	rw	**Filter Coefficient b_0 for ADC channel 2** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH1	3:2	rw	**Filter Coefficient b_0 for ADC channel 1** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample

Field	Bits	Type	Description
CH0	1:0	rw	**Filter Coefficient b_0 for ADC channel 0** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample

Filter Coefficients ADC Channel 6-9

The register is reset by RESET_TYPE_4.

FILTCOEFF6_9
Filter Coefficients ADC Channel 6-9 **($4C_H$)** **Reset Value: $0000\ 00D5_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								CH9		CH8		CH7		CH6	
r								rwp		rwp		rwp		rwp	

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
CH9	7:6	rwp	**Filter Coefficients ADC channel 9** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH8	5:4	rwp	**Filter Coefficients ADC channel 8** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH7	3:2	rwp	**Filter Coefficients ADC channel 7** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample
CH6	1:0	rwp	**Filter Coefficients ADC channel 6** 00_B **1/2**, weight of current sample 01_B **1/4**, weight of current sample 10_B **1/8**, weight of current sample 11_B **1/16**, weight of current sample

ADC or Filter Output Channel 0

The register is reset by RESET_TYPE_3. This registers reflects the current value of channel 0 of the measurement chain.

FILT_OUT0
ADC or Filter Output Channel 0 **(50_H)** **Reset Value: $0000\ 0000\ 0000\ 0000\ 0000\ 00XX\ XXXX\ XXXX_B$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH0									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH0	9:0	r	**ADC2 output value channel 0** For filtered output set **CTRL4**.FILT_OUT_SEL_5_0[0] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_5_0[0] = 0

ADC or Filter Output Channel 1

The register is reset by RESET_TYPE_3.

FILT_OUT1
ADC or Filter Output Channel 1 **(54_H)** **Reset Value: $0000\ 0000\ 0000\ 0000\ 0000\ 00XX\ XXXX\ XXXX_B$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH1									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH1	9:0	r	**ADC or filter output value channel 1** For filtered output set **CTRL4**.FILT_OUT_SEL_5_0[1] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_5_0[1] = 0

ADC or Filter Output Channel 2

The register is reset by RESET_TYPE_3.

FILT_OUT2
ADC or Filter Output Channel 2 **(58_H)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH2									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH2	9:0	r	**ADC or filter output value channel 2** For filtered output set **CTRL4**.FILT_OUT_SEL_5_0[2] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_5_0[2] = 0

ADC or Filter Output Channel 3

The register is reset by RESET_TYPE_3.

FILT_OUT3
ADC or Filter Output Channel 3 **($5C_H$)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH3									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH3	9:0	r	**ADC or filter output value channel 3** For filtered output set **CTRL4**.FILT_OUT_SEL_5_0[3] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_5_0[3] = 0

ADC or Filter Output Channel 4

The register is reset by RESET_TYPE_3.

FILT_OUT4
ADC or Filter Output Channel 4 **(60_H)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH4									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH4	9:0	r	**ADC or filter output value channel 4** For filtered output set **CTRL4**.FILT_OUT_SEL_5_0[4] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_5_0[4] = 0

ADC or Filter Output Channel 5

The register is reset by RESET_TYPE_3.

FILT_OUT5
ADC or Filter Output Channel 5 **(64_H)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH5									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH5	9:0	r	**ADC or filter output value channel 5** For filtered output set **CTRL4**.FILT_OUT_SEL_5_0[5] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_5_0[5] = 0

ADC or Filter Output Channel 6

The register is reset by RESET_TYPE_3.

FILT_OUT6
ADC or Filter Output Channel 6 **(68_H)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH6									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH6	9:0	r	**ADC or filter output value channel 6** For filtered output set **CTRL4**.FILT_OUT_SEL_9_6[0] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_9_6[0] = 0

ADC or Filter Output Channel 7

The register is reset by RESET_TYPE_3.

FILT_OUT7
ADC or Filter Output Channel 7 **($6C_H$)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH7									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH7	9:0	r	**ADC or filter output value channel 7** For filtered output set **CTRL4**.FILT_OUT_SEL_9_6[1] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_9_6[1] = 0

ADC or Filter Output Channel 8

The register is reset by RESET_TYPE_3.

FILT_OUT8
ADC or Filter Output Channel 8 **(70_H)** **Reset Value: 0000 0000 0000 0000 0000 00XX XXXX XXXX_B**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH8									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH8	9:0	r	**ADC or filter output value channel 8** For filtered output set **CTRL4**.FILT_OUT_SEL_9_6[2] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_9_6[2] = 0

ADC or Filter Output Channel 9

The register is reset by RESET_TYPE_3.

FILT_OUT9
ADC or Filter Output Channel 9 **(74_H)** **Reset Value: $0000\ 0000\ 0000\ 0000\ 0000\ 00XX\ XXXX\ XXXX_B$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH9									
r						r									

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
OUT_CH9	9:0	r	**ADC or filter output value channel 9** For filtered output set **CTRL4**.FILT_OUT_SEL_9_6[3] = 1 For unfiltered output set **CTRL4**.FILT_OUT_SEL_9_6[3] = 0

23.7 Signal Processing

23.7.1 Functional Description

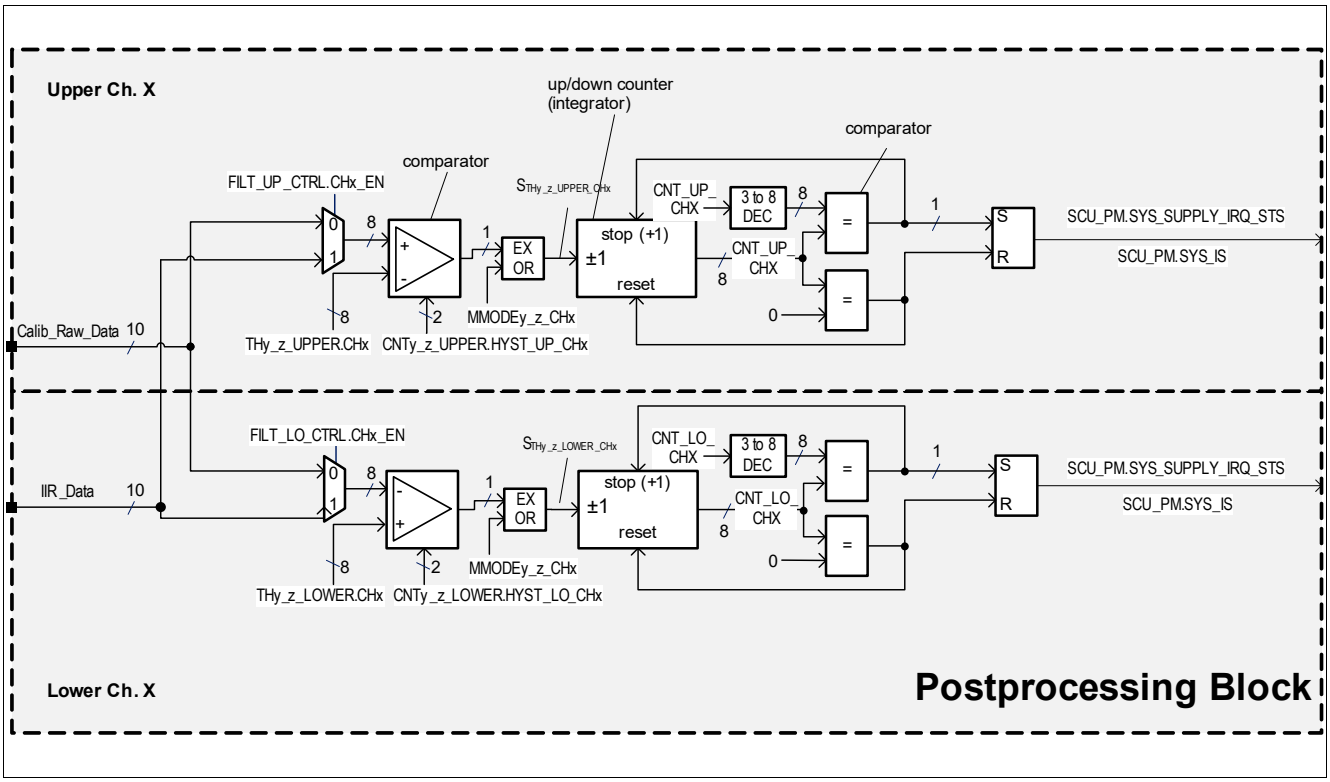


Figure 204 Post-processing channel block diagram for voltage and temperature measurements

As shown in **Figure 204** an adjustable filter can be applied for the upper and the lower measurement channel, which averages 2, 4, 8 or 16 measurement values continuously. The filtered signal or the demultiplexed ADC output signal ADC_OUTx is compared with an upper threshold THy_z_UPPPER.CHx and a lower threshold THy_z_LOWER.CHx. When the thresholds are exceeded, the comparator outputs get active. For all measurement modes a freely adjustable hysteresis can be defined which is defined with the CNTy_z_LOWER.HYST_UP_CHx and CNTy_z_LOWER.HYST_LO_CHx values.

In addition to the first filter stage, the second filters (counters) integrate the comparator output values S_{TH_UP/LO_CHX} until an individual upper and lower timing threshold $2^{CNT_UP/LO_CHX}$ is reached. When reaching the upper timing threshold $2^{CNT_UP_CHX}$, the upper counter increment is stalled and the status output CHx_UP_STS is set. For MMODE_OV = 1, the inverted lower comparator output signal $S_{TH_LO_CHX}$ is normalized again. When the output signal is above THy_z_LOWER.CHx, the lower counter is incremented until the max. threshold $2^{CNT_LO_CHX}$ is reached. Individual interrupts for the upper and lower channel can be triggered with the rising edge of the status signals UP/LO_x_STS.

In general the IIR filter stage suppresses higher frequency noise efficiently and triggering with the upper and lower threshold TH_UP/LO_CHx are dependent on the measured values. Hence short high-level spikes might pass the thresholds. In opposite to the first stage the nature of the second filter stage is more a time filter, which is less dependent on the measurement values but on event durations of S_{TH_LO/UP_CHX} as generated by the first comparator stage. Therefore the second stage has a lower noise suppression performance for higher frequencies and also adds a delay for the trigger time proportional to $2^{CNT_LO/HI_CHX}$.

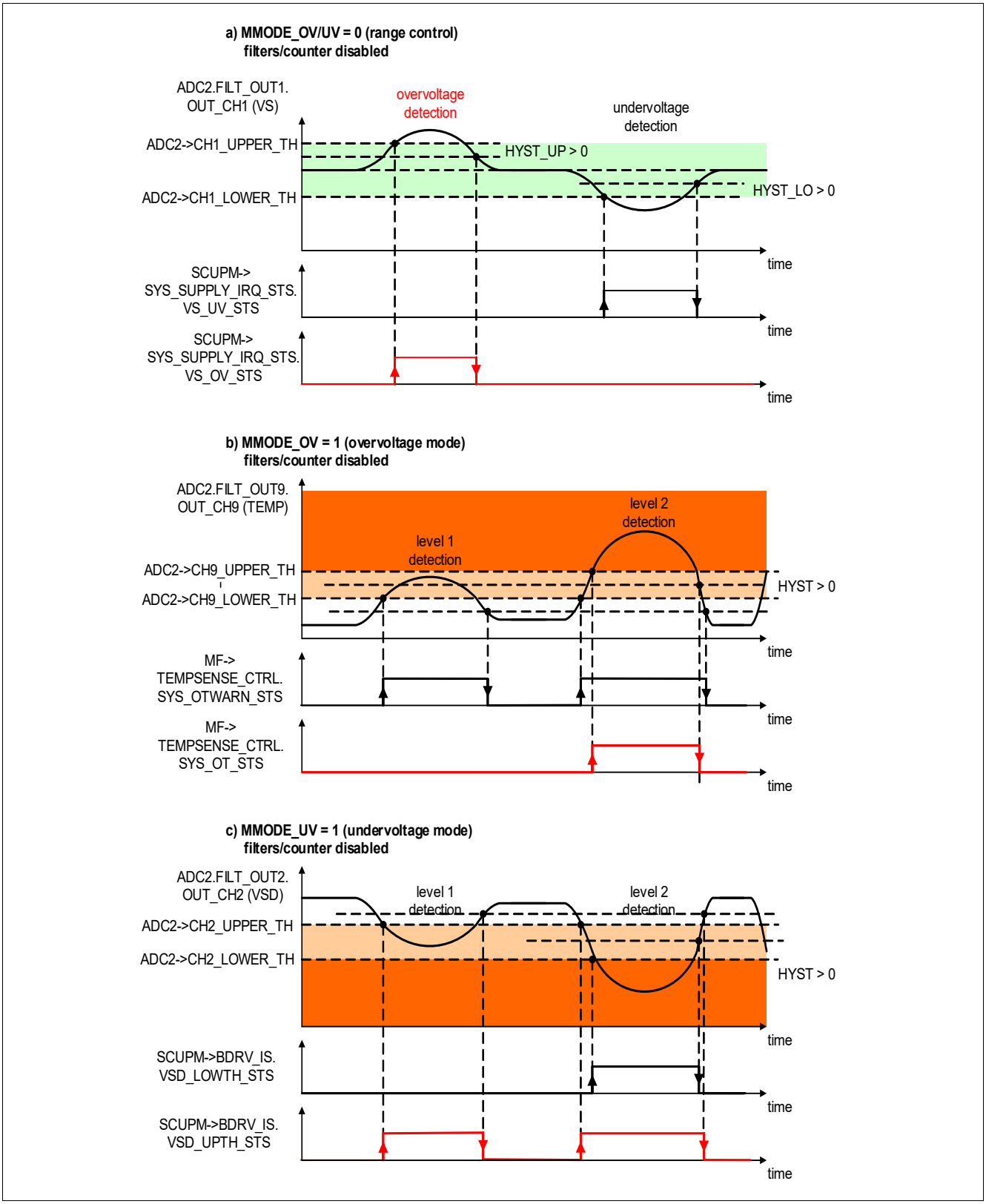


Figure 205 Measurement examples of a measurement channel with disabled filters

Figure 205 shows three examples, a range-control and an over- and undervoltage detection.

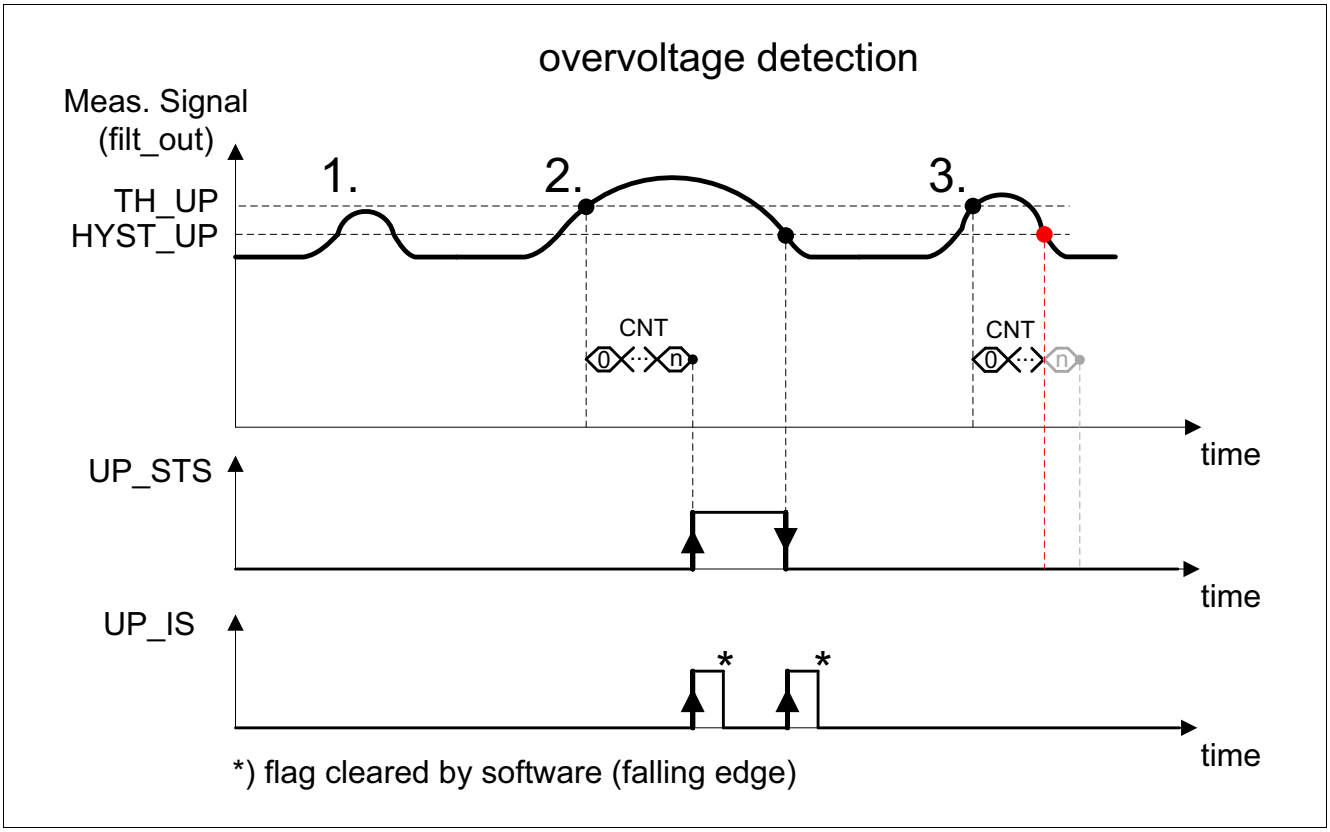


Figure 206 Overvoltage detection

Figure 206 shows the overvoltage (upper threshold) detection in three cases. Overvoltage is detected based on the digital value either provided by the calibration unit, or the IIR filter unit, see **FILT_UP_CTRL**. If the digital value exceeds the TH_UP value (**TH0_3_UPPER**, **TH4_5_UPPER** or **TH6_9_UPPER**) then the post-processing unit recognized the overvoltage condition. The exceedance of the TH_UP starts an internal counter (based on MI_CLK), configured in the registers **CNT0_3_UPPER**, **CNT4_5_UPPER** and **CNT6_9_UPPER**. Once the timer is elapsed the status is being set (UP_STS), along with that the interrupt status is set. The interrupt status has to be cleared by software.

If the digital value falls below the HYST_UP value, which is being defined in **CNT0_3_UPPER**, **CNT4_5_UPPER** and **CNT6_9_UPPER**, the status is being reset. Along with that the interrupt status is set, in order to signal the leaving of the overvoltage zone to the application. The interrupt status has to be cleared by software. The hysteresis threshold is defined by subtracting the hysteresis register value from the upper threshold value:

Example for channel1:

Hysteresis threshold = TH0_3_UPPER.CH1 - CNT0_3_UPPER.HYST_UP_CH1.

Description of the three cases in **Figure 206**:

1. the digital value does not exceed the UP_TH value, no overvoltage detected, no status being set
2. the digital value exceeds the UP_TH value, the counter starts counting and elapses, the status is being set. The digital value falls below the HYST_UP and resets the status.
3. the digital value exceeds the UP_TH value, the counter starts counting, before the counter elapses the digital value falls below the HYST_UP value. No overvoltage is being signalled.

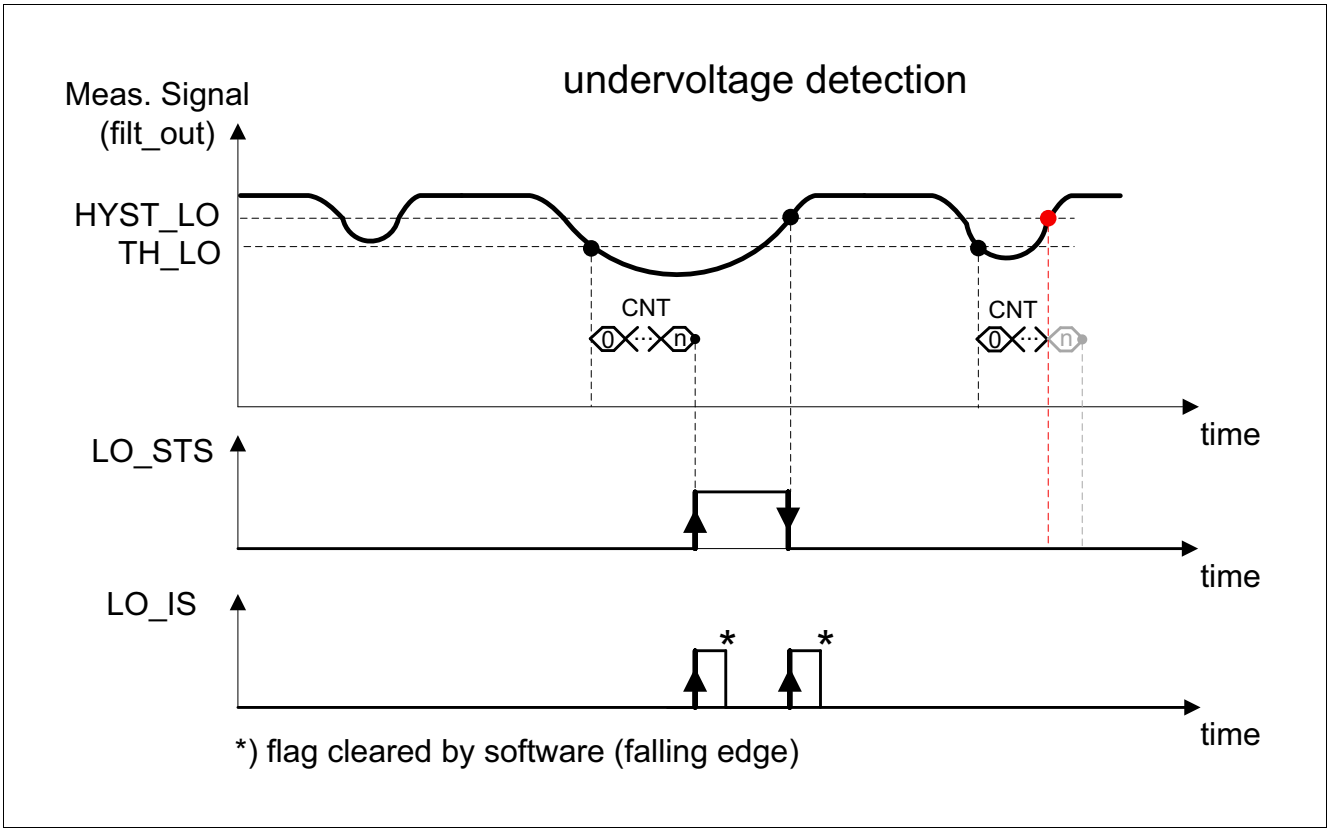


Figure 207 Undervoltage detection

Figure 207 shows the undervoltage (lower threshold) detection in three cases. Undervoltage is detected based on the digital value either provided by the calibration unit, or the IIR filter unit, see **FILT_LO_CTRL**. If the digital value exceeds the TH_LO value (**TH0_3_LOWER**, **TH4_5_LOWER** or **TH6_9_LOWER**) then the post-processing unit recognized the undervoltage condition. The undershoot of the digital value below TH_LO starts an internal counter (based on MI_CLK), configured in the registers **CNT0_3_LOWER**, **CNT4_5_LOWER** and **CNT6_9_LOWER**. Once the timer is elapsed the status is being set (LO_STS), along with that the interrupt status is set. The interrupt status has to be cleared by software.

If the digital value rises above the HYST_UP value, which is being defined in **CNT0_3_LOWER**, **CNT4_5_LOWER** and **CNT6_9_LOWER**, the status is being reset. Along with that the interrupt status is set, in order to signal the leaving of the undervoltage zone to the application. The interrupt status has to be cleared by software. The hysteresis threshold is defined by adding the hysteresis register value to the lower threshold value:

Example for channel1:

Hysteresis threshold = TH0_3_LOWER.CH1 + CNT0_3_LOWER.HYST_LO_CH1.

Description of the three cases in **Figure 207**:

1. the digital value does not fall below LO_TH value, no undervoltage detected, no status being set
2. the digital value falls below the LO_TH value, the counter starts counting and elapses, the status is being set. The digital value rises above the HYST_LO and resets the status.
3. the digital value falls below the LO_TH value, the counter starts counting, before the counter elapses the digital value rises above the HYST_LO value. No undervoltage is being signalled.

23.7.2 Post-processing Control Registers

The Temperature Sensor is fully controllable by the below listed SFR Registers.

Table 173 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Post-processing Control Registers			
FILT_UP_CTRL	Upper Threshold Filter Enable	78_H	$0000\ 003F_H$
FILT_LO_CTRL	Lower Threshold Filter Enable	$7C_H$	$0000\ 003F_H$
TH0_3_LOWER	Lower Comparator Trigger Level Channel 0-3	80_H	$3C2C\ 3A42_H$
TH4_5_LOWER	Lower Comparator Trigger Level Channel 4&5	84_H	$0000\ 9A2C_H$
TH6_9_LOWER	Lower Comparator Trigger Level Channel 6-9	88_H	$C7D3\ BBDB_H$
TH0_3_UPPER	Upper Comparator Trigger Level Channel 0-3	$8C_H$	$A8AB\ DAE2_H$
TH4_5_UPPER	Upper Comparator Trigger Level Channel 4&5	90_H	$0000\ BC42_H$
TH6_9_UPPER	Upper Comparator Trigger Level Channel 6-9	94_H	$E2FA\ C6EE_H$
CNT0_3_LOWER	Lower Counter Trigger Level Channel 0-3	98_H	$1213\ 1312_H$
CNT4_5_LOWER	Lower Counter Trigger Level Channel 4&5	$9C_H$	$0000\ 0A0A_H$
CNT6_9_LOWER	Lower Counter Trigger Level Channel 6-9	$A0_H$	$0A0A\ 0A0A_H$
CNT0_3_UPPER	Upper Counter Trigger Level Channel 0-3	$A4_H$	$1213\ 1B1A_H$
CNT4_5_UPPER	Upper Counter Trigger Level Channel 4&5	$A8_H$	$0000\ 1212_H$
CNT6_9_UPPER	Upper Counter Trigger Level Channel 6-9	AC_H	$1A1A\ 1911_H$
MMODE0_5	Overvoltage Measurement Mode of Ch 0-5	$B0_H$	$0000\ 0000_H$

The registers are addressed wordwise.

Upper Threshold Filter Enable

Setting the corresponding channel configuration flag connects the IIR filter output to the post-processing for upper threshold detection of this channel.

The register is reset by RESET_TYPE_4.

FILT_UP_CTRL
Upper Threshold Filter Enable(**(78_H)** **Reset Value: 0000 $003F_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				RES				RES		Ch5_EN	Ch4_EN	Ch3_EN	Ch2_EN	Ch1_EN	Ch0_EN
r				r				r		rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:12	r	**Reserved** Always read as 0
RES	11:8	r	**Reserved** Always read as 0
RES	7:6	r	**Reserved** Always read as 0
Ch5_EN	5	rw	**Upper threshold IIR filter enable ch 5** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch4_EN	4	rw	**Upper threshold IIR filter enable ch 4** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch3_EN	3	rw	**Upper threshold IIR filter enable ch 3** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch2_EN	2	rw	**Upper threshold IIR filter enable ch 2** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch1_EN	1	rw	**Upper threshold IIR filter enable ch 1** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch0_EN	0	rw	**Upper threshold IIR filter enable ch 0** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled

Lower Threshold Filter Enable

The register is reset by RESET_TYPE_4. Setting the corresponding channel configuration flag connects the IIR filter output to the post-processing for lower threshold detection of this channel.

FILT_LO_CTRL
Lower Threshold Filter Enable **($7C_H$)** **Reset Value: 0000 $003F_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				RES				RES		Ch5_EN	Ch4_EN	Ch3_EN	Ch2_EN	Ch1_EN	Ch0_EN
r				r				r		rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:12	r	**Reserved** Always read as 0
RES	11:8	r	**Reserved** Always read as 0
RES	7:6	r	**Reserved** Always read as 0
Ch5_EN	5	rw	**Lower threshold IIR filter enable ch 5** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch4_EN	4	rw	**Lower threshold IIR filter enable ch 4** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch3_EN	3	rw	**Lower threshold IIR filter enable ch 3** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch2_EN	2	rw	**Lower threshold IIR filter enable ch 2** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch1_EN	1	rw	**Lower threshold IIR filter enable ch 1** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled
Ch0_EN	0	rw	**Lower threshold IIR filter enable ch 0** 0_B **DISABLE**, filter disabled 1_B **ENABLE**, filter enabled

Overvoltage Measurement Mode of Ch 0-5

The register is reset by RESET_TYPE_4.

MMODE0_5
Overvoltage Measurement Mode of Ch 0-5 **($B0_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				Ch5		Ch4		Ch3		Ch2		Ch1		Ch0	
r				rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
RES	31:12	r	**Reserved** Always read as 0
Ch5	11:10	rw	**Measurement mode ch 5** 00_B **MMODE0**, upper & lower voltage/limit measurement 01_B **MMODEUV**, undervoltage/-limit measurement 10_B **MMODEOV**, overvoltage/-limit measurement 11_B **RESERVED**, reserved
Ch4	9:8	rw	**Measurement mode ch 4** 00_B **MMODE0**, upper & lower voltage/limit measurement 01_B **MMODEUV**, undervoltage/-limit measurement 10_B **MMODEOV**, overvoltage/-limit measurement 11_B **RESERVED**, reserved
Ch3	7:6	rw	**Measurement mode ch 3** 00_B **MMODE0**, upper & lower voltage/limit measurement 01_B **MMODEUV**, undervoltage/-limit measurement 10_B **MMODEOV**, overvoltage/-limit measurement 11_B **RESERVED**, reserved
Ch2	5:4	rw	**Measurement mode ch 2** 00_B **MMODE0**, upper & lower voltage/limit measurement 01_B **MMODEUV**, undervoltage/-limit measurement 10_B **MMODEOV**, overvoltage/-limit measurement 11_B **RESERVED**, reserved
Ch1	3:2	rw	**Measurement mode ch 1** 00_B **MMODE0**, upper & lower voltage/limit measurement 01_B **MMODEUV**, undervoltage/-limit measurement 10_B **MMODEOV**, overvoltage/-limit measurement 11_B **RESERVED**, reserved

Field	Bits	Type	Description
Ch0	1:0	rw	**Measurement mode ch 0** 00_B **MMODE0**, upper & lower voltage/limit measurement 01_B **MMODEUV**, undervoltage/-limit measurement 10_B **MMODEOV**, overvoltage/-limit measurement 11_B **RESERVED**, reserved

Upper Comparator Trigger Level Channel 0-3

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of ABBD $C5C0_H$.

TH0_3_UPPER

Upper Comparator Trigger Level Channel 0-3 ($8C_H$) Reset Value: $A8AB\ DAE2_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3								**CH2**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1								**CH0**							
rw								rw							

Field	Bits	Type	Description
CH3	31:24	rw	**Channel 3 upper trigger level, ADC2_CH3_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH2	23:16	rw	**Channel 2 upper trigger level, ADC2_CH2_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH1	15:8	rw	**Channel 1 upper trigger level, ADC2_CH1_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH0	7:0	rw	**Channel 0 upper trigger level, ADC2_CH0_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255

Upper Comparator Trigger Level Channel 4 & 5

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of 0000 $BC00_H$.

TH4_5_UPPER

Upper Comparator Trigger Level Channel 4&5 (90_H) **Reset Value: 0000 $BC42_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH5								CH4							
rw								rw							

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
CH5	15:8	rw	**Channel 5 upper trigger level, ADC2_CH5_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH4	7:0	rw	**Channel 4 upper trigger level, ADC2_CH4_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255

Upper Comparator Trigger Level Channel 6-9

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of DFFA CBFA$_H$.

TH6_9_UPPER

Upper Comparator Trigger Level Channel 6-9 (94_H) Reset Value: E2FA C6EE$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH9								CH8							
rwp								rwp							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH7								CH6							
rwp								rwp							

Field	Bits	Type	Description
CH9	31:24	rwp	**Channel 9 upper trigger level, ADC2_CH9_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH8	23:16	rwp	**Channel 8 upper trigger level, ADC2_CH8_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH7	15:8	rwp	**Channel 7 upper trigger level, ADC2_CH7_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255
CH6	7:0	rwp	**Channel 6 upper trigger level, ADC2_CH6_UPPER_TH** Corresponding voltage value see equation. 00_H **MINTH**, min. threshold value = 0 **...** FF_H **MAXTH**, max. threshold value = 255

Upper Counter Trigger Level Channel 0-3

The register is reset by RESET_TYPE_4.

CNT0_3_UPPER
Upper Counter Trigger Level Channel 0-3 **($A4_H$)** **Reset Value: 1213 1B1A_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_UP_CH3		CNT_UP_CH3			RES			HYST_UP_CH2		CNT_UP_CH2		
r			rw		rw			r			rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			HYST_UP_CH1		CNT_UP_CH1			RES			HYST_UP_CH0		CNT_UP_CH0		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
RES	31:29	r	**Reserved** Always read as 0
HYST_UP_CH3	28:27	rw	**Channel 3 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH3	26:24	rw	**Upper timer trigger threshold channel 3** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	23:21	r	**Reserved** Always read as 0
HYST_UP_CH2	20:19	rw	**Channel 2 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16

Field	Bits	Type	Description
CNT_UP_CH2	18:16	rw	**Upper timer trigger threshold channel 2** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	15:13	r	**Reserved** Always read as 0
HYST_UP_CH1	12:11	rw	**Channel 1 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH1	10:8	rw	**Upper timer trigger threshold channel 1** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	7:5	r	**Reserved** Always read as 0
HYST_UP_CH0	4:3	rw	**Channel 0 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH0	2:0	rw	**Upper timer trigger threshold channel 0** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements

Upper Counter Trigger Level Channel 4 & 5

The register is reset by RESET_TYPE_4.

CNT4_5_UPPER
Upper Counter Trigger Level Channel 4&5 **($A8_H$)** **Reset Value: $0000\ 1212_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			HYST_UP_CH5		CNT_UP_CH5			RES			HYST_UP_CH4		CNT_UP_CH4		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
RES	31:13	r	**Reserved** Always read as 0
HYST_UP_CH5	12:11	rw	**Channel 5 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH5	10:8	rw	**Upper timer trigger threshold channel 5** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	7:5	r	**Reserved** Always read as 0
HYST_UP_CH4	4:3	rw	**Channel 4 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16

Field	Bits	Type	Description
CNT_UP_CH4	2:0	rw	**Upper timer trigger threshold channel 4** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements

Upper Counter Trigger Level Channel 6-9

The register is reset by RESET_TYPE_4.

CNT6_9_UPPER

Upper Counter Trigger Level Channel 6-9 **(AC_H)** **Reset Value: $1A1A\ 1911_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_UP_CH9		CNT_UP_CH9			RES			HYST_UP_CH8		CNT_UP_CH8		
r			rwp		rwp			r			rwp		rwp		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			HYST_UP_CH7		CNT_UP_CH7			RES			HYST_UP_CH6		CNT_UP_CH6		
r			rwp		rwp			r			rwp		rwp		

Field	Bits	Type	Description
RES	31:29	r	**Reserved** Always read as 0
HYST_UP_CH9	28:27	rwp	**Channel 9 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH9	26:24	rwp	**Upper timer trigger threshold channel 9** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	23:21	r	**Reserved** Always read as 0

Field	Bits	Type	Description
HYST_UP_CH8	20:19	rwp	**Channel 8 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH8	18:16	rwp	**Upper timer trigger threshold channel 8** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	15:13	r	**Reserved** Always read as 0
HYST_UP_CH7	12:11	rwp	**Channel 7 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH7	10:8	rwp	**Upper timer trigger threshold channel 7** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	7:5	r	**Reserved** Always read as 0
HYST_UP_CH6	4:3	rwp	**Channel 6 upper hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_UP_CH6	2:0	rwp	**Upper timer trigger threshold channel 6** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements

Lower Comparator Trigger Level Channel 0-3

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of 182F 423A$_H$.

TH0_3_LOWER

Lower Comparator Trigger Level Channel 0-3 (80$_H$) **Reset Value: 3C2C 3A42$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3								**CH2**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1								**CH0**							
rw								rw							

Field	Bits	Type	Description
CH3	31:24	rw	**Channel 3 lower trigger level, ADC2_CH3_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH2	23:16	rw	**Channel 2 lower trigger level, ADC2_CH2_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH1	15:8	rw	**Channel 1 lower trigger level, ADC2_CH1_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH0	7:0	rw	**Channel 0 lower trigger level, ADC2_CH0_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value

Lower Comparator Trigger Level Channel 4 & 5

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of 0000 9A00$_H$.

TH4_5_LOWER

Lower Comparator Trigger Level Channel 4&5 (84$_H$) **Reset Value: 0000 9A2C$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH5								CH4							
rw								rw							

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
CH5	15:8	rw	**Channel 5 lower trigger level, ADC2_CH5_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH4	7:0	rw	**Channel 4 lower trigger level, ADC2_CH4_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value

Lower Comparator Trigger Level Channel 6-9

The register is reset by RESET_TYPE_4.

This register is being initialized by the BootROM during the boot-up process to a value of C6D3 9ECD$_H$.

TH6_9_LOWER

Lower Comparator Trigger Level Channel 6-9 (88$_H$) **Reset Value: C7D3 BBDB$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH9								CH8							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH7								CH6							
rw								rw							

Field	Bits	Type	Description
CH9	31:24	rw	**Channel 9 lower trigger level, ADC2_CH9_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH8	23:16	rw	**Channel 8 lower trigger level, ADC2_CH8_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH7	15:8	rw	**Channel 7 lower trigger level, ADC2_CH7_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value
CH6	7:0	rw	**Channel 6 lower trigger level, ADC2_CH6_LOWER_TH** 00$_H$ **MINTH**, Min. threshold value **...** FF$_H$ **MAXTH**, Max. threshold value

Lower Counter Trigger Level Channel 0-3

The register is reset by RESET_TYPE_4.

CNT0_3_LOWER
Lower Counter Trigger Level Channel 0-3 **(98_H)** **Reset Value: $1213\ 1312_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_LO_CH3		CNT_LO_CH3			RES			HYST_LO_CH2		CNT_LO_CH2		
r			rw		rw			r			rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			HYST_LO_CH1		CNT_LO_CH1			RES			HYST_LO_CH0		CNT_LO_CH0		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
RES	31:29	r	**Reserved** Always read as 0
HYST_LO_CH3	28:27	rw	**Channel 3 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH3	26:24	rw	**Lower timer trigger threshold channel 3** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	23:21	r	**Reserved** Always read as 0
HYST_LO_CH2	20:19	rw	**Channel 2 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16

Field	Bits	Type	Description
CNT_LO_CH2	18:16	rw	**Lower timer trigger threshold channel 2** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	15:13	r	**Reserved** Always read as 0
HYST_LO_CH1	12:11	rw	**Channel 1 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH1	10:8	rw	**Lower timer trigger threshold channel 1** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	7:5	r	**Reserved** Always read as 0
HYST_LO_CH0	4:3	rw	**Channel 0 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH0	2:0	rw	**Lower timer trigger threshold channel 0** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements

Lower Counter Trigger Level Channel 4 & 5

The register is reset by RESET_TYPE_4.

CNT4_5_LOWER
Lower Counter Trigger Level Channel 4&5 ($9C_H$) Reset Value: 0000 $0A0A_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			HYST_LO_CH5		CNT_LO_CH5			RES			HYST_LO_CH4		CNT_LO_CH4		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
RES	31:13	r	**Reserved** Always read as 0
HYST_LO_CH5	12:11	rw	**Channel 5 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH5	10:8	rw	**Lower timer trigger threshold channel 5** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	7:5	r	**Reserved** Always read as 0
HYST_LO_CH4	4:3	rw	**Channel 4 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16

Field	Bits	Type	Description
CNT_LO_CH4	2:0	rw	**Lower timer trigger threshold channel 4** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements

Lower Counter Trigger Level Channel 6-9

The register is reset by RESET_TYPE_4.

CNT6_9_LOWER

Lower Counter Trigger Level Channel 6-9 **($A0_H$)** **Reset Value: 0A0A 0A0A$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_LO_CH9		CNT_LO_CH9			RES			HYST_LO_CH8		CNT_LO_CH8		
r			rwp		rwp			r			rwp		rwp		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			HYST_LO_CH7		CNT_LO_CH7			RES			HYST_LO_CH6		CNT_LO_CH6		
r			rwp		rwp			r			rwp		rwp		

Field	Bits	Type	Description
RES	31:29	r	**Reserved** Always read as 0
HYST_LO_CH9	28:27	rwp	**Channel 9 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH9	26:24	rwp	**Lower timer trigger threshold channel 9** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	23:21	r	**Reserved** Always read as 0

Field	Bits	Type	Description
HYST_LO_CH8	20:19	rwp	**Channel 8 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH8	18:16	rwp	**Lower timer trigger threshold channel 8** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	15:13	r	**Reserved** Always read as 0
HYST_LO_CH7	12:11	rwp	**Channel 7 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH7	10:8	rwp	**Lower timer trigger threshold channel 7** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements
RES	7:5	r	**Reserved** Always read as 0
HYST_LO_CH6	4:3	rwp	**Channel 6 lower hysteresis** 0_H **HYSTOFF**, hysteresis switched off 1_H **HYST4**, hysteresis = 4 2_H **HYST8**, hysteresis = 8 3_H **HYST16**, hysteresis = 16
CNT_LO_CH6	2:0	rwp	**Lower timer trigger threshold channel 6** 0_H **MEAS1**, 1 measurement 1_H **MEAS2**, 2 measurements 2_H **MEAS4**, 4 measurements 3_H **MEAS8**, 8 measurements 4_H **MEAS16**, 16 measurements 5_H **MEAS32**, 32 measurements 6_H **MEAS64**, 64 measurements 7_H **MEAS128**, 128 measurements

23.8 Start-up Behavior after Reset

After the end of a reset phase the measurement sources and the post-processing units need some time for settling. In order to avoid undesired triggering of interrupts until the measurement signal acquisition is in a steady state, the status signals are forced to zero during the start-up phase.

The end of the start-up phase is indicated by the ready signal MCM_RDY.

Measurement Core start-up procedure: the startup time of the complete signal chain is 2200 EoC cycles. The IIR-filter coefficient is set to a=2^-1 (fastest response time of the IIR-filter).

During the startup phase, the DPP will use SQ=11_1111_1111, regardless of the sequence registers configuration.

23.9 Post-processing Default Values

The following table shows the assigned measurements of the particular channels and the reset default values which are read from Firmware during power-up. Since the channels 6-9 of the unit are exclusively used for internal measurements, they can only be partly accessed by the application software.

Table 174 Grade 1 - Channel allocation and post-processing default settings (effective after reset)

Channel / MMODE [1]	Analog	Digital [2]	Hyste-resis [3]	IIR - Filter [4]	Counters [5]	Functional Description
Ch. 1 / 0_H **VS**	5.78 V	42_H	2_H (8)	2_H (8)	3_H (8)	Battery supply voltage input, lower
	17.3 V	$C5_H$	3_H (16)		3_H (8)	upper
Ch. 2 / 0_H **VSD**	5.78 V	$2F_H$	2_H (8)	2_H (8)	3_H (8)	VSD voltage, lower
	23.4 V	BD_H	2_H (8)		3_H (8)	VSD voltage, upper
Ch. 3 / 0_H **VCP**	5.2 V	18_H	2_H (8)	2_H (8)	2_H (4)	VCP voltage, lower
	35 V	AB_H	2_H (8)		2_H (4)	VCP voltage, uppper
Ch. 4/ 0_H **MON/VBAT_SENSE**	-	00_H	1_H (4)	2_H (8)	2_H (4)	MON voltage, lower
	-	00_H	2_H (8)		2_H (4)	MON voltage, upper
Ch. 5/ 0_H **VDDP**	4.5 V	$9A_H$	1_H (4)	2_H (8)	2_H (4)	+5V, Port supply voltage, lower
	5.5 V	BC_H	2_H (8)		2_H (4)	+5V, Port supply voltage, upper
Ch. 6/ 0_H **VAREF**	4.5 V	CD_H	1_H (4)	1_H (4)	2_H (4)	VAREF, lower
	5.5 V	FA_H	2_H (8)		1_H (2)	VAREF upper
Ch. 7/ 0_H **VBG**	1.01 V	$9E_H$	1_H (4)	1_H (4)	2_H (4)	VBG, lower
	1.3 V	CB_H	1_H (4)		1_H (2)	VBG, upper
Ch. 8/ 0_H **VDDC**	1.35V	$D3_H$	1_H (4)	2_H (8)	2_H (4)	Core supply voltage, lower
	1.6 V	FA_H	3_H (16)		2_H (4)	Core supply voltage, upper
Ch. 9/ 2_H **TEMP**	0.94 V	$C6_H$	1_H (4)	2_H (8)	2_H (4)	temperature sensor: lower hysteresis threshold value corresponding to approx. 120°C
	1.05 V	DF_H	3_H (16)		2_H (4)	over-temperature threshold corresponding to nominal 168°C. The hysteresis corresponds to approximately 32°C.

1) MMODE of each channel is defined by SFR reset values: 00_B range control, 01_B under-voltage mode, 10_B over-voltage mode. The measurement mode for CH6 - CH9 can not be programmed by the user.
2) register: THx_y_LOWER / THx_y_UPPER
3) register: CNTx_y_LOWER / CNTx_y_UPPER; selectable decimal values [0, 4, 8, 16]
4) register: FILTCOEFF0_5 / FILTCOEFF6_9
5) register: CNTx_y_UPPER.CNT_UP_CHx / CNTx_y_LOWER.CNT_LO_CHx

Table 175 Grade 0 - Channel allocation and post-processing default settings (effective after reset)

Channel / MMODE [1]	Analog	Digital [2]	Hyste-resis [3]	IIR - Filter [4]	Counters [5]	Functional Description
Ch. 1 / 0_H **VS**	5.78 V	42_H	2_H (8)	2_H (8)	3_H (8)	Battery supply voltage input, lower
	17.3 V	$C5_H$	3_H (16)		3_H (8)	upper
Ch. 2 / 0_H **VSD**	5.78 V	$2F_H$	2_H (8)	2_H (8)	3_H (8)	VSD voltage, lower
	23.4 V	BD_H	2_H (8)		3_H (8)	VSD voltage, upper
Ch. 3 / 0_H **VCP**	5.2 V	18_H	2_H (8)	2_H (8)	2_H (4)	VCP voltage, lower
	35 V	AB_H	2_H (8)		2_H (4)	VCP voltage, uppper
Ch. 4/ 0_H **MON**	-	00_H	1_H (4)	2_H (8)	2_H (4)	MON voltage, lower
	-	00_H	2_H (8)		2_H (4)	MON voltage, upper
Ch. 5/ 0_H **VDDP**	4.5 V	$9A_H$	1_H (4)	2_H (8)	2_H (4)	+5V, Port supply voltage, lower
	5.5 V	BC_H	2_H (8)		2_H (4)	+5V, Port supply voltage, upper
Ch. 6/ 0_H **VAREF**	4.5 V	CD_H	1_H (4)	1_H (4)	2_H (4)	VAREF, lower
	5.5 V	FA_H	2_H (8)		1_H (2)	VAREF upper
Ch. 7/ 0_H **VBG**	1.01 V	$9E_H$	1_H (4)	1_H (4)	2_H (4)	VBG, lower
	1.3 V	$E1_H$	1_H (4)		1_H (2)	VBG, upper
Ch. 8/ 0_H **VDDC**	1.35V	$D3_H$	1_H (4)	2_H (8)	2_H (4)	Core supply voltage, lower
	1.6 V	FA_H	3_H (16)		2_H (4)	Core supply voltage, upper
Ch. 9/ 2_H **TEMP**	0.94 V	$C6_H$	1_H (4)	2_H (8)	2_H (4)	temperature sensor: lower hysteresis threshold value corresponding to approx. 120°C
	1.14 V	$F0_H$	3_H (16)		2_H (4)	over-temperature threshold corresponding to nominal 203°C. The hysteresis corresponds to approximately 32°C.

1) MMODE of each channel is defined by SFR reset values: 00_B range control, 01_B under-voltage mode, 10_B over-voltage mode. The measurement mode for CH6 - CH9 can not be programmed by the user.
2) register: THx_y_LOWER / THx_y_UPPER
3) register: CNTx_y_LOWER / CNTx_y_UPPER; selectable decimal values [0, 4, 8, 16]
4) register: FILTCOEFF0_5 / FILTCOEFF6_9
5) register: CNTx_y_UPPER.CNT_UP_CHx / CNTx_y_LOWER.CNT_LO_CHx

24 10-bit analog-to-digital converter (ADC1)

24.1 Features

The principal features of the ADC1 are:

- Up to 8 analog input channels.
- Flexible results handling
  - 10-bit resolution.
- Flexible source selection due to sequencer:
  - Insert one exceptional sequence (ESM).
  - Insert an interrupt measurement into the current sequence (EIM).
  - Software mode.
- Conversion sample time (separate for each channel) adjustable to adapt to sensors and reference.
- Standard external reference (VAREF) to support ratiometric measurements and different signal scales.
- DMA support, transfer ADC conversion results via DMA into RAM.
- Support of suspended and power-saving modes.
- Result data protection for slow CPU access (Wait-for-Read mode).
- Programmable clock divider.
- Integrated sample and hold circuitry.

24.2 Introduction

The TLE987x includes a high-performance 10-bit analog-to-digital converter (ADC1) with eight multiplexed analog input channels. The ADC1 uses a successive approximation technique to convert the analog voltage levels from up to eight different sources. The analog input channels of the ADC1 are available at AN0, AN2 to AN5.

24.2.1 Block diagram

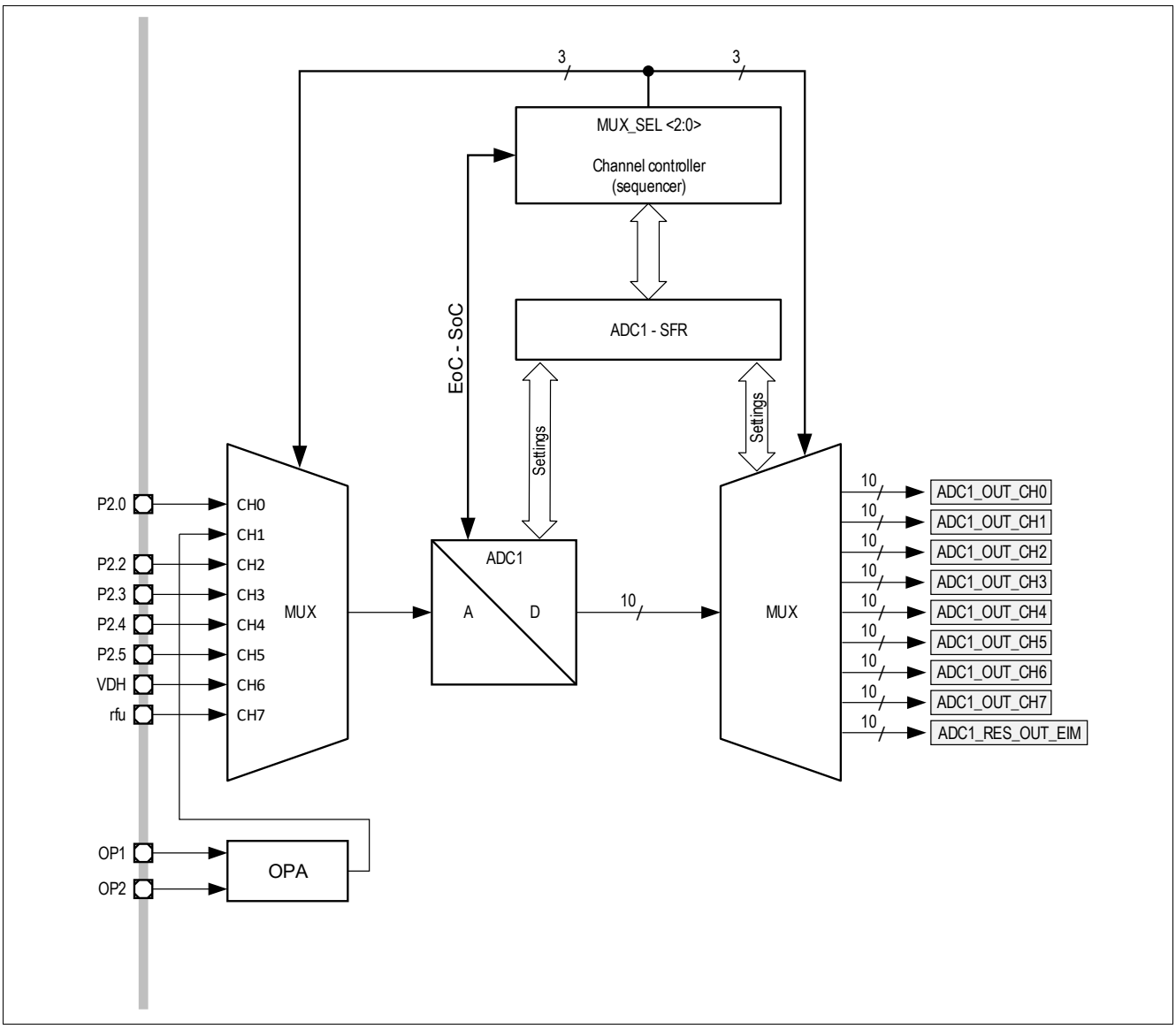


Figure 208 ADC1 top-level block diagram

As shown in the figure above, the ADC1 postprocessing block consists of a channel controller (Sequencer) and an 8-channel demultiplexer. The channel control block controls the multiplexer sequencing on the analog side, before the ADC1, and in the digital domain, behind the ADC1. The channel sequence can be controlled in a flexible way, which allows a certain degree of channel prioritization.

This capability can be used, e.g., to give supply voltage channels a higher priority than the other channel measurements.

24.2.2 ADC1 Modes Overview

The channel controller (Sequencer) runs in one of the following modes:

“Normal Sequencer Mode”, channels are selected out of 8 Sequence registers which contain individual enables for each of the 8 channels (SQ_FB.SQ_RUN = 1).

“Exceptional Interrupt Measurement”, upon a hardware event, a high priority channel is inserted into the current sequence.

“Exceptional Sequence Measurement”, upon a hardware event, a complete sequence is inserted into the current sequence. The current sequence (up to 8 measurements) is interrupted before the ESM sequence is inserted.

“Suspend Mode”: Exceptional Sequencer and Interrupt Measurement can be triggered, (all sequence register loaded with “00” and SQ_FB.SQ_RUN = 1, measurements by software can not be triggered.

“Debug Suspend Mode”: The corresponding bit MODSUSP2.ADC1_SUSP is set.

“Software Mode”, Exceptional Sequencer and Interrupt Measurement are ignored, each measurement is triggered by software, SQ_FB.SQ_RUN = 0

24.3 ADC1 - Core (10-Bit ADC)

24.3.1 Functional Description

In order to enable the ADC1 the following registers need to be set in order to enable the ADC1

- PMCON1.ADC1DIS = 0
- GLOBCTR.ANON = 11
- ADC1->CTRL_STS.PD_N = 1
- program sequencer (via ADC1->SQ_1_4 and ADC1->SQ_5_8)
- start sequencer ADC1->SQ_FB.SQ_RUN = 1

The different sequencer modes are controlled by SFR Register:

- “Normal Sequencer Mode” described in the Chapter **Channel Controller**.
- “Exceptional Interrupt Measurement” (EIM), upon hardware event, the channel programmed in **CHx_EIM** is inserted immediately. The current measurement is aborted, then the EIM measurement is inserted, it will start two tADCI cycles after the trigger edge. After the EIM is done the aborted measurement is repeated and the sequence continues, see **Figure 209**. An EIM can even interrupt an ESM. The EIM trigger is level sensitive, as long as the EIM trigger input is high level the ADC1 performs EIM measurement(s). The number of EIMs performed depends on the setting of the **CHx_EIM**.REP flag.
- “Exceptional Sequence Measurement“ (ESM), upon hardware event, the sequence programmed in **CHx_ESM** is inserted in the currently running sequence. The current measurement is aborted, then the ESM sequence will be executed, two tADCI cycles after the trigger. After the ESM is finished the aborted measurement will be repeated and the sequence continues, see **Figure 210**. After the Exceptional Sequence Measurement is finished an interrupt can be requested. See **SQ1_4** and following sequence registers.
- “Software Mode”, in Software Mode the control of the Channel Controller (Sequencer) is disabled, instead the conversions are fully controlled by software. During Software Mode EIM and ESM hardware events are ignored. See **SQ1_4** and following sequence registers.

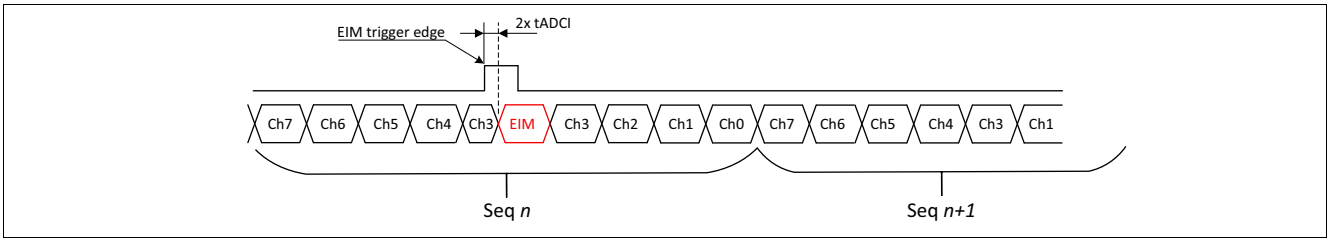


Figure 209 ADC1 EIM measurement within sequence example

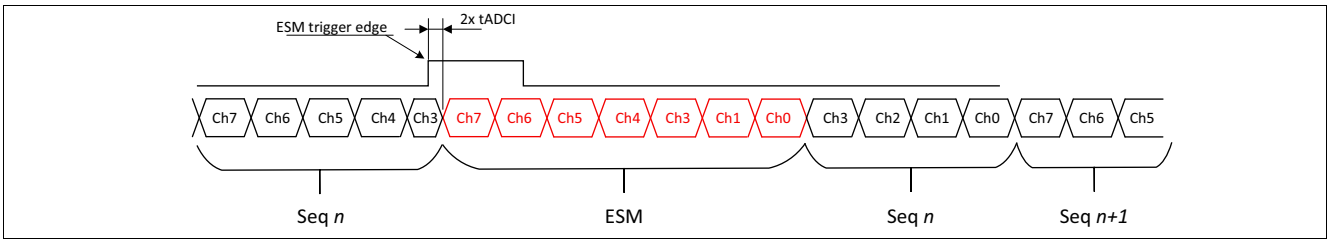


Figure 210 ADC1 ESM measurement within sequence example

The default mode after reset of the ADC1 is Software Mode.

In case EIM and ESM hardware events occur at the same time, the EIM event is processed first.

While an ESM sequence is running, the sequence can be interrupted by one or more EIMs.

The end of ESM and EIM is flagged via interrupt. In case of repeat count of EIM, the EIM interrupt is generated at the end of EIM sequence when the last EIM measurement is done.

Software Mode:

- Software mode is entered
  - by disabling the Sequencer ADC1.**SQ_FB**.SQ_RUN to zero. The Software mode is entered when the current measurement is finished.
  In software mode, the conversion are controlled via **CTRL_STS**.
- The Software Mode is left
  - by enabling the Sequencer ADC1.**SQ_FB**.SQ_RUN to one. If the Sequencer is enabled, the Sequencer always starts with SQ1.

In Software Mode measurements are triggered by writing the **CTRL_STS**.SOC bit. This bit is active as long as the conversion is in progress. The user polls the **CTRL_STS**.EOC bit. Once this bit is ‘1’ the conversion is finished and the EOC bit is cleared on read (rh). After the EOC bit is cleared a new conversion can be started **CTRL_STS**.SOC.

In Software Mode DMA requests and interrupts are generated.

Suspend Mode:

Suspend Mode is entered if all Sequences are programmed to zero but the Sequencer is still enabled, **SQ_FB**.ASQ_RUN is set to one. Exceptional Sequencer and Interrupt Measurement can be triggered, measurements by software can not be triggered (via **CTRL_STS**.SOC).

In Suspend Mode DMA requests and interrupts (for ESM and EIM) are generated.

Debug Suspend Mode:

During Debug Suspend Mode the Sequencer is stopped once the current measurement is finished (after the next EOC event). As long as the Debug Suspend Mode is active no measurements are performed by the Sequencer. EIM and ESM are ignored during Debug Suspend Mode.

During Debug Suspend Mode, conversions can be triggered via **CTRL_STS**.SOC bit in Software Mode.

24.3.2 ADC1 Control and Status Registers

Table 176 shows the module base address.

Table 176 Register Address Space

Module	Base Address	End Address	Note
ADC1	40004000_H	$40007FFF_H$	ADC-SAR10B/DPP

Table 177 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
ADC1 Control and Status Registers			
CTRL_STS	ADC1 Control and Status Register	00_H	$0000\ 0000_H$
GLOBCTR	Global Control Register	04_H	$0000\ 0000_H$
GLOBSTR	Global Status Register	74_H	$0000\ 0000_H$

The registers are addressed wordwise.

Global Status Register

The register is reset by RESET_TYPE_3.

GLOBSTR
Global Status Register **(74_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						ANON_ST		RES		CHNR			RES	SAMPLE	BUSY
r						r		r		r			r	r	r

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
RES	15:10	r	**Reserved** Always read as 0
ANON_ST	9:8	r	**Analog Part Switched On** 00_B **OFF**, ADC1 switched off 01_B **S_STANDBY**, Slow standby mode 10_B **F_STANDBY**, Fast standby mode 11_B **NORMAL** , Normal Operation
RES	7:6	r	**Reserved** Always read as 0
CHNR	5:3	r	**Channel Number** Indicates the current or last converted analog input channel. This bitfield is updated when a conversion is started.
RES	2	r	**Reserved** Always read as 0
SAMPLE	1	r	**Sample Phase Indication** 0_B **IDLE**, ADC1 is idle or converting 1_B **ACTIVE**, The Input signal is being sampled
BUSY	0	r	**Analog Part Busy** 0_B **IDLE**, ADC1 idle 1_B **ACTIVE**, ADC1 Conversion is currently running

Global Control Register

The register is reset by RESET_TYPE_3.

GLOBCTR
Global Control Register **(04_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	RES					ANON		RES		DIVA					
r	r					rw		r		rw					

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
RES	15	r	**Reserved** Always read as 0
RES	14:10	r	**Reserved** Always read as 0
ANON	9:8	rw	**Analog Part Switched On** 00_B **OFF**, ADC1 switched off 01_B **S_STANDBY**, Slow standby mode 10_B **F_STANDBY**, Fast standby mode 11_B **NORMAL** , Normal Operation
RES	7:6	r	**Reserved** Always read as 0
DIVA	5:0	rw	**Divide Factor for the Analog internal clock** Defines the frequency of the basic converter clock f_{ADCI} (base clock for conversion and sample phase). *Note:* *f_{ADC} is derived from f_{SYS}* 00_H **DIV1** , $f_{ADCI} = f_{ADC}$ 01_H **DIV2**, $f_{ADCI} = f_{ADC}/2$ 02_H **DIV3**, $f_{ADCI} = f_{ADC}/3$ 03_H , ... $3F_H$ **DIV64**, $f_{ADCI} = f_{ADC}/64$

ADC1 Control and Status Register

The register is reset by RESET_TYPE_3.

CTRL_STS
ADC1 Control and Status Register **(00_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES									IN_MUX_SEL			EOC	SOC	RES	PD_N
r									rw			rh	rwh1	r	rw

Field	Bits	Type	Description
RES	31:7	r	**Reserved** Always read as 0
IN_MUX_SEL	6:4	rw	**Channel for software mode** 000_B **CH0_EN**, Channel 0 enable 001_B **CH1_EN**, Channel 1 enable 010_B **CH2_EN**, Channel 2 enable 011_B **CH3_EN**, Channel 3 enable 100_B **CH4_EN**, Channel 4 enable 101_B **CH5_EN**, Channel 5 enable 110_B **CH6_EN**, Channel 6 enable 111_B **CH7_EN**, Channel 7 enable
EOC	3	rh	**ADC1 End of Conversion (software mode)** 0_B **Pending**, conversion still running 1_B **Finished**, conversion has finished
SOC	2	rwh1	**ADC1 Start of Conversion (software mode)** *Note: Bit is set by software to start conversion and it is cleared by hardware once the conversion is finished. ADC2_SOC can be only written if the DPP is in software mode.* 0_B **Disable**, no conversion is started 1_B **Enable**, conversion is started
RES	1	r	**Reserved** Always read as 0
PD_N	0	rw	**ADC1 Power Down Signal** 0_B **POWER DOWN**, ADC1 is powered down 1_B **ACTIVE**, ADC1 is switched on

24.4 Channel Controller

24.4.1 Functional Description

The task of each channel controller is a prioritization of the individual measurement channels. The sequencing scheme is illustrated in the example of following table and can be programmed individually for measurement unit.

Table 178 Measurement Channel Sequence (Definition Example)

Measurement channel n	MSB CH7	CH6	CH5	CH4	CH3	CH2	CH1	LSB CH0
$SQ_1[7:0]$	1	1	1	1	0	1	1	1
$SQ_2[7:0]$	1	1	1	0	1	0	0	0
$SQ_3[7:0]$	1	1	1	1	0	1	1	0
$SQ_4[7:0]$	1	1	1	0	1	0	0	1
$SQ_5[7:0]$	1	1	1	1	0	1	1	0
$SQ_6[7:0]$	1	1	1	0	1	0	0	0
$SQ_7[7:0]$	1	1	1	1	0	1	1	1
$SQ_8[7:0]$	1	1	1	0	1	0	0	0

The sequence registers SQ_n define the time sequence of the measurement channels by the following rules:

- The sequence registers define the measurement sequence and are evaluated from register 1 to 8 and for each bit from MSB to LSB, which defines a max. overall measurement count of 64 sampling and conversion cycles.
- If the individual bit in the sequence register is set to ’1’, the corresponding channel is measured.
- If the individual bit in the sequence register is set to ’0’, the corresponding channel is skipped.

In the upper example, the resulting channel sequence is defined as:

CH7, CH6, CH5, CH4, CH2, CH1, CH0, CH7, CH6, CH5, CH3,....

The following equations can be used to calculate the periodicity of the required channel measurement.

The overall measurement periodicity of all measurements in A/D conversion cycles is defined as:

$$\overline{N_{meas}} = \sum_{m=1}^{8}\left(\sum_{n=1}^{8} SQ_m[n]\right) \tag{24.1}$$

The average measurement periodicity of channel n in A/D conversion cycles is defined as

$$\overline{N_{meas,\,n}} = \frac{\left(\sum_{m=1}^{8} SQ_m[n]\right)}{\overline{T_{meas}}} \tag{24.2}$$

Once a channel is selected by the sequence, the corresponding

- Sample Time t_{SAMPLE} = (2 + STC) / f_{ADCI}, **STC_0_3** and **STC_4_7**

is selected.

Conversion Time

Note: f_{ADC} is derived from f_{SYS}.

The total time required for a conversion depends on several user-definable factors:

- The ADC conversion clock frequency, where $f_{ADCI} = f_{ADC}$ / (DIVA+1).
- The selected sample time, where $t_{sample} = (2 + STC) \times t_{ADCI}$; (STC = additional sample time defined in **STC_0_3** and **STC_4_7**)
- The selected result width N (8/10 bits), defined in **DWSEL**, $t_{conv_step} = (1 + N) \times t_{ADCI}$
- Synchronization steps done at module clock speed, $t_{sync} = 2 \times t_{ADC}$

The conversion time is the sum of sample time, conversion steps, and synchronization. It can be computed with the following formula:

$$t_{conv} = t_{sample} + t_{conv_step} + t_{sync} = (3 + STC + N) \times t_{ADCI} + 2 \times t_{ADC} \quad (24.3)$$

Minimum conversion time: t_{CN} = [3+0+8(result width)] * 1 (min DIVA) + 2 = 13 clock cycles

24.4.2 Channel Controller Control Registers

Table 179 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Channel Controller Control Registers			
CHx_EIM	Channel Setting Bits for Exceptional Interrupt Measurement	08_H	$0000\ 0000_H$
CHx_ESM	Channel Setting Bits for Exceptional Sequence Measurement	$0C_H$	$0000\ 0000_H$
SQ1_4	Measurement Channel Enable Bits for Cycle 1-4	18_H	$0000\ 0000_H$
SQ5_8	Measurement Channel Enable Bits for Cycle 5-8	$1C_H$	$0000\ 0000_H$
DWSEL	Measurement Channel Data Width Selection	24_H	$0000\ 0000_H$
STC_0_3	Measurement Channel Sample Time Control 0-3	28_H	$0000\ 0000_H$
STC_4_7	Measurement Channel Sample Time Control 4-7	$2C_H$	$0000\ 0000_H$
SQ_FB	Sequencer Feedback Register	50_H	$0000\ 0000_H$

The registers are addressed wordwise.

Measurement Channel Enable Bits for Cycle 1-4

The register is reset by RESET_TYPE_4.

SQ1_4

Measurement Channel Enable Bits for Cycle 1-4 (18_H) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SQ4								**SQ3**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ2								**SQ1**							
rw								rw							

Field	Bits	Type	Description
SQ4	31:24	rw	**Sequence 4 channel enable** The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable

Field	Bits	Type	Description
SQ3	23:16	rw	**Sequence 3 channel enable** The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable
SQ2	15:8	rw	**Sequence 2 channel enable** The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable
SQ1	7:0	rw	**Sequence 1 channel enable** The following values can be or'ed: *Note: SQ1-SQ4 should be only written is sequencer is disabled /Software Mode enabled SQ_RUN = 0. SQ1-SQ4 should not be written with "0000 0000".* $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable

Measurement Channel Enable Bits for Cycle 5-8

The register is reset by RESET_TYPE_4.

SQ5_8
Measurement Channel Enable Bits for Cycle 5-8 ($1C_H$) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SQ8								SQ7							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ6								SQ5							
rw								rw							

Field	Bits	Type	Description
SQ8	31:24	rw	**Sequence 8 channel enable** The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable
SQ7	23:16	rw	**Sequence 7 channel enable** The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable
SQ6	15:8	rw	**Sequence 6 channel enable** The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable

Field	Bits	Type	Description
SQ5	7:0	rw	**Sequence 5 channel enable** The following values can be or'ed: *Note: SQ5-SQ8 should be only written is sequencer is disabled /Software Mode enabled SQ_RUN = 0.* *SQ5-SQ8 should not be written with "0000 0000"* $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable

Measurement Channel Data Width Selection

The register is reset by RESET_TYPE_4.

DWSEL
Measurement Channel Data Width Selection **(24_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								ch7	ch6	ch5	ch4	ch3	ch2	ch1	ch0
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:8	r	**Reserved** Always read as 0
ch7	7	rw	**Data Width channel 7** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch6	6	rw	**Data Width channel 6** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch5	5	rw	**Data Width channel 5** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch4	4	rw	**Data Width channel 4** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch3	3	rw	**Data Width channel 3** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch2	2	rw	**Data Width channel 2** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch1	1	rw	**Data Width channel 1** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)
ch0	0	rw	**Data Width channel 0** 0_B **10BIT**, The result is 10 bits wide (bits 11 .. 2) 1_B **8BIT**, The result is 8 bits wide (bits 11 .. 4)

Measurement Channel Sample Time Control 0-3

The register is reset by RESET_TYPE_4.

STC_0_3

Measurement Channel Sample Time Control 0-3 (28_H) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ch3								ch2							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ch1								ch0							
rw								rw							

Field	Bits	Type	Description
ch3	31:24	rw	**Sample Time Control for Channel 3** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$
ch2	23:16	rw	**Sample Time Control for Channel 2** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$
ch1	15:8	rw	**Sample Time Control for Channel 1** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$
ch0	7:0	rw	**Sample Time Control for Channel 0** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$

Measurement Channel Sample Time Control 4-7

The register is reset by RESET_TYPE_4.

STC_4_7
Measurement Channel Sample Time Control 4-7 ($2C_H$) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ch7								ch6							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ch5								ch4							
rw								rw							

Field	Bits	Type	Description
ch7	31:24	rw	**Sample Time Control for Channel 7** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$
ch6	23:16	rw	**Sample Time Control for Channel 6** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$
ch5	15:8	rw	**Sample Time Control for Channel 5** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$
ch4	7:0	rw	**Sample Time Control for Channel 4** Number of additional clock cycles to be added to the minimum sample phase of 2 analog clock cycles: $t_{SAMPLE} = (2 + STC) / f_{ADCI}$

Sequencer Feedback Register

The register is reset by RESET_TYPE_3.

SQ_FB
Sequencer Feedback Register **(50_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													CHx		
r													r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		SQx			ESM_ACTIVE	EIM_ACTIVE	SQ_RUN	RES							
r		r			r	r	rw	r							

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
CHx	18:16	r	**Current Channel** 000_B **CH0**, Channel 0 001_B **CH1**, Channel 1 010_B **CH2**, Channel 2 011_B **CH3**, Channel 3 100_B **CH4**, Channel 4 101_B **CH5**, Channel 5 110_B **CH6**, Channel 6 111_B **CH7**, Channel 7
RES	15:14	r	**Reserved** Always read as 0
SQx	13:11	r	**Current Active Sequence in Sequencer Mode** 000_B **SQ0**, Sequence 1 001_B **SQ1**, Sequence 2 010_B **SQ2**, Sequence 3 011_B **SQ3**, Sequence 4 100_B **SQ4**, Sequence 5 101_B **SQ5**, Sequence 6 110_B **SQ6**, Sequence 7 111_B **SQ7**, Sequence 8
ESM_ACTIVE	10	r	**ADC1 ESM active** 0_B **not active**, ESM not active 1_B **active**, ESM active
EIM_ACTIVE	9	r	**ADC1 EIM active** 0_B **not active**, EIM not active 1_B **active**, EIM active

Field	Bits	Type	Description
SQ_RUN	8	rw	**ADC1 Sequencer RUN** 0_B **SQ Stopped**, Post-processing Sequencer in stopped / Software mode 1_B **SQ Running**, Post-processing Sequencer is running
RES	7:0	r	**Reserved** Always read as 0

Channel Setting for Exceptional Interrupt Measurement

The register is reset by RESET_TYPE_3.

CHx_EIM

Channel Setting Bits for Exceptional Interrupt Measurement (08_H) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													TRIG_SEL		
r													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES									REP			RES	CHx		
r									rw			r	rw		

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
TRIG_SEL	18:16	rw	**Trigger selection for exceptional interrupt measurement (EIM)** 000_B **NONE**, 001_B **COUT63**, CCU6 Channel3 010_B **GPT12_T6OUT**, GPT12E.T6 toggle latch output 011_B **GPT12_T3OUT**, GPT12E.T3 toggle latch output 100_B **T2**, t2_adc_trigger 101_B **T21**, t21_adc_trigger 110_B **CCU_6_INT**, ccu6_int, Timer3 output 111_B **RES**, reserved
RES	15:7	r	**Reserved** Always read as 0
REP	6:4	rw	**Repeat count for exceptional interrupt measurement (EIM)** 000_B **n**, performs EIM as long as trigger signal is high level 001_B **2**, Measurements 010_B **4**, Measurements 011_B **8**, Measurements 100_B **16**, Measurements 101_B **32**, Measurements 110_B **64**, Measurements 111_B **128**, Measurements

Field	Bits	Type	Description
RES	3	r	**Reserved** Always read as 0
CHx	2:0	rw	**Channel set for exceptional interrupt measurement (EIM)** 000_B **CH0_EN**, Channel 0 enable 001_B **CH1_EN**, Channel 1 enable 010_B **CH2_EN**, Channel 2 enable 011_B **CH3_EN**, Channel 3 enable 100_B **CH4_EN**, Channel 4 enable 101_B **CH5_EN**, Channel 5 enable 110_B **CH6_EN**, Channel 6 enable 111_B **CH7_EN**, Channel 7 enable

Channel Setting for Exceptional Sequence Measurement

The register is reset by RESET_TYPE_3.

CHx_ESM

Channel Setting Bits for Exceptional Sequence Measurement ($0C_H$) **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													TRIG_SEL		
r													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								ESM_0							
r								rw							

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
TRIG_SEL	18:16	rw	**Trigger selection for exceptional interrupt measurement (ESM)** 000_B **NONE**, 001_B **COUT63**, CCU6 Channel3 010_B **GPT12_T6OUT**, GPT12E.T6 toggle latch output 011_B **GPT12_T3OUT**, GPT12E.T3 toggle latch output 100_B **T2**, t2_adc_trigger 101_B **T21**, t21_adc_trigger 110_B **CCU_6_INT**, ccu6_int, Timer3 output 111_B **RES**, reserved
RES	15:8	r	**Reserved** Always read as 0
ESM_0	7:0	rw	**Channel Sequence for Exceptional Sequence Measurement (ESM)** *Note: Update to ESM sequence should be written while trigger selection (ADC1_ESM_TRIG_SEL) is set to none.* The following values can be or'ed: $0000\ 0001_B$**CH0_EN**, Channel 0 enable $0000\ 0010_B$**CH1_EN**, Channel 1 enable $0000\ 0100_B$**CH2_EN**, Channel 2 enable $0000\ 1000_B$**CH3_EN**, Channel 3 enable $0001\ 0000_B$**CH4_EN**, Channel 4 enable $0010\ 0000_B$**CH5_EN**, Channel 5 enable $0100\ 0000_B$**CH6_EN**, Channel 6 enable $1000\ 0000_B$**CH7_EN**, Channel 7 enable

24.5 Conversion Result Handling

24.5.1 Functional Description

Each Result Register **RES_OUT0** - **RES_OUT7** has a valid flag and an optional wait-for-read mode (WFR) configuration bit.

The valid flag indicates if the corresponding result register contains valid data. The valid bit is set by hardware once the ADC conversion is ready and stored in the corresponding result register. It is cleared once the result is read by software (or DMA request).

Wait-for-read mode

The wait-for-read mode is a feature to prevent data loss due to overwriting a result register with a new conversion result before the CPU (or DMA transfer) has read the previous data.

Wait-for-read mode prevents overwriting of result register. The conversion is always started, however the result is only written to the result register in case the valid bit is set to zero. In case the valid bit is set to one, the result of the ADC is ignored.

24.5.2 Result Registers

In Software Mode, the WFR configuration bit is ignored. In software mode the result register is always updated, but VF is still set and OF retains its value.

Table 180 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Result Registers			
RES_OUT_EIM	ADC1 Output Channel EIM	40_H	$0000\ 0XXX_H$
RES_OUT7	ADC1 Output Channel 7	54_H	$0000\ 0XXX_H$
RES_OUT6	ADC1 Output Channel 6	58_H	$0000\ 0XXX_H$
RES_OUT5	ADC1 Output Channel 5	$5C_H$	$0000\ 0XXX_H$
RES_OUT4	ADC1 Output Channel 4	60_H	$0000\ 0XXX_H$
RES_OUT3	ADC1 Output Channel 3	64_H	$0000\ 0XXX_H$
RES_OUT2	ADC1 Output Channel 2	68_H	$0000\ 0XXX_H$
RES_OUT1	ADC1 Output Channel 1	$6C_H$	$0000\ 0XXX_H$
RES_OUT0	ADC1 Output Channel 0	70_H	$0000\ 0XXX_H$

The registers are addressed wordwise.

ADC1 Output Channel 0

The register is reset by RESET_TYPE_3.

RES_OUT0
ADC1 Output Channel 0 **(70_H)** **Reset Value: $0000\ 0XXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF0	VF0	WFR0
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				OUT_CH0											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF0	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF0	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH0 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read
WFR0	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0

Field	Bits	Type	Description
OUT_CH0	11:0	r	**Output reset value Channel 0** 8 bit conversion: OUT_CH0[11:4] = result[7:0], the four LSB OUT_CH0[3:0] are padded with “0000” 10 bit conversion: OUT_CH0[11:2] = result[9:0], the two LSB OUT_CH0[1:0] are padded with “00”

ADC1 Output Channel 1

The register is reset by RESET_TYPE_3.

RES_OUT1
ADC1 Output Channel 1 **($6C_H$)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													**OF1**	**VF1**	**WFR1**
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**OUT_CH1**											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF1	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF1	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH1 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is reads* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read

Field	Bits	Type	Description
WFR1	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0
OUT_CH1	11:0	r	**ADC1 Output result value Channel 1** 8 bit conversion: OUT_CH1[11:4] = result[7:0], the four LSB OUT_CH1[3:0] are padded with “0000” 10 bit conversion: OUT_CH1[11:2] = result[9:0], the two LSB OUT_CH1[1:0] are padded with “00”

ADC1 Output Channel 2

The register is reset by RESET_TYPE_3.

RES_OUT2
ADC1 Output Channel 2 **(68_H)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF2	VF2	WFR2
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				OUT_CH2											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF2	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten

Field	Bits	Type	Description
VF2	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH2 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read
WFR2	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0
OUT_CH2	11:0	r	**ADC1 Output result value Channel 2** 8 bit conversion: OUT_CH2[11:4] = result[7:0], the four LSB OUT_CH2[3:0] are padded with “0000” 10 bit conversion: OUT_CH2[11:2] = result[9:0], the two LSB OUT_CH2[1:0] are padded with “00”

ADC1 Output Channel 3

The register is reset by RESET_TYPE_3.

RES_OUT3

ADC1 Output Channel 3 **(64_H)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													**OF3**	**VF3**	**WFR3**
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**OUT_CH3**											
r				r											

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0

Field	Bits	Type	Description
OF3	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF3	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH3 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read
WFR3	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0
OUT_CH3	11:0	r	**ADC1 Output result value Channel 3** 8 bit conversion: OUT_CH3[11:4] = result[7:0], the four LSB OUT_CH3[3:0] are padded with “0000” 10 bit conversion: OUT_CH3[11:2] = result[9:0], the two LSB OUT_CH3[1:0] are padded with “00”

ADC1 Output Channel 4

The register is reset by RESET_TYPE_3.

RES_OUT4
ADC1 Output Channel 4 **(60_H)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF4	VF4	WFR4
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				OUT_CH4											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF4	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF4	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH4 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read
WFR4	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0

Field	Bits	Type	Description
OUT_CH4	11:0	r	**ADC1 Output result value Channel 4** 8 bit conversion: OUT_CH4[11:4] = result[7:0], the four LSB OUT_CH4[3:0] are padded with “0000” 10 bit conversion: OUT_CH4[11:2] = result[9:0], the two LSB OUT_CH4[1:0] are padded with “00”

ADC1 Output Channel 5

The register is reset by RESET_TYPE_3.

RES_OUT5
ADC1 Output Channel 5 **($5C_H$)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													**OF5**	**VF5**	**WFR5**
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**OUT_CH5**											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF5	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF5	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH5 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read

Field	Bits	Type	Description
WFR5	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0
OUT_CH5	11:0	r	**ADC1 Output result value Channel 5** 8 bit conversion: OUT_CH5[11:4] = result[7:0], the four LSB OUT_CH5[3:0] are padded with “0000” 10 bit conversion: OUT_CH5[11:2] = result[9:0], the two LSB OUT_CH5[1:0] are padded with “00”

ADC1 Output Channel 6

The register is reset by RESET_TYPE_3.

RES_OUT6
ADC1 Output Channel 6 **(58_H)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF6	VF6	WFR6
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				OUT_CH6											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF6	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten

Field	Bits	Type	Description
VF6	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH6 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read
WFR6	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0
OUT_CH6	11:0	r	**ADC1 Output result value Channel 6** 8 bit conversion: OUT_CH6[11:4] = result[7:0], the four LSB OUT_CH6[3:0] are padded with “0000” 10 bit conversion: OUT_CH6[11:2] = result[9:0], the two LSB OUT_CH6[1:0] are padded with “00”

ADC1 Output Channel 7

The register is reset by RESET_TYPE_3.

RES_OUT7
ADC1 Output Channel 7 **(54_H)** **Reset Value: $0000\ 0XXX_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF7	VF7	WFR7
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				OUT_CH7											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF7	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF7	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH7 *Note: Bit is set by hardware on update of result register and it is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read
WFR7	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0

Field	Bits	Type	Description
OUT_CH7	11:0	r	**ADC1 Output result value Channel 7** 8 bit conversion: OUT_CH7[11:4] = result[7:0], the four LSB OUT_CH7[3:0] are padded with “0000” 10 bit conversion: OUT_CH7[11:2] = result[9:0], the two LSB OUT_CH7[1:0] are padded with “00”

ADC1 Output EIM Channel

The register is reset by RESET_TYPE_3.

RES_OUT_EIM

ADC1 Output Channel EIM **(40_H)** **Reset Value: 0000 0XXX$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													**OF8**	**VF8**	**WFR8**
						r							r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**OUT_CH_EIM**											
		r								r					

Field	Bits	Type	Description
RES	31:19	r	**Reserved** Always read as 0
OF8	18	r	**Overrun Flag** Indicates if the result register is overwritten with new content (bit is set if VFx = 1 and new result is updated by hardware). *Note: Only set in WFRx = DISABLE and no software mode, clear on read of result register* 0_B **NO OVERRUN**, Result register not overwritten 1_B **OVERRUN**, Result register overwritten
VF8	17	rh	**Valid Flag** Indicates valid contents in result register bit field OUT_CH_EIM *Note: Bit is set by hardware on update of result register and if all repeat counts have been processed. It is cleared by software once the result register is read* 0_B **NOT VALID**, No new valid data available 1_B **VALID**, Result register contains valid data and has not yet been read

Field	Bits	Type	Description
WFR8	16	rw	**Wait-for-Read Mode** Enables wait-for-read mode for result register 0_B **DISABLE**, overwrite mode 1_B **ENABLE**, wait-for-read mode enabled
RES	15:12	r	**Reserved** Always read as 0
OUT_CH_EIM	11:0	r	**ADC1 output result value EIM** 8 bit conversion: OUT_CH_EIM[11:4] = result[7:0], the four LSB OUT_CH_EIM[3:0] are padded with “0000” 10 bit conversion: OUT_CH_EIM[11:2] = result[9:0], the two LSB OUT_CH_EIM[1:0] are padded with “00”

24.6 DMA Requests

Note: DMA is not available in all products of the TLE987x family. See the datasheet for details..

The DMA Controller implements the following hardware DMA requests:

ADC1 one sequence done, burst request

ADC1 exceptional sequence (ESM) done, burst request

ADC1 channel 0 conversion done

ADC1 channel 1 conversion done

ADC1 channel 2 conversion done

ADC1 channel 3 conversion done

ADC1 channel 4 conversion done

ADC1 channel 5 conversion done

ADC1 channel 6 conversion done

ADC1 channel 7 conversion done

DMA requests are generated if the VFx flag of the respective result register is set to ‘1’ (in all sequencer mode except software mode).

24.7 Interrupts

The ADC1 generates the following interrupts:

ADC1 Exceptional Sequence Measurement finished, status bit **IS**.ESM_STS

ADC1 Exceptional Interrupt Measurement finished, status bit **IS**.EIM_STS

ADC1 channel 0 conversion done interrupt, **IS**.CH0

ADC1 channel 1 conversion done interrupt, **IS**.CH1

ADC1 channel 2 conversion done interrupt, **IS**.CH2

ADC1 channel 3 conversion done interrupt, **IS**.CH3

ADC1 channel 4 conversion done interrupt, **IS**.CH4

ADC1 channel 5 conversion done interrupt, **IS**.CH5

ADC1 channel 6 conversion done interrupt, **IS**.CH6

ADC1 channel 7 conversion done interrupt, **IS**.CH7

Interrupts are generated if the VFx flag of the respective result register is set to ‘1’ (in all sequencer modes except software mode).

Figure 211 shows the interrupt generation of ADC1.

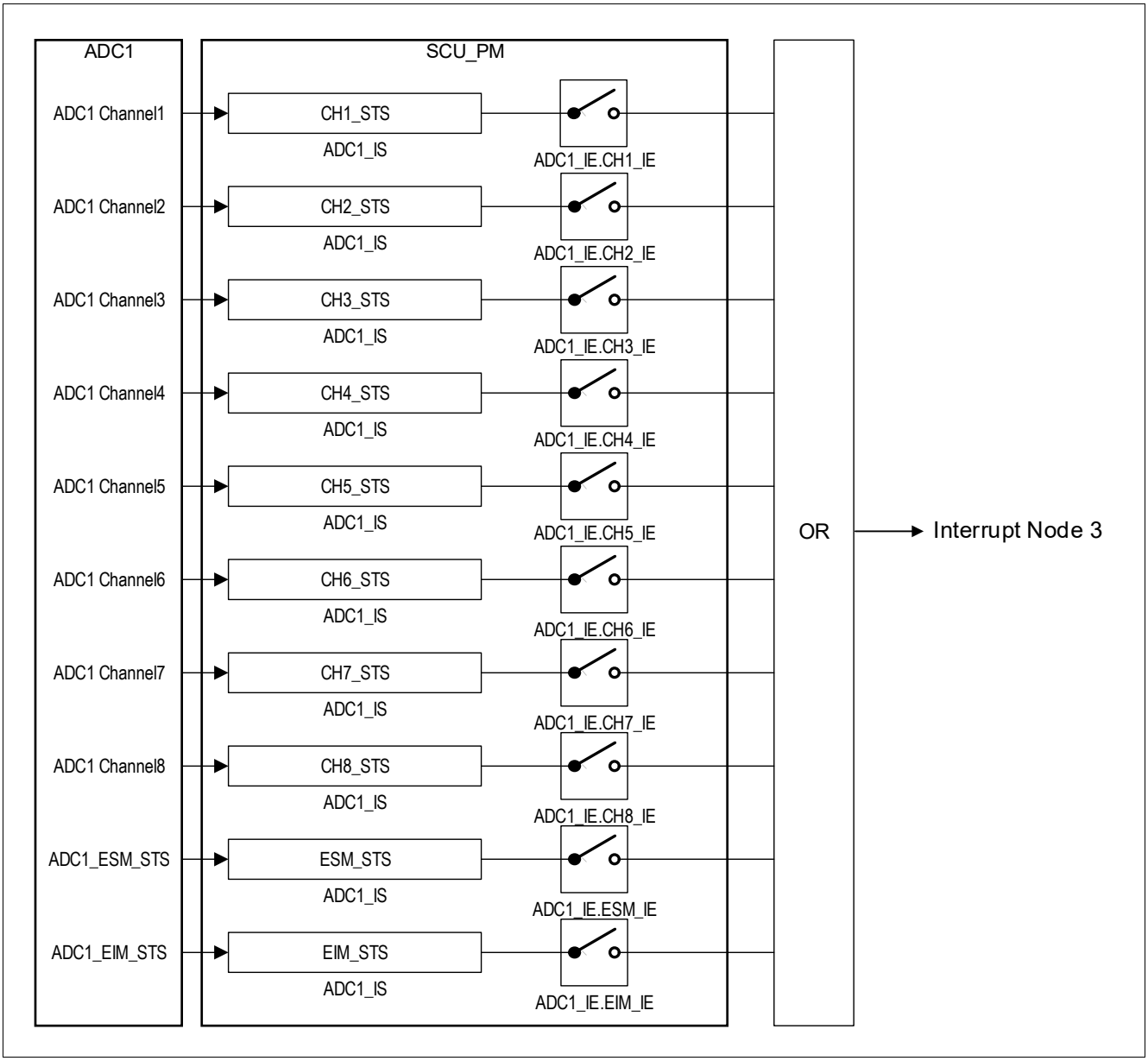


Figure 211 ADC1 Interrupt Generation

24.7.1 Interrupt Registers

Table 181 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Interrupt Registers			
IS	ADC1 Interrupt Status Register	78_H	$0000\ 0000_H$
IE	ADC1 Interrupt Enable Register	$7C_H$	$0000\ 0000_H$
ICLR	ADC1 Interrupt Status Clear Register	80_H	$0000\ 0000_H$

The registers are addressed wordwise.

ADC1 Interrupt Status Register

The register is reset by RESET_TYPE_3.

IS

ADC1 Interrupt Status Register **(78_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						**ESM_STS**	**EIM_STS**	**CH7_STS**	**CH6_STS**	**CH5_STS**	**CH4_STS**	**CH3_STS**	**CH2_STS**	**CH1_STS**	**CH0_STS**
r						r	r	r	r	r	r	r	r	r	r

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
ESM_STS	9	r	**Exceptional Sequence Measurement (ESM) Status** 0_B **INACTIVE**, No ESM has occurred 1_B **ACTIVE**, ESM occurred
EIM_STS	8	r	**Exceptional Interrupt Measurement (EIM) Status** 0_B **INACTIVE**, No EIM occurred 1_B **ACTIVE**, EIM occurred
CH7_STS	7	r	**ADC1 Channel 7 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 7 Interrupt has occurred 1_B **ACTIVE**, Channel 7 Interrupt has occurred
CH6_STS	6	r	**ADC1 Channel 6 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 6 Interrupt has occurred 1_B **ACTIVE**, Channel 6 Interrupt has occurred

Field	Bits	Type	Description
CH5_STS	5	r	**ADC1 Channel 5 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 5 Interrupt has occurred 1_B **ACTIVE**, Channel 5 Interrupt has occurred
CH4_STS	4	r	**ADC1 Channel 4 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 4 Interrupt has occurred 1_B **ACTIVE**, Channel 4 Interrupt has occurred
CH3_STS	3	r	**ADC1 Channel 3 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 3 Interrupt has occurred 1_B **ACTIVE**, Channel 3 Interrupt has occurred
CH2_STS	2	r	**ADC1 Channel 2 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 2 Interrupt has occurred 1_B **ACTIVE**, Channel 2 Interrupt has occurred
CH1_STS	1	r	**ADC1 Channel 1 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 1 Interrupt has occurred 1_B **ACTIVE**, Channel 1 Interrupt has occurred
CH0_STS	0	r	**ADC1 Channel 0 Interrupt Status** Conversion of Channel has finished 0_B **INACTIVE**, No Channel 0 Interrupt has occurred 1_B **ACTIVE**, Channel 0 Interrupt has occurred

ADC1 Interrupt Status Clear Register

The register is reset by RESET_TYPE_3.

ICLR
ADC1 Interrupt Status Clear Register **(80_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						ESM_ICLR	EIM_ICLR	CH7_ICLR	CH6_ICLR	CH5_ICLR	CH4_ICLR	CH3_ICLR	CH2_ICLR	CH1_ICLR	CH0_ICLR
r						wi	wi	wi	wi	wi	wi	wi	wi	wi	wi

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
ESM_ICLR	9	wi	**Exceptional Sequence Measurement (ESM) Status Clear** 0_B **Do not clear**, ESM is not cleared 1_B **Clear**, ESM is cleared
EIM_ICLR	8	wi	**Exceptional Interrupt Measurement (EIM) Status Clear** 0_B **Do not clear**, EIM is not cleared 1_B **Clear**, EIM is cleared
CH7_ICLR	7	wi	**ADC1 Channel 7 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared
CH6_ICLR	6	wi	**ADC1 Channel 6 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared
CH5_ICLR	5	wi	**ADC1 Channel 5 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared
CH4_ICLR	4	wi	**ADC1 Channel 4 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared
CH3_ICLR	3	wi	**ADC1 Channel 3 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared

Field	Bits	Type	Description
CH2_ICLR	2	wi	**ADC1 Channel 2 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared
CH1_ICLR	1	wi	**ADC1 Channel 1 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared
CH0_ICLR	0	wi	**ADC1 Channel 0 Interrupt Status Clear** Interrupt status is cleared 0_B **Do not clear**, interrupt status is not cleared 1_B **Clear**, interrupt status is cleared

ADC1 Interrupt Enable Register

The register is reset by RESET_TYPE_3.

IE

ADC1 Interrupt Enable Register **($7C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						**ESM_IE**	**EIM_IE**	**CH7_IE**	**CH6_IE**	**CH5_IE**	**CH4_IE**	**CH3_IE**	**CH2_IE**	**CH1_IE**	**CH0_IE**
r						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RES	31:10	r	**Reserved** Always read as 0
ESM_IE	9	rw	**Exceptional Sequence Measurement (ESM) Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
EIM_IE	8	rw	**Exceptional Interrupt Measurement (EIM) Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH7_IE	7	rw	**ADC1 Channel 7 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH6_IE	6	rw	**ADC1 Channel 6 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled

Field	Bits	Type	Description
CH5_IE	5	rw	**ADC1 Channel 5 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH4_IE	4	rw	**ADC1 Channel 4 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH3_IE	3	rw	**ADC1 Channel 3 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH2_IE	2	rw	**ADC1 Channel 2 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH1_IE	1	rw	**ADC1 Channel 1 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled
CH0_IE	0	rw	**ADC1 Channel 0 Interrupt Enable** 0_B **DISABLED**, Interrupt disabled 1_B **ENABLED**, Interrupt enabled

24.8 Module Interfaces

24.8.1 ADC1 Hardware Trigger Selection

Hardware Trigger Inputs

Figure 212 shows the ADC1 trigger selection.

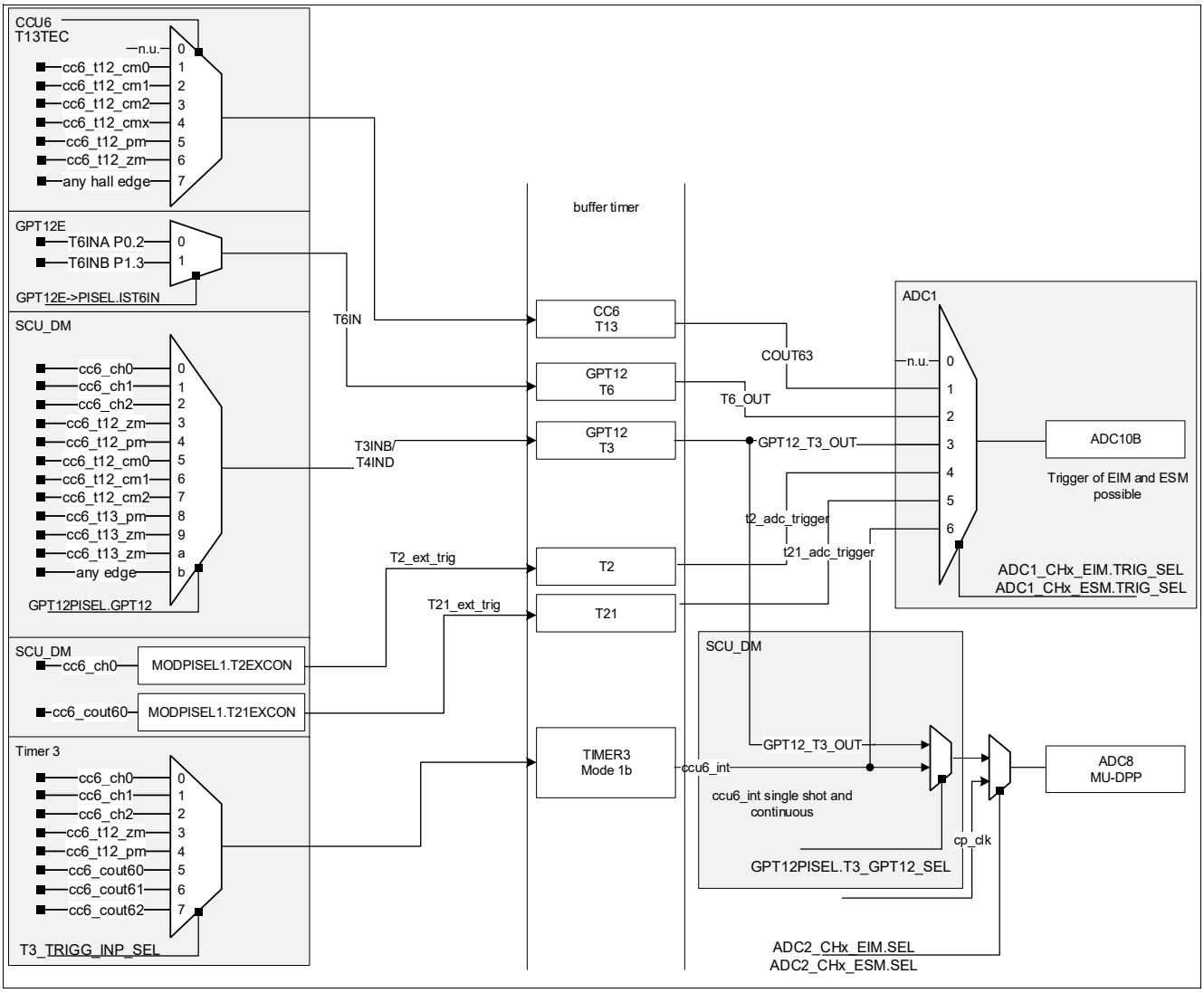


Figure 212 ADC1 Trigger Selection

25 14-Bit Sigma Delta ADC (ADC3 / ADC4)

Note: SDADC is not available in all products of the TLE987x family. See the datasheet for details.

25.1 Features

Module features:

- Full differential capacitive input
- 14 bit resolution
- Full differential scale: $V_{diff,lin}$
- Extended differential input range with reduced accuracy: $V_{diff,nonlin}$
- Sampling frequency: $f_{ADC3/4}$, $f_{ADC3/4}$ derived from f_{SYS} divided by SDADCCLK_DIV (see **APCLK_CTRL2**)
- Programmable oversampling ratio: *OSR*
- High supply rejection ratio
- Internal clock jittering

25.2 Functional description

The 2 integrated 14 Bit Sigma Delta ADCs are building a Sensor Interface for an external AMR / GMR Sensor. The application configuration is shown below.

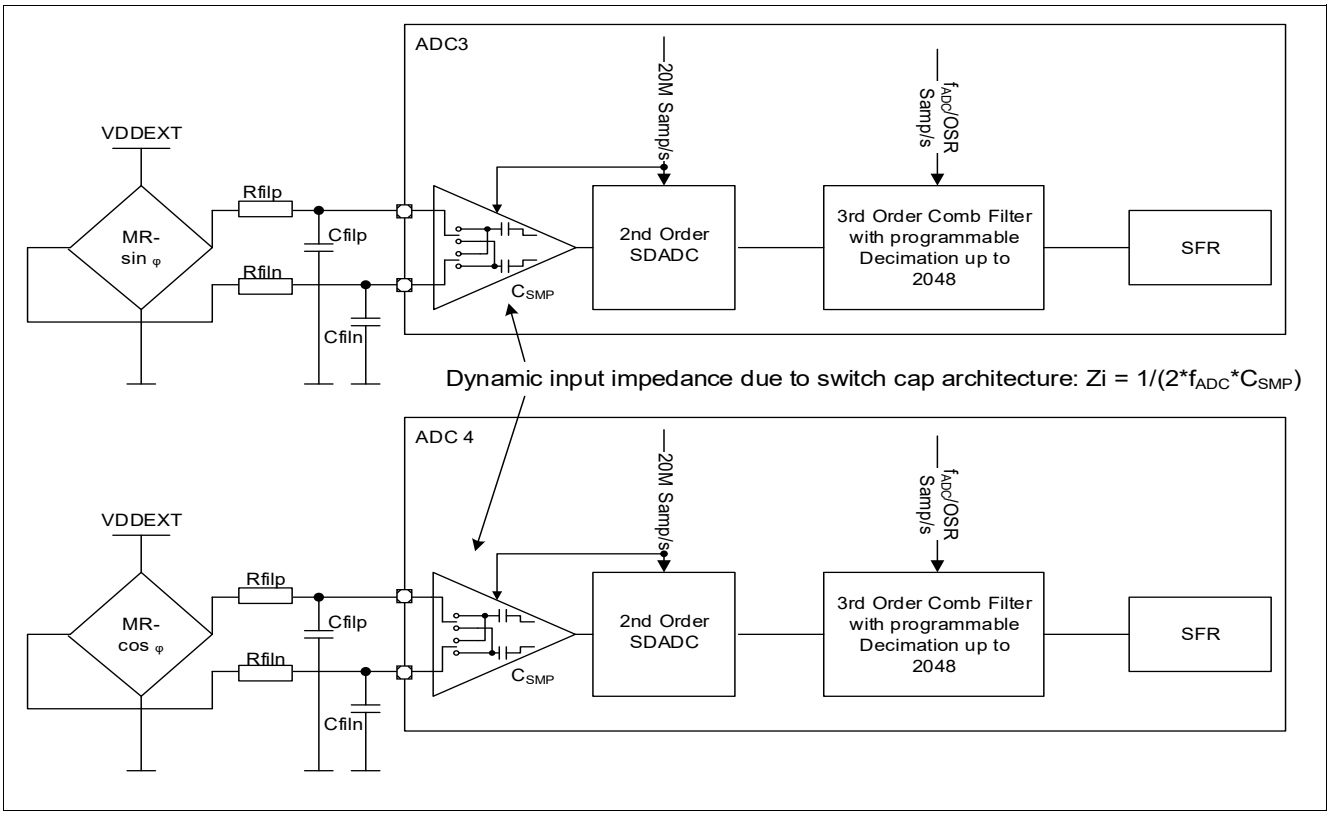


Figure 213 Application of integrated Sigma Delta ADCs

The AMR / GMR Sensor provides two output signals, which are $V_x = A_x \cos \phi_x + B_x$ and $V_y = A_y \cos \phi_y + B_y$. The sine and cosine are a result of the sensed motor rotation.

Both signals $V_x = A_x \cos \phi_x + B_x$ and $V_y = A_y \cos \phi_y + B_y$ have an amplitude A and an Offset B (here called V_{CMX} and V_{CMY}). The min/max-values of A and B are listed in the electrical characteristics of the product.

To determine the angle of this motor rotation two Sigma Delta ADC modules are required. The following well known equation gives the required angle:

(25.1)

$$\varphi = \arctan\left(\frac{V_x}{V_y}\right)$$

With the knowledge of the angle ϕ the motor motion can be controlled. The Sensor is supplied with the VDDEXT supply. As its output signal is ratiometric to its supply voltage, the ADC has also to take the supply voltage as reference.

The ADCs used for this application contain second order Sigma Delta ADCs, with programmable oversampling ratios of up to 2048. Its inputs are connected to Port 2.w, Port 2.x, Port 2.y and Port 2.z. The required input configuration registers are placed in the 10 Bit SAR ADC module.

Attention: ***to ensure a proper functionality of the ADC Module, it is required that already at power-up a proper configuration of the ADC inputs is done, otherwise ADC can become instable. In addition it is recommended to activate the analog ADC 30 fADC clock cycles prior to enabling the digital filter section. This is accomplished by setting the corresponding control register bits in subsequent SFR write operations. First valid results are available after 3*OSR / fADC***

The bandwidth of the sensor shown as an example is 30 kHz. Nevertheless there are applications with higher motor operating frequency and thus higher sensor bandwidth. The maximum input bandwidth is limited by the Nyquist criteria with the overall sampling rate defined using the following formula:

$f_{SAMP} = f_{ADC3/4} / OSR$.

Thus at high OSR rates an additional external low pass filter is required to limit the input signal bandwidth.

Note: due to the switched capacitor nature of each ADC's input stage an $f_{ADC3/4}$ dependent input impedance has to be considered: the input signal will be attenuated depending on the external series resistances at each ADC input.

A rough estimation of the ADC's input impedance is given below:

$zI(ADC) = 1 / (2 * f_{ADC3/4} * 100fF)$;

i.e.: $f_{ADC3/4} = 20MHz$: $zI = 250kOhm$

The required angle resolution is < 0,1° (TUE), which is <13 Bit.

The detailed structure of the used SDADC is shown below. It has the following components:

- differential capacitive input with a fixed gain stage
- 2 SC-Integrators running at $f_s = 20$ MHz
- Quantizer
- two feedback DACs
- 3rd order comb filter

25.2.1 Input voltage range of SDADC

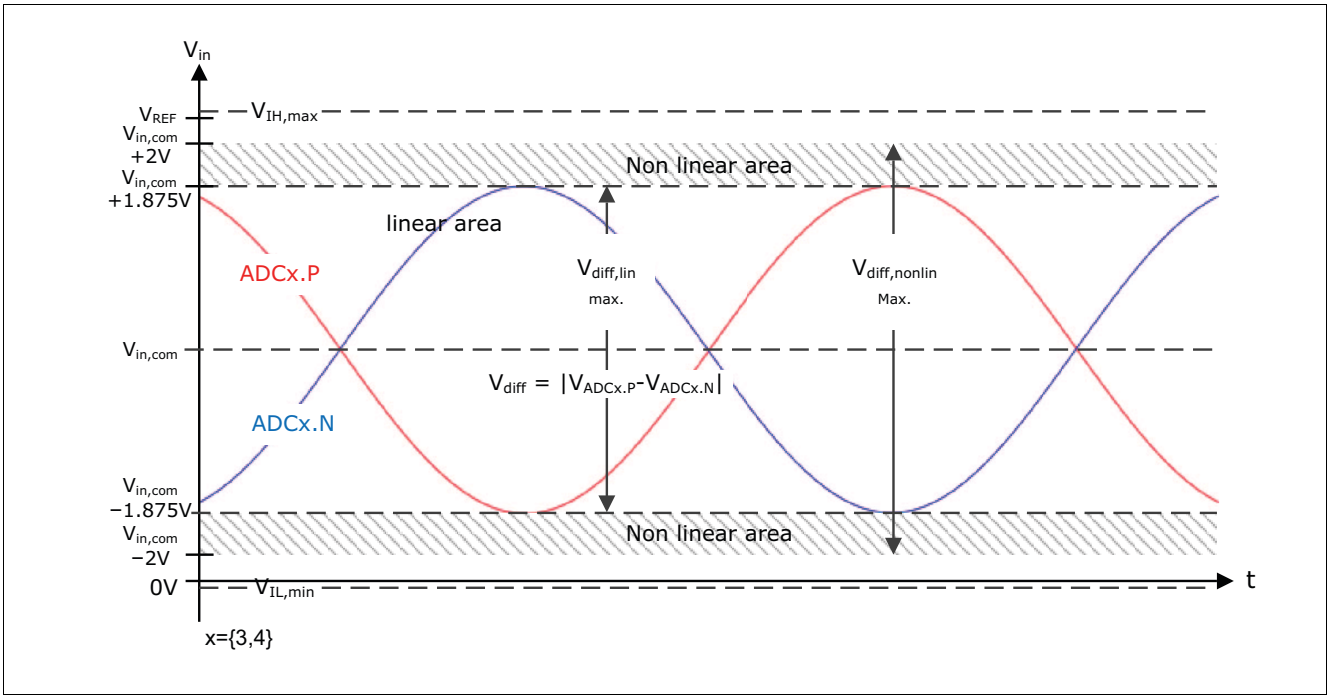


Figure 214 Input voltage range

25.2.2 Calculation of SDADC output code

The code at the output of the decimation filter is calculated as follows:

$$\text{DEC_Filt_Output} = \left(\frac{V_{ADCx.P} - V_{ADCx.N}}{V_{REF}}\right) * ATT_{ADC34} * \left(2^{14} - 1\right)$$

Figure 215 Calculation of SDADC output code

25.2.3 Offset Compensation

The SDADC makes it possible to measure the offset of the internal circuit (caused by integrated circuit non-idealities) using an integrated dedicated block to determine the offset value, which than can be compensated by subtracting it from every subsequent measurement.

During the offset measurement phase the inputs of the SDADC are disconnected from the module and switched to an internal voltage, so it is not possible to measure the input signal at the same time.

To enable the offset measurement the corresponding enable bit **ADC3_OFS_MEAS_EN** for ADC3 or **ADC4_OFS_MEAS_EN** for ADC4 has to be set.

The offset result can be read in the **ADC3_ADC4_RESU** register.

25.3 SDADC Control Register

The next chapter lists the configuration possibilities of the two SDADCs, ADC3 and ADC4, used as TMR / AMR / GMR Sensor interface.

Table 182 Register Address Space

Module	Base Address	End Address	Note
ADC34	40008000_H	$4000BFFF_H$	SDADC

Table 183 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
CTRL_STS	ADC3 & ADC4 Control and Status	00_H	$0480\ 0000_H$
RESU	ADC3_ADC4_RESU	04_H	$0000\ 0000_H$

The registers are addressed wordwise.

ADC3 / ADC4 Control and Status

The register is reset by RESET_TYPE_3.

The following register consists of control and status bits. This Register is cleared by every reset.

CTRL_STS

ADC3 & ADC4 Control and Status **(00_H)** **Reset Value: $0480\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC4_OSR				ADC34_DITHVAL				ADC34_DITHEN	ADC34_EoC_CNT		ADC4_EoC_STS	ADC4_EoC	ADC4_SOC	ADC4_OFS_MEAS_EN	ADC4_EN
rw				rw				rw	rw		r	r	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3_OSR				ADC34_REF_SEL	RES			RES	ADC34_DREQ_SEL		ADC3_EoC_STS	ADC3_EoC	ADC3_SOC	ADC3_OFS_MEAS_EN	ADC3_EN
rw				rw	r			r	rw		r	r	rw	rw	rw

Field	Bits	Type	Description
ADC4_OSR	31:28	rw	**ADC4 Oversampling Ratio** 0111_B **128**, Factor 128 1000_B **256**, Factor 256 1001_B **512**, Factor 512 1010_B **1024**, Factor 1024 1011_B **2048**, Factor 2048
ADC34_DITHVAL	27:24	rw	**ADC3/4 Dither Value (Amplitude) = ADC34_DITHVAL * 12,5mV** 0000_B **0**, Factor 0 0001_B **1**, Factor 1 0010_B **2**, Factor 2 0011_B **3**, Factor 3 0100_B **4**, Factor 4 0101_B **5**, Factor 5 0110_B **6**, Factor 6 0111_B **7**, Factor 7 1000_B **8**, Factor 8 1001_B **9**, Factor 9 1010_B **10**, Factor 10 1011_B **11**, Factor 11 1011_B **12**, Factor 12 1011_B **13**, Factor 13 1011_B **14**, Factor 14 1011_B **15**, Factor 15
ADC34_DITHEN	23	rw	**ADC3/4 Dithering Enable** 0_B **Disable**, ADC34 Dithering 1_B **Enable**, ADC34 Dithering

Field	Bits	Type	Description
ADC34_EoC_CNT	22:21	rw	**ADC3/4 EoC counter** 00_B **0**, ADC EoC with every real conversion (IRQ with every EoC) 01_B **8**, ADC EoC with every 8th real conversion (IRQ with every EoC) 10_B **16**, ADC EoC with every 16th real conversion (IRQ with every EoC) 11_B **32**, ADC EoC with every 32nd real conversion (IRQ with every EoC)
ADC4_EoC_STS	20	r	**ADC4 End of Conversion** 0_B **Conversion ongoing**, ADC4 conversion still in progress 1_B **Conversion finished**, ADC4 conversion terminated
ADC4_EoC	19	r	**ADC4 End of Conversion (DMA Ch13 Request)** 0_B **No DMA Request**, ADC4 does not request DMA resource 1_B **DMA Request**, ADC4 requests DMA resource
ADC4_SOC	18	rw	**ADC4 conversion enable (set after enabling ADC to allow settling of the analog converter part)**[1)] 0_B **Disable**, ADC4 data processing 1_B **Enable**, ADC4 data processing
ADC4_OFS_MEAS_EN	17	rw	**ADC4 Offset Measurement Enable** 0_B **DISABLE**, Offset Measurement 1_B **ENABLE**, Offset Measurement
ADC4_EN	16	rw	**ADC4 analog frontend Enable** 0_B **DISABLE**, ADC4 analog frontend switched off 1_B **ENABLE**, ADC4 analog frontend switched on
ADC3_OSR	15:12	rw	**ADC3 Oversampling Ratio** 0111_B **128**, Factor 128 1000_B **256**, Factor 256 1001_B **512**, Factor 512 1010_B **1024**, Factor 1024 1011_B **2048**, Factor 2048
ADC34_REF_SEL	11	rw	**ADC3/4 reference voltage selection** 0_B **VAREF**, use VAREF / VAGND as reference 1_B **VDD_EXT**, use VDD_EXT / GND as reference
RES	10:8	r	**Reserved** Always read as 0
RES	7	r	**Reserved** Always read as 0
ADC34_DREQ_SEL	6:5	rw	**DMA request source select for DMA Ch13** 00_B **no DMA request**, DMA request masked for both ADCs 01_B **ADC3 DREQ**, only ADC3_EOC DMA request 10_B **ADC4 DREQ**, only ADC4_EOC DMA request 11_B **ADC34 DREQ**, ADC3_EOC or ADC4_EOC DMA request

Field	Bits	Type	Description
ADC3_EoC_STS	4	r	**ADC3 End of Conversion** 0_B **Conversion ongoing**, ADC3 conversion still in progress 1_B **Conversion finished**, ADC3 conversion terminated
ADC3_EoC	3	r	**ADC3 End of Conversion (DMA Ch13 Request)** 0_B **No DMA Request**, ADC3 does not request DMA resource 1_B **DMA Request**, ADC3 requests DMA resource
ADC3_SOC	2	rw	**ADC3 conversion enable (set after enabling ADC to allow settling of the analog converter part)**[2)] 0_B **Disable**, ADC3 data processing 1_B **Enable**, ADC3 data processing
ADC3_OFS_MEAS_EN	1	rw	**ADC3 Offset Measurement Enable** 0_B **DISABLE**, Offset Measurement 1_B **ENABLE**, Offset Measurement
ADC3_EN	0	rw	**ADC3 analog frontend Enable** 0_B **DISABLE**, ADC3 analog frontend switched off 1_B **ENABLE**, ADC3 analog frontend switched on

1) ADC4_SOC: Note that while ADC4_SOC is set the digital data processing is active independent of the state of the analog frontend (ADC4_EN). It is therefore recommended to deactivate the SDADC by clearing both, ADC4_SOC and ADC4_EN within one register access.

2) ADC3_SOC: Note that while ADC3_SOC is set the digital data processing is active independent of the state of the analog frontend (ADC3_EN). It is therefore recommended to deactivate the SDADC by clearing both, ADC3_SOC and ADC3_EN within one register access.

ADC3_ADC4_RESU

The register is reset by RESET_TYPE_3.

RESU

ADC3_ADC4_RESU **(04_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC4_RESU															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3_RESU															
r															

Field	Bits	Type	Description
ADC4_RESU	31:16	r	**ADC4 Result Register** 2th Complement ADC Conversion Result
ADC3_RESU	15:0	r	**ADC3 Result Register** 2th Complement ADC Conversion Result

26 High-voltage monitor input

26.1 Features

- High-voltage input with V_{MONth} threshold voltage
- Integrated selectable pull-up and pull-down current sources
- Wake capability for power-saving modes
- Level change sensitivity configurable for transitions from low to high, high to low, or both directions
- The MON input can also be evaluated with **ADC2** in active mode (see also Chapter **Analog Digital Converter**).

26.2 Introduction

This module is dedicated to monitor external voltage levels above or below a specified threshold or it can be used to detect a wake-up event at the high-voltage MON pin in low-power mode. The input level can be monitored if the module is enabled (PMU_MON_CNF).

To use the Wake function when the IC is in a low-power mode, the monitoring pin is switched to Sleep mode via the SFR bit EN.

26.2.1 Block diagram

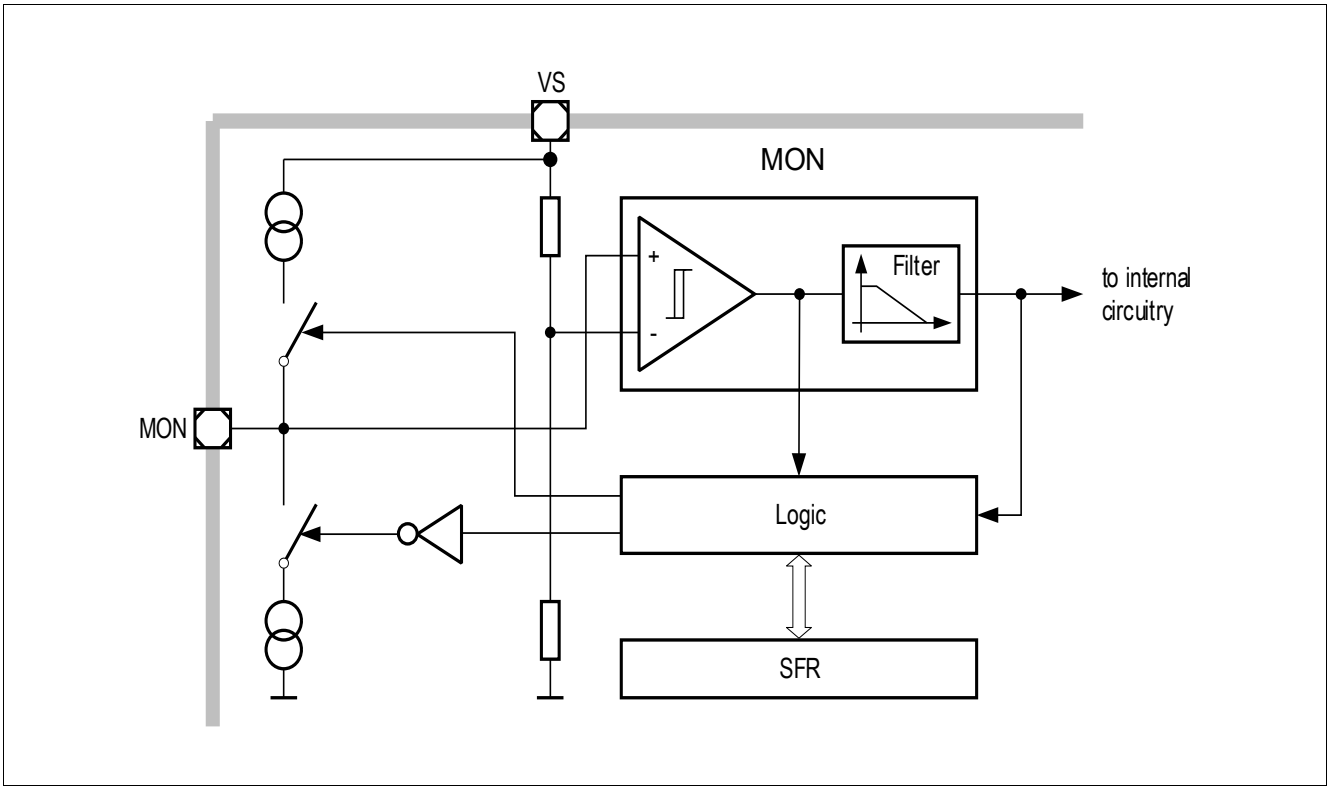


Figure 216 High-voltage monitor input block diagram

26.3 Functional Description

For a wake-up on a positive voltage transition, the **RISE** bit has to be configured. For a wake-up on a negative voltage transition, the corresponding bit **FALL** has to be set. This configuration can also be used for level change detection in active mode.

The IN also includes an input circuit with pull-up (can be activated by **PU** bit) and pull-down (can be activated by **PD** Bit) current sources to define a certain voltage level with open inputs and a filter function to avoid wake-up events caused by unwanted voltage transients at the module input.

When automatic current source selection is enabled, a voltage level at the IN input of $V_{MON_th} < V_{MONx} < V_S$-1V activates the pull-up current source. If the IN voltage is between 1 V < $V_{MONx} < V_{MON_th}$ the pull-down sink is activated, providing stable levels at the monitoring/wake-up inputs. Below and above these voltage ranges the current is minimized to a leakage current. This automatic activation of the current sources, has to be done by setting **PU** and **PD** bit to one at the same time.

Note: **In case a Monitoring Input is deactivated by setting bit EN to zero, it can neither be used as a wake-up source nor can it be used to detect logic levels!**

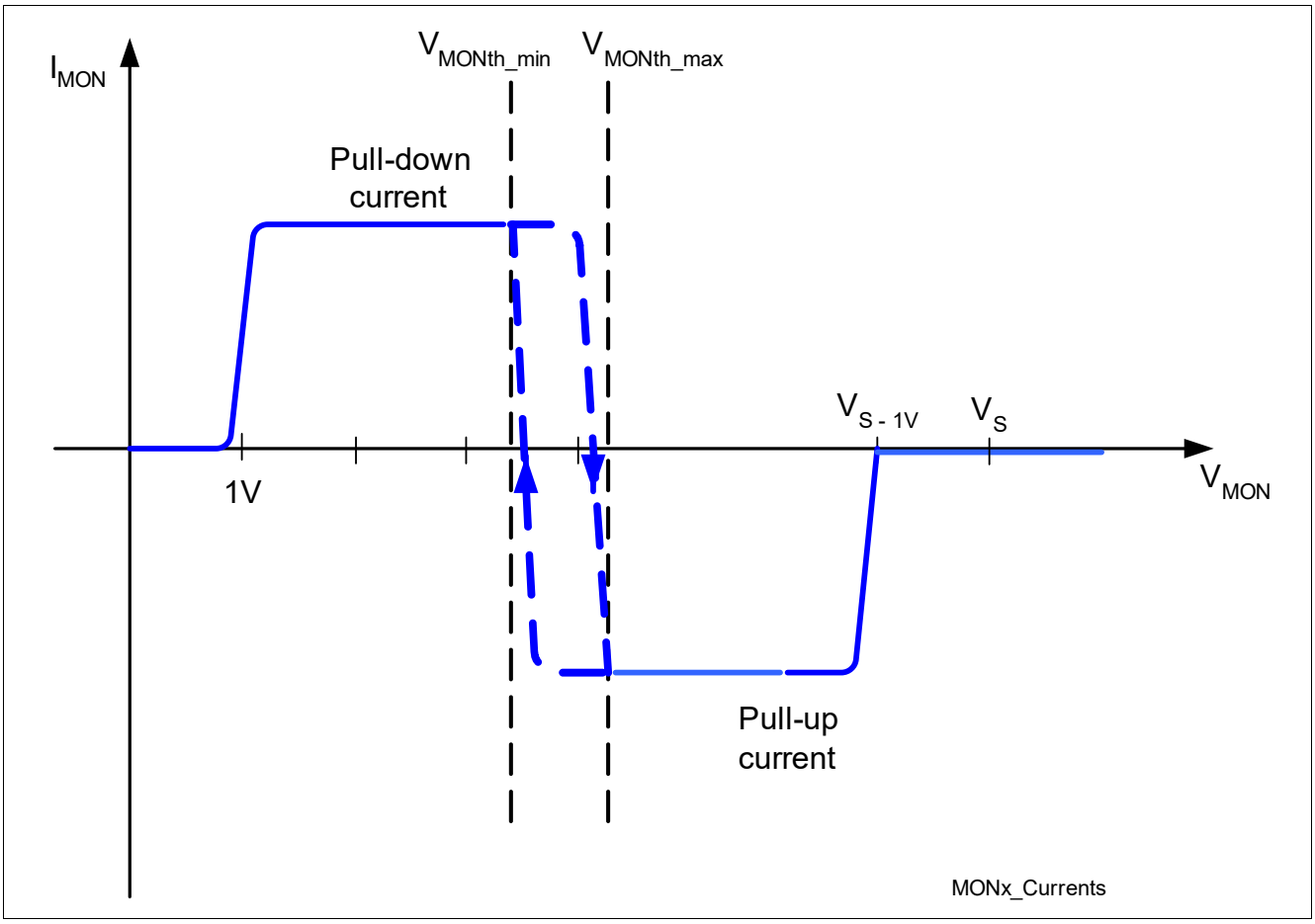


Figure 217 Module - HV_MON Input Characteristics for switchable pull current and static pull-down (on top) or pull-up

High-voltage monitor input

The following tables provides an overview of the configuration possibilities on the INs via **SFR**.

Table 184 includes all pull-up and pull-down setup scenarios which can be chosen for one IN. **Table 185** shows an overview of the available states of a IN.

Table 184 Pull-Up / Pull-Down Input Current

PU	PD	Output Current	Description
0	0	leakage current[1)]	pull-up/down current source disabled
0	1	pull-down	pull-down current source enabled (for low active switches)
1	0	pull-up	pull-up current source enabled (for high active switches)
1	1	switchable[2)]	pull-up/down depending on input voltage

1) All current sources switched off.
2) Will be automatically switched by the MON_IN circuit depending on level of input signal.

Table 185 EN MON Mode definition

EN	Mode	Description
0	disabled[1)]	Monitoring input is disabled (no wake-up possible!)
1	Sleep Mode	Monitoring input is set to low power mode (use for device Low Power Mode)
1	normal mode	Monitoring input is in active mode

1) If a IN is disabled it cannot be used as a wake-up source anymore.

26.4 Register Definition

This chapter describes the configuration register for MON1.

The **Base Address** for this Register is 50004000_H.

The registers listed below are located in the **SFR** address space. To access them, no paging is required.

Table 186 Register Address Space

Module	Base Address	End Address	Note
MON	50004000_H	$50004FFF_H$	Monitor Inputs

Table 187 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Register Definition, Monitor Input Registers			
CNF	Settings Monitor 1	034_H	$0100\ 0111_B$

The registers are addressed wordwise.

26.4.1 Monitor Input Registers

The monitor input registers are part of the PMU. This is due to the fact that this circuit requires supply and clock, during system wide low power modes.

Settings Monitor 1

The register is reset by RESET_TYPE_2.

CNF
Settings Monitor 1 **(034_H)** **Reset Value: $0100\ 0111_B$**

7	6	5	4	3	2	1	0
STS	**RES**	**PU**	**PD**	**RES**	**RISE**	**FALL**	**EN**
r	r	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
STS	7	r	**MON Status Input will only be updated if one of the wake flags CNF.RISE or CNF.FALL are set** **For direct MON status see MODIEN3.MONSTS** 0_B **LOW**, MON input has low status 1_B **HIGH**, MON input has high status
RES	6	r	**Reserved**
PU	5	rw	**Pull-Up Current Source for MON Input Enable** *Note: Works only if EN is enabled* 0_B **DISABLED**, Pull-up source disabled 1_B **ENABLED**, Pull-up source enabled

Field	Bits	Type	Description
PD	4	rw	**Pull-Down Current Source for MON Input Enable** *Note: Works only if EN is enabled* 0_B **DISABLED**, Pull-down source disabled 1_B **ENABLED**, Pull-down source enabled
RES	3	rw	**Reserved** 0_B **DISABLED**, Reading or writing this bit has no effect.
RISE	2	rw	**MON Wake-up on Rising Edge Enable** *Note: Works only if EN is enabled* 0_B **DISABLED**, Wake-up disabled 1_B **ENABLED**, Wake-up enabled
FALL	1	rw	**MON Wake-up on Falling Edge Enable** *Note: Works only if EN is enabled* 0_B **DISABLED**, Wake-up disabled 1_B **ENABLED**, Wake-up enabled
EN	0	rw	**MON Enable** 0_B **DISABLED**, MON disabled 1_B **ENABLED**, MON enabled

27 Bridge driver (incl. charge pump)

27.1 Features

The MOSFET driver is intended to drive external normal-level NFET transistors in bridge configurations. The driver provides many diagnostic functions for detecting faults.

Functional features

- External power NFET transistor driver stage with driver capability of Q_{tot_max} (see data sheet for values).
- Adjustable cross-conduction protection.
- Supply voltage (VSD) monitoring incl. adjustable over- and undervoltage shutdown with configurable interrupt signalling.
- VSD operating range: V_{SD_AM} (see datasheet for values).
- VDS comparators for short-circuit-detection in both on- and off-states.
- Open-load detection in the off-state.
- Flexible PWM frequency range. Rates above 25 kHz require power dissipation and duty-cycle resolution analysis.

27.2 Introduction

The MOSFET driver stage can be used for controlling external power NFET transistors (normal level). The module output is controlled by the SFR or the System PWM Machine (CCU6).

27.2.1 Block diagram

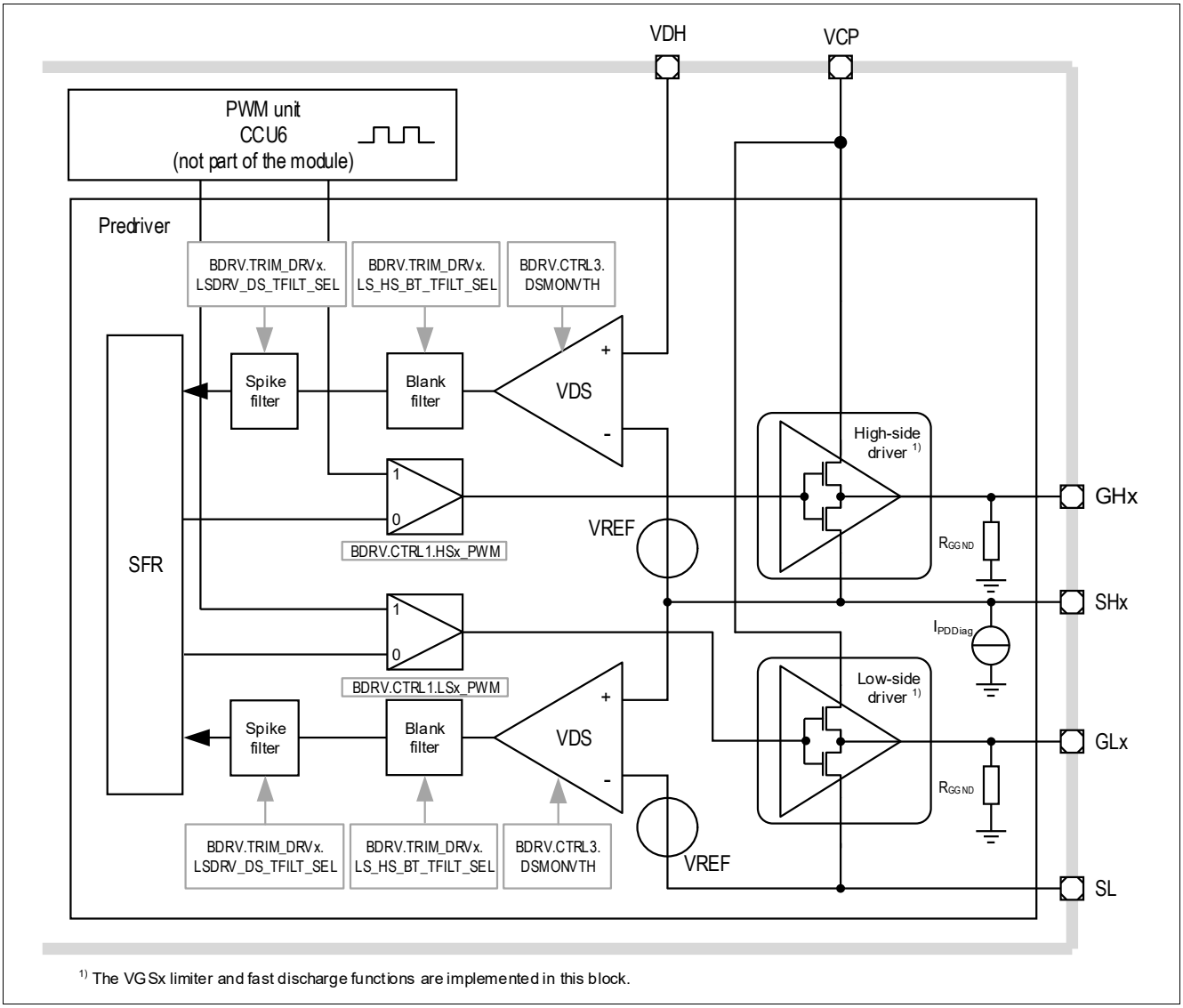


Figure 218 Bridge driver module block diagram (incl. system connections)

27.2.2 General

The bridge driver can be controlled:

- The PWM mode can be enabled by setting the corresponding bit in **CTRL1** and **CTRL2**. The PWM must be configured in the System PWM Module (CCU6). All protection functions are available in PWM mode as well.

Protection functions

- Overcurrent detection and shutdown feature for external MOSFET based on drain-source measurements.
- Programmable minimum cross-current protection time.
- Open-load detection feature in the off-state for external MOSFET.

27.3 Functional Description

27.3.1 Slew Rate Control

For Slew Rate Control the charge and discharge current of the MOSFET gate can be adjusted. This is done by programming the corresponding bits in the SFR.

A separate slew rate control is implemented for both, turn-on and turn-off control phases

The drive current for the MOSFET gate can be shaped for the charge and discharge phase. Therefore 4 current steps with timings can be programmed for charge and discharge phase. The drive current for each current step can be programmed with 5 bits (32 levels) and the length of the current step with 3 bits (50ns each step).

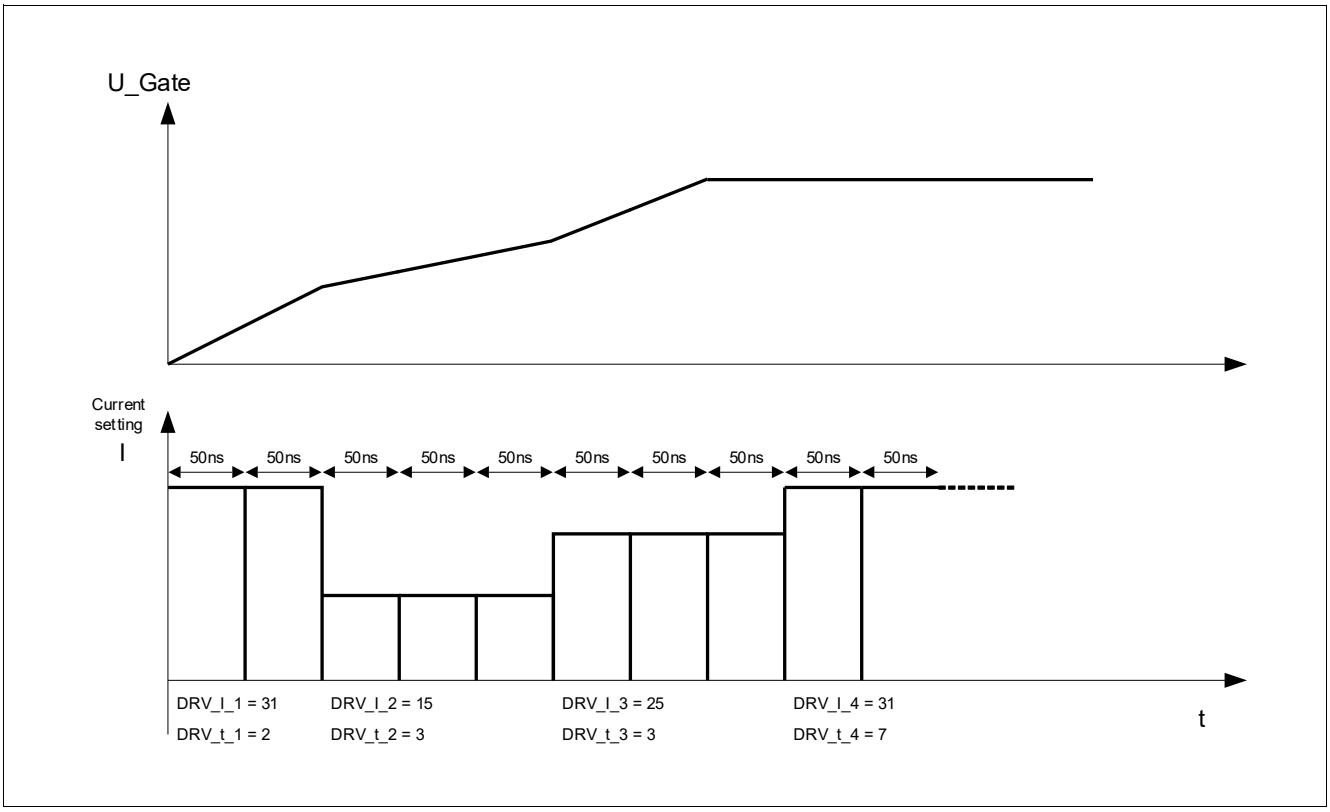


Figure 219 Driver Slew Rate Control

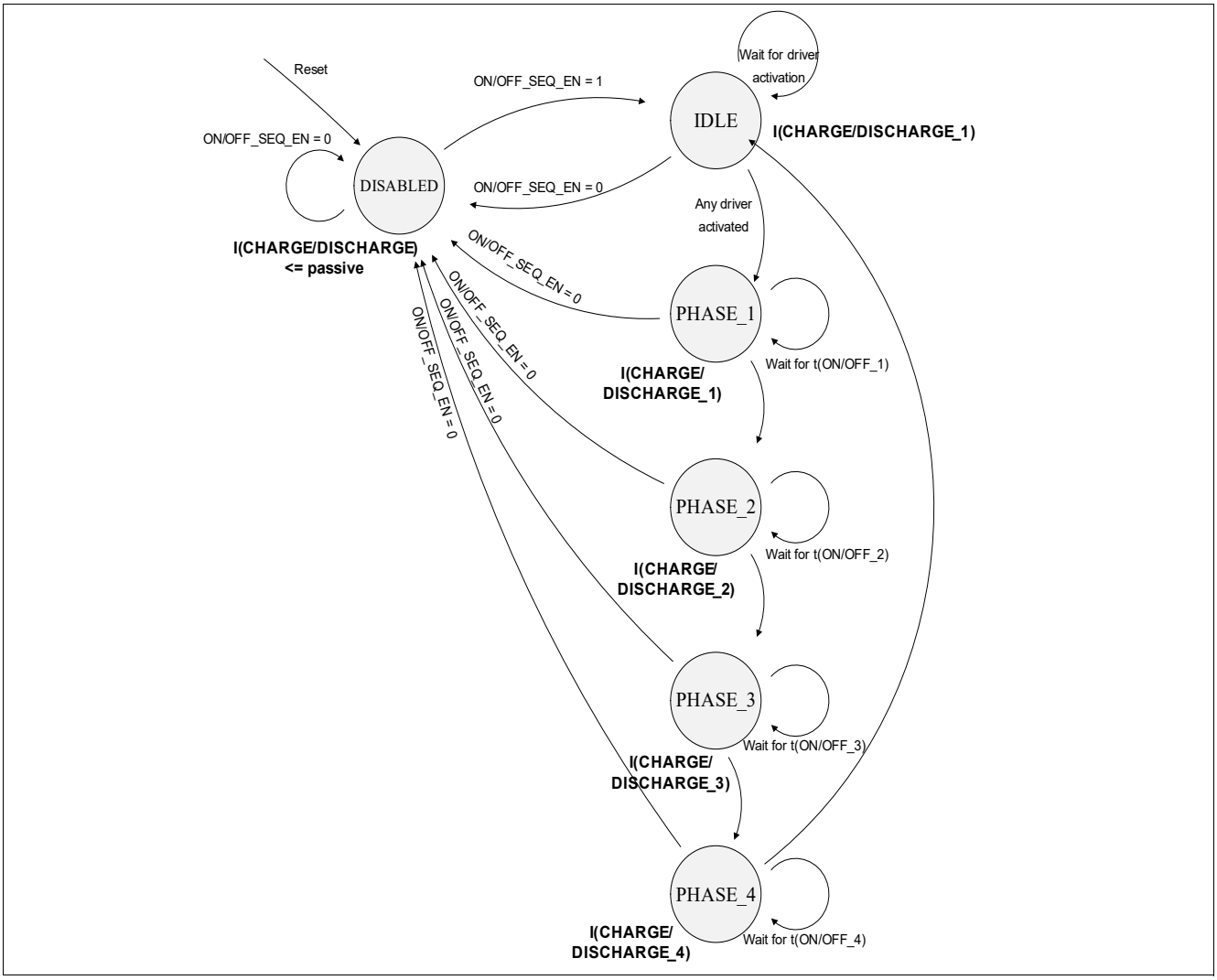


Figure 220 Driver Slew Rate Control Sequencer

27.3.2 PWM Operation

In PWM Mode the Driver Stages are controlled by the on-chip PWM Generator (CCU6). The PWM Mode of the Driver Stages has to be enabled by the corresponding bit in **CTRL1.LSx/HSx_PWM and CTRL2.LSx/HSx_PWM**. The control of the Driver Stages by the PWM Generator gets only active when all LSx/HSx_PWM bits are set to one. The proper PWM Generator configuration must be done in the SFRs of the PWM Module (CCU6).

In PWM Mode the Driver should be enabled by the corresponding bits in the **CTRL1**/**CTRL2** register, first. If this has been done the PWM State Machine can be enabled to deliver the PWM Signal. This procedure is recommended to avoid any unwanted glitches at the driver output.

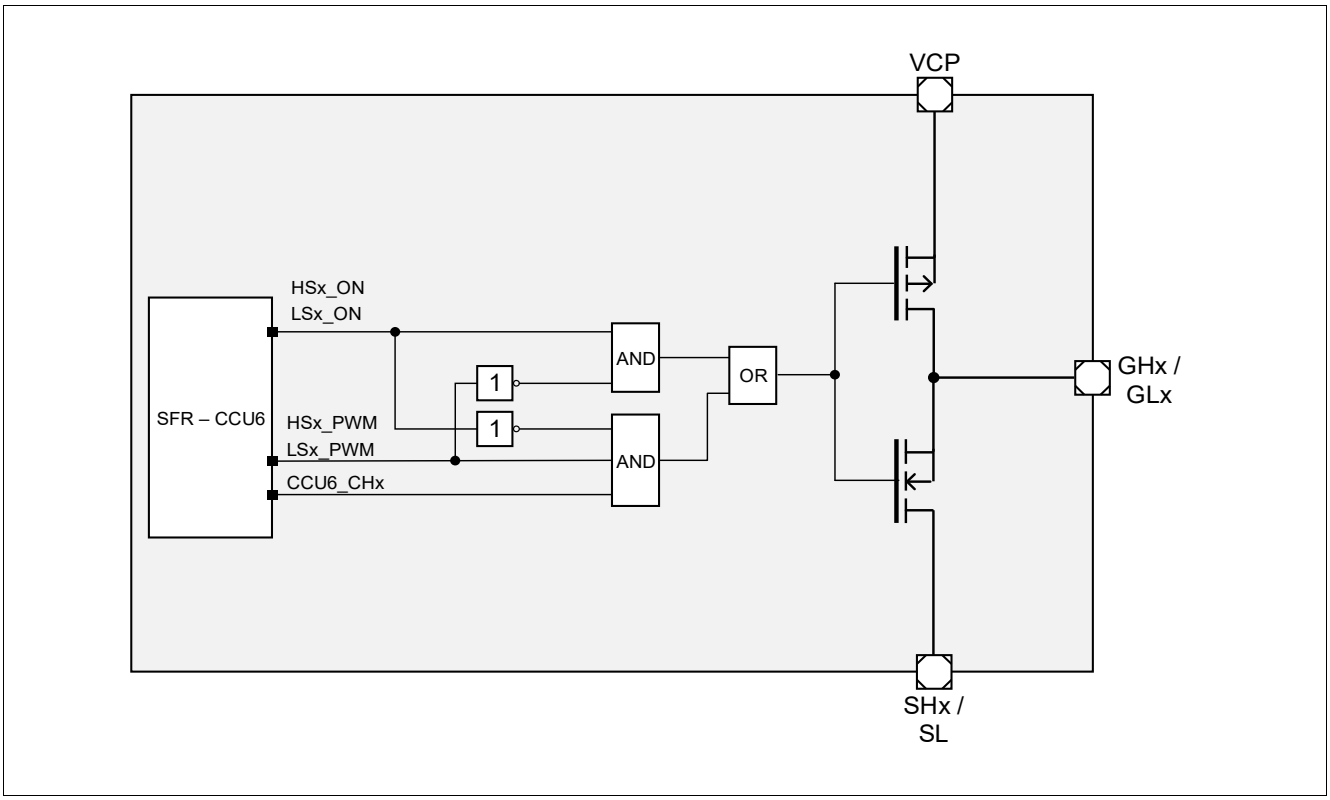


Figure 221 Control Logic of Driver in PWM Mode

The MOSFET Drivers are mapped to following CCU6 channels:

Table 188 Driver Control by CCU6

Driver dedicated to external FET:	Corresponding CCU6 channel (CCU6_Chx)	Comment
LS1	COUT60	
LS2	COUT61	
LS3	COUT62	
HS1	CC60	
HS2	CC61	
HS3	CC62	

27.3.3 Adjustable Cross-Conduction Protection

The cross-conduction protection feature can be enabled or disabled by the bit DRV_CCP_DIS of the register **CTRL3**.

The minimum cross-conduction protection time t_{CCP} can be programmed by the bit fields **TRIM_DRVx**.DRV_CCPTMUL and **CTRL3**.DRV_CCP_TIMSEL according to **Table 189**:

Note: The CCP is based on MI_CLK. The values correspond to a MI_CLK = 20MHz. For a MI_CLK different from 20MHz other timing values have to be expected.

Table 189 Cross-conduction protection time settings

	DRV_CCPTMUL= 00_B	DRV_CCPTMUL= 01_B	DRV_CCPTMUL= 10_B	DRV_CCPTMUL= 11_B
DRV_CCP_TIMSEL=00_B	0.2 µs	0.4 µs	0.8 µs	1.6 µs
DRV_CCP_TIMSEL=01_B	0.4 µs	0.8 µs	1.6 µs	3.2 µs
DRV_CCP_TIMSEL=10_B	0.8 µs	1.6 µs	3.2 µs	6.4 µs
DRV_CCP_TIMSEL=11_B	1.6 µs	3.2 µs	6.4 µs	12.8 µs

Note: The cross-conduction protection feature of the Bridge Driver is only able to increase the effective dead time between the complementary control signals of one half bridge (LSx_ON, HSx_ON) to the selected cross-conduction protection time. If the dead time of the programmed control signals (e.g. from CCU6) is already greater than the selected cross-conduction protection time this feature has no effect.

27.3.4 Supply-Generation of Driver

The driver is supplied by a 2 Stage Charge Pump. The charge pump enables a duty cycle range from 0 - 100%. The regulated output voltage is typically VSD + 14V The charge pump output VCP is monitored via an undervoltage comparator. If undervoltage is detected the drivers will be switched off. The threshold for this undervoltage detection can be adjusted by **VCP_LOWTH2**.

As shown in **Figure 222** the pump requires three external capacitors.

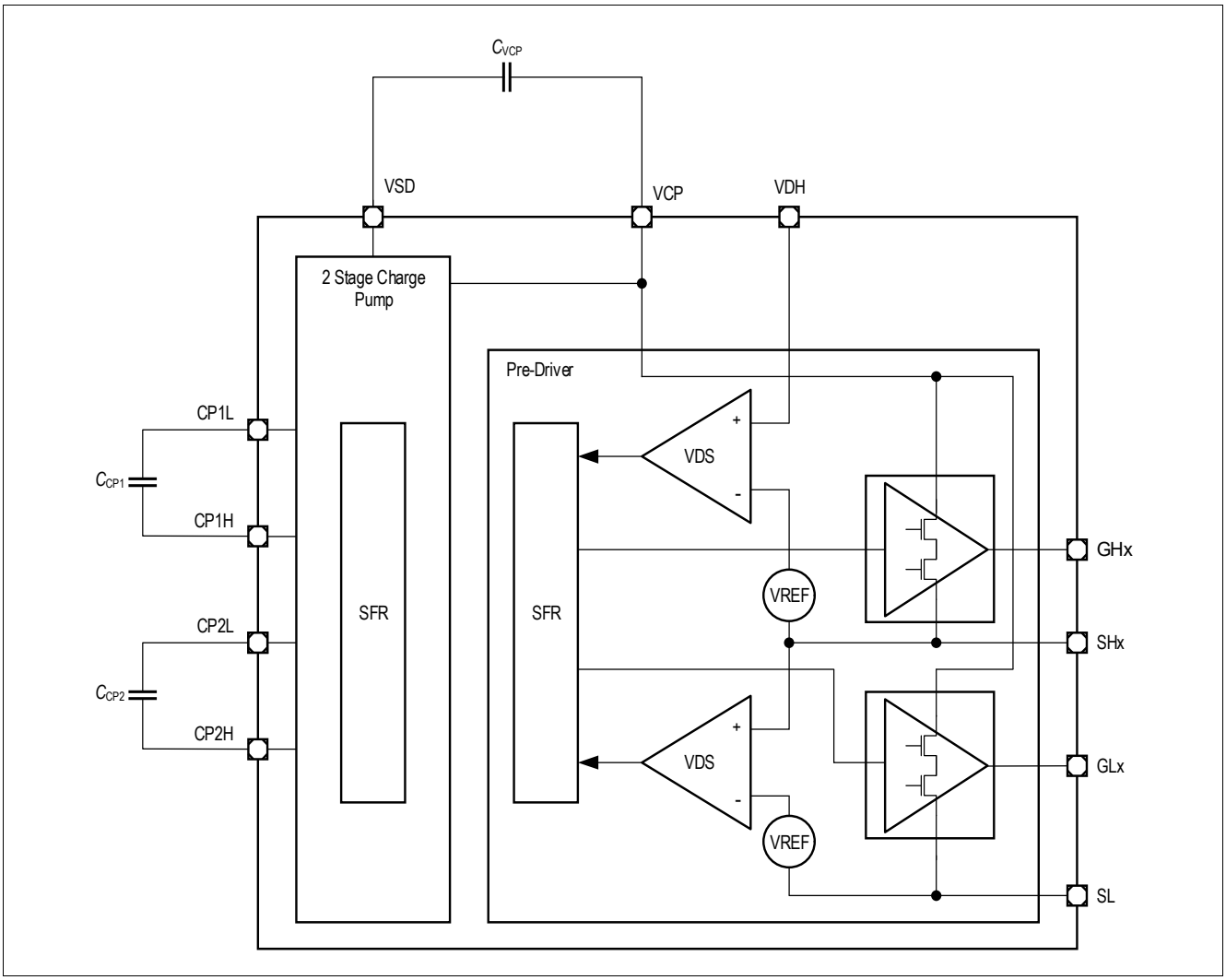


Figure 222 Supply Generation of Driver

27.3.4.1 Clock Generator of Driver Supply

The clock generator of the charge pump uses a spread spectrum technique to minimize emission caused by the charge pump operation on the supply voltage VSD. The structure of the clock generation for the charge pump is shown in the figure below:

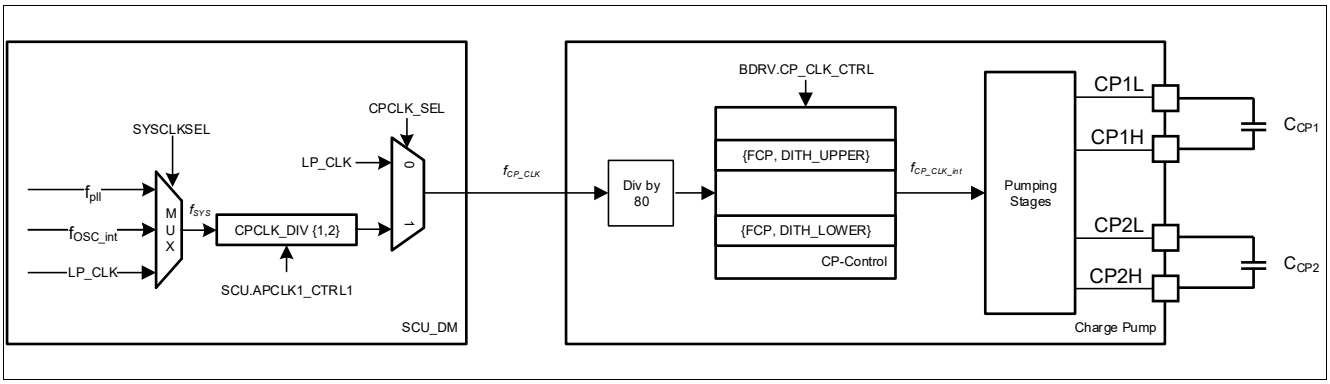


Figure 223 Clock Generation of Charge Pump Block

The charge pump clock $f_{CP_CLK_int}$ is derived from the system clock f_{sys} or LP_CLK. Inside the SCU_DM the system clock is divided by a configurable value and provided as f_{CP_CLK} to the charge pump control block. During operation the frequency of the charge pump is varied between two frequency boundaries. These boundaries are defined by the concatenation of {FCP, DITH_UPPER} bits for the upper boundary and {FCP, DITH_LOWER} bits for the lower boundary (the concatenated bits represent a frequency divider value for f_{CP_CLK}).

27.3.5 Overcurrent Detection - VDS measurement

The Drain Source Voltage (VDS) of each MOSFET is monitored by a comparator. In case the VDS voltage is higher then the limit set in DSMONVTH during the on phase of the MOSFET all drivers are switched off. The bit HSx_OC_STS or LSx_OC_STS is set.

The feature of switching off all drivers (global shutdown) in case one driver has detected a short condition can be disabled by bit HSx_OC_DIS / LSx_OC_DIS. If the bit is set to one only the driver which detected the short will be turned off (local shutdown).

The filter time for the VDS measurements can be adjusted by the bits HSDRV_DS_TFILT_SEL and LSDRV_DS_TFILT_SEL.

The blank time for the VDS measurement can be adjusted by the bits LS_HS_BT_TFILT_SEL.

27.3.6 OFF-State Short Circuit Detection

The Short circuit detection in OFF State is mainly performed by an internally generated test current and the built-in Drain-Source monitoring.

To perform the short circuit detection in off state the following procedure has to be executed:

- The MOSFET Drivers are in off-state
- To detect short circuit on the low side MOSFET, a test current ITEST out of SH1 is activated by a high side current source on pin SH1. This is done by setting the bit HS1_DCS_EN.
- After the test current has settled the VDS Status flag shows the status on the low side MOSFET. If VDS status flag is set the complete voltage drop is across the high side current source, VDS measurement on the LS MOSFET is below the VDS threshold, indicating a short on the low side FET.

27.3.7 Channel turn-on/turn-off delay measurement

For functional test and drive scheme timing optimization a dedicated timer is available to measure the delay between intended external MOS activation and actual turn on (VDS supervision). The principle function is shown in the following figure:

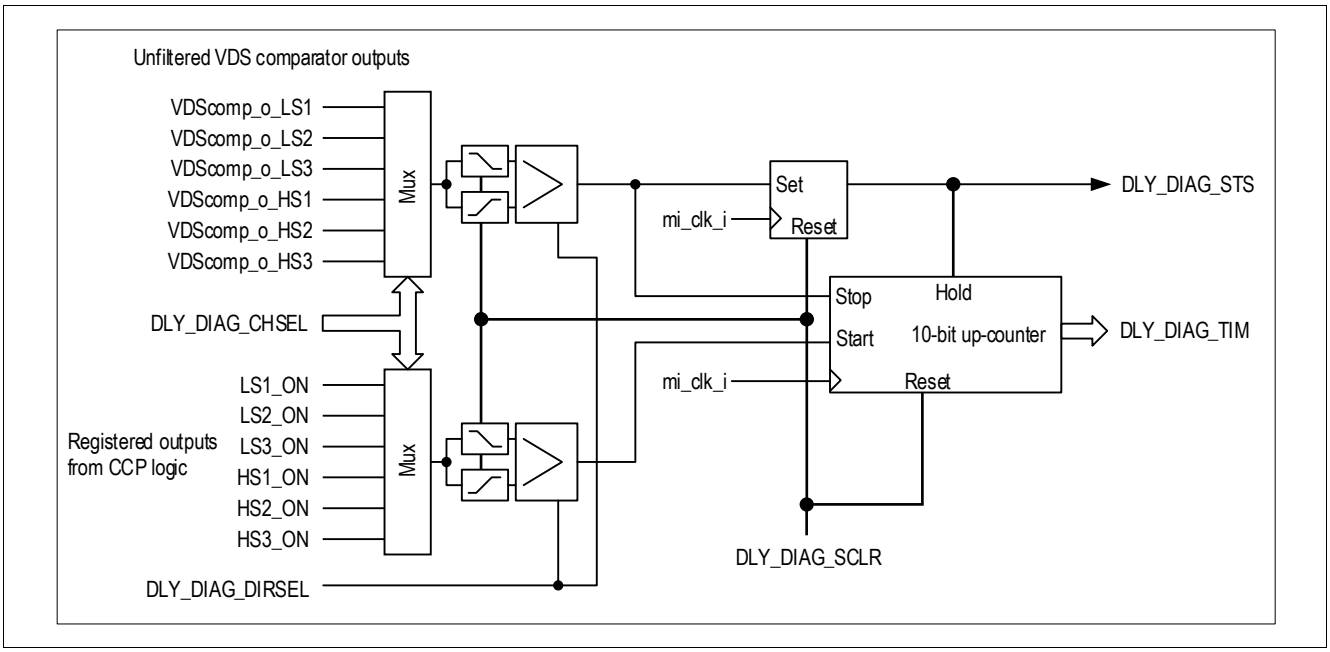


Figure 224 Principle of channel turn-on/turn-off timing measurement

Note: DLY_DIAG_TIM[9:2] in CTRL2 is mapped to the most significant 8 bits of the 10-bit delay measurement counter while DLY_DIAG_TIM[0] = DLY_DIAG_TIM[1] = DLY_DIAG_TIM[2]. For MI_CLK = 20 MHz, the effective delay measurement resolution is 200 ns.

27.4 Application Hints

27.4.1 Bridge Driver Diagnosis

27.4.1.1 Off-Diagnosis

Off-Diagnosis describes the capability to recognize a short on a motor contact to either VBAT or GND without the need to set the MOSFETs in "ON"-state. An unwanted cross-current flow through the bridge MOSFETs can be prevented that way.

27.4.1.1.1 Preparation

In order to run the Off-Diagnosis the following settings needs to be applied to the Bridge Driver registers:

A: All phases must be switched in "Enabled"-state (but not "ON"-state)

31	30	29	28	27	26	25	24
HS2_OC_DIS	HS2_OC_STS	HS2_SUPERR_STS	HS2_DS_STS	HS2_DCS_EN	HS2_ON	HS2_PWM	HS2_EN
rw	r	r	r	rw	rwhir	rwhir	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HS1_OC_DIS	HS1_OC_STS	HS1_SUPERR_STS	HS1_DS_STS	HS1_DCS_EN	HS1_ON	HS1_PWM	HS1_EN
rw	r	r	r	rw	rwhir	rwhir	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
LS2_OC_DIS	LS2_OC_STS	LS2_SUPERR_STS	LS2_DS_STS	LS2_DCS_EN	LS2_ON	LS2_PWM	LS2_EN
rw	r	r	r	rw	rwhir	rwhir	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LS1_OC_DIS	LS1_OC_STS	LS1_SUPERR_STS	LS1_DS_STS	LS1_DCS_EN	LS1_ON	LS1_PWM	LS1_EN
rw	r	r	r	rw	rwhir	rwhir	rw

Figure 225 CTRL1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_*	HS3_OC_*	HS3_SUP*	HS3_DS_*	HS3_DCS*	HS3_ON	HS3_PWM	HS3_EN	LS3_OC_*	LS3_OC_*	LS3_SUP*	LS3_DS_*	LS3_DCS*	LS3_ON	LS3_PWM	LS3_EN
rw	r	r	r	rw	rwhir	rwhir	rw	rw	r	r	r	rw	rwhir	rwhir	rw

Figure 226 CTRL2

The DS-Status Flags are still set in this state of the Bridge Driver, but since the Drain-Source-Current Sources are not enabled these flags do not have a meaning yet.

B:

The **CTRL3.IDISCHARGE_TRIM** value, which sets the current driven by the DS current sources, must be set to 1.

31	30	29 … 27	26	25 … 24	23 … 19	18 … 16
RES	RES	RES	DRV_CCP*	DRV_CCP_TIMSEL	RES	DSMONVTH
r	w	r	rw	rw	r	rw

15	14	13	12 … 8	7	6	5	4 … 0
OFF_SEQ*	IDISCHA*	RES	IDISCHARGE_TRIM	ON_SEQ_*	ICHARGE*	RES	ICHARGE_TRIM
rw	rw	r	rw	rw	rw	r	rw

Figure 227 CTRL3

C: Enable Drain-Source Current Sources on the High-Side (HSx) drivers

The DS-Current sources are internally connected to the corresponding SHx pins of the device. The "Enable"-Flags of each phase remains set. It is recommended to even enable the DS-Current source flag for the LSx drivers, just to be able to recognize the validity of the LSx_DS_STS flags inside the BDRV interrupt service routine, but it is not required for the Off-Diagnosis.

31	30	29	28	27	26	25	24
HS2_OC_DIS	HS2_OC_STS	HS2_SUPERR_STS	HS2_DS_STS	HS2_DCS_EN	HS2_ON	HS2_PWM	HS2_EN
rw	r	r	r	rw	rwhir	rwhir	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HS1_OC_DIS	HS1_OC_STS	HS1_SUPERR_STS	HS1_DS_STS	HS1_DCS_EN	HS1_ON	HS1_PWM	HS1_EN
rw	r	r	r	rw	rwhir	rwhir	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
LS2_OC_DIS	LS2_OC_STS	LS2_SUPERR_STS	LS2_DS_STS	LS2_DCS_EN	LS2_ON	LS2_PWM	LS2_EN
rw	r	r	r	rw	rwhir	rwhir	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LS1_OC_DIS	LS1_OC_STS	LS1_SUPERR_STS	LS1_DS_STS	LS1_DCS_EN	LS1_ON	LS1_PWM	LS1_EN
rw	r	r	r	rw	rwhir	rwhir	rw

Figure 228 CTRL1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_*	HS3_OC_*	HS3_SUP*	HS3_DS_*	HS3_DCS*	HS3_ON	HS3_PWM	HS3_EN	LS3_OC_*	LS3_OC_*	LS3_SUP*	LS3_DS_*	LS3_DCS*	LS3_ON	LS3_PWM	LS3_EN
rw	r	r	r	rw	rwhir	rwhir	rw	rw	r	r	r	rw	rwhir	rwhir	rw

Figure 229 CTRL2

D:

Assuming the motor contacts do not have a short to VBAT or to GND, all the DS-Status flags of each phase can be cleared now through SCUPM_BDRV_ISCLR, and should stay cleared.

31	30	29	28	27	26	25	24
HS2_OC_DIS	HS2_OC_STS	HS2_SUPERR_STS	HS2_DS_STS	HS2_DCS_EN	HS2_ON	HS2_PWM	HS2_EN
rw	r	r	r	rw	rwhir	rwhir	rw
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
HS1_OC_DIS	HS1_OC_STS	HS1_SUPERR_STS	HS1_DS_STS	HS1_DCS_EN	HS1_ON	HS1_PWM	HS1_EN
rw	r	r	r	rw	rwhir	rwhir	rw
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
LS2_OC_DIS	LS2_OC_STS	LS2_SUPERR_STS	LS2_DS_STS	LS2_DCS_EN	LS2_ON	LS2_PWM	LS2_EN
rw	r	r	r	rw	rwhir	rwhir	rw
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LS1_OC_DIS	LS1_OC_STS	LS1_SUPERR_STS	LS1_DS_STS	LS1_DCS_EN	LS1_ON	LS1_PWM	LS1_EN
rw	r	r	r	rw	rwhir	rwhir	rw

Figure 230 CTRL1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_*	HS3_OC_*	HS3_SUP*	HS3_DS_*	HS3_DCS*	HS3_ON	HS3_PWM	HS3_EN	LS3_OC_*	LS3_OC_*	LS3_SUP*	LS3_DS_*	LS3_DCS*	LS3_ON	LS3_PWM	LS3_EN
rw	r	r	r	rw	rwhir	rwhir	rw	rw	r	r	r	rw	rwhir	rwhir	rw

Figure 231 CTRL2

In case one of these flags cannot be cleared then this is already an indication of a short of a motor contact to VBAT or to GND.

27.4.1.1.2 Detection

Evaluate the state of the HSx_DS_STS flags and of the LSx_DS_STS flags in order to check if a short is present. If any HSx_DS_STS flag is set, then it means at least one of the motor contacts has a short to VBAT. If any LSx_DS_STS flag is set, then it means at least one of the motor contacts has a short to GND. Due to the fact that the motor coils do have a very low resistance usually all HSx_DS_STS flags are set together and analogue for the LSx_DS_STS flags as well. In case no motor is connected, then only the flags for the phase which sees the short are set.

To return to normal operation, disable the DS-Current sources for each phase, the value of the DS-STS flags have to be ignored from now on.

27.4.1.2 On-Diagnosis

The On-Diagnosis supervises the current flowing through the MOSFETs with the internal Drain-Source monitoring. The enabling of the integrated Drain-Source-Current source is not required for the On-Diagnosis.

27.4.1.2.1 Preparation

The bridge driver will be operated in a normal way. All the phases are enabled and those phases which are required to operate the motor in the required way are either statically "ON" or set to "PWM" (connection to CCU6). These settings are defined by the application in order to drive the motor.

The bit field **CTRL3**.DSMONVTH defines the threshold for the DS-monitoring to recognize an over-current condition in one of the MOSFETs. The DSMONVTH value defines a voltage across the drain-source path of a

voltage, the required current flowing through the MOSFET needs to be calculated based on the RDS_ON value of the MOSFET.

Example: 8mOhm MOSFET (RDS_ON=8mOhm), DSMONVTH = 0.5V

(27.1)

$$I = \frac{U}{R} = \frac{0.5V}{8mOhm} = 62.5A$$

If the current flow through the MOSFET exceeds 62.5A the DS-monitoring will recognize an over-current condition and shut off the phase, or even the entire bridge (see **CTRL1** register bits).

27.4.1.2.2 Detection

Two ways of evaluating the On-Diagnosis are possible:

1. Polling: the user application shall read the **CTRL1**/**CTRL2** registers and check the OC_STS flags of each phase to recognize whether an over-current condition has happened.
2. Interrupt based: the user shall enable the BDRV interrupt node **NVIC_ISER0**.Int_BDRV = 1; the user shall enable the HSx_OC_IE flags, along with the LSx_OC_IE flags inside the **BDRV_IRQ_CTRL** register.

Inside the BDRV Interrupt service routine the user shall check the HSx_OC_STS and LSx_OC_STS flags to see which MOSFET has detected the overcurrent condition. These flags need to be cleared before the ISR will be exited.

27.4.1.3 Open-Load Detection

The Open-Load detection can be used to check whether a motor is connected to the bridge or not.

The Open-Load detection operates similar to the Off-Diagnosis

27.4.1.3.1 Preparation

In order to run the Off-Diagnosis the following settings need to be applied to the bridge driver registers

1. All phases must be switched in "Enabled"-state (but not "ON"-state).
2. The **CTRL3.IDISCHARGE_TRIM** value, which sets the current driven by the DS current sources, must be set to 1.
3. Enable Drain-Source Current Sources on the High-Side (HSx) drivers. The DS-Current sources are internally connected to the corresponding SHx pins of the device. The "Enable"-Flags of each phase remains set. The LSx DS-Current Source flag remains disabled.
4. For a 3-phase Motor the checking of the Motor connection shall be done in three steps, for each phase separately. The HSx DS-Current Source has to be enabled always for 2 phases together, then the 3rd phase is checked whether a motor-phase is connected. This has to be iterated around until all three phases are checked. Always two phases the HSx_DCS_EN is set, while the remaining phase the HSx_DCS_EN flag is off.
5. Apply some settlement time (few µs) for the voltage at each MOSFET to stabilize.
6. Assuming the motor contacts are all connected, all the DS-Status flags of each phase can be cleared now through SCUPM_BDRV_ISCLR, and should stay cleared. If a flag cannot be cleared of a phase then the DS_STS flags remain set and will signal an unconnected motor contact.

27.5 SHx pin voltage level

The voltage levels of the SHx pins depend on the settings of the HSx_EN, LSx_EN, HSx_ON, and LSx_ON bitfields (x = 1, 2, 3) in the BDRV control registers (CTRL1 and CTRL2). For all settings, the charge-pump is assumed to be enabled (CP_CTRL_STS.CP_EN = 1):

The states of the SHx pins is determined by:

- External MOSFETs: Low- resistive connection to either GND or VDH results in strong termination.
- Internal BDRV charge circuitry: A connection to VCP which is limited to the VDH+ diode voltage (VD) by an external MOSFET's body diode results in medium termination.
- Internal gate pull-down resistor: A connection to GND results in weak termination.

Table 190 Impact of high-side settings on the SHx pins

SHx pin state	HSx_ON = 0	HSx_ON = 1
HSx_EN = 0	GND (weak termination)	GND (weak termination)
HSx_EN = 1	VDH+ VD (medium termination)	VDH (strong termination)

Table 191 Impact of low-side settings on the SHx pins

SHx pin state	LSx_ON = 0	LSx_ON = 1
LSx_EN = 0	Controlled by high-side settings	Controlled by high-side settings
LSx_EN = 1	Controlled by high-side settings	GND (strong termination)

Note: HSx_ON = 1 and LSx_ON = 1 must not be combined because doing so might lead to the destruction of the external MOSFETs.

27.6 Register Definition

The Bridge Driver registers are located in the address space below.

Table 192 Register Address Space

Module	Base Address	End Address	Note
BDRV	40034000_H	$40037FFF_H$	Bridge Driver

Table 193 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
Driver Register			
CTRL1	Half Bridge Driver Control 1	00_H	$0101\ 0101_H$
CTRL2	Half Bridge Driver Control 2	04_H	$0000\ 0000_H$
CTRL3	Half Bridge Driver Control 3	08_H	$0001\ 1111_H$
OFF_SEQ_CTRL	Turn-on Slewrate Sequencer Control	10_H	$0000\ 0000_H$
ON_SEQ_CTRL	Turn-off Slewrate Sequencer Control	14_H	$0000\ 0000_H$
Driver Trimming Register			
TRIM_DRVx	Trimming of Driver	18_H	$8000\ 0000_H$
Charge Pump Control and Status Register			
CP_CTRL_STS	Charge Pump Control and Status Register	20_H	$0000\ 0004_H$
CP_CLK_CTRL	Charge Pump Clock Control Register	24_H	$0000\ CA16_H$

The registers are addressed wordwise.

27.6.1 Driver Register

B6-Bridge (Half Bridge) Driver Control Register 1

The register is reset by RESET_TYPE_3.

Attention: ***The Bridge Driver module can only be enabled when all FET drivers are enabled in the register below.***

CTRL1

B6-Bridge Driver Control 1 **(00_H)** **Reset Value: $0101\ 0101_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HS2_OC_DIS	**HS2_OC_STS**	**HS2_SUPERR_STS**	**HS2_DS_STS**	**HS2_DCS_EN**	**HS2_ON**	**HS2_PWM**	**HS2_EN**	**HS1_OC_DIS**	**HS1_OC_STS**	**HS1_SUPERR_STS**	**HS1_DS_STS**	**HS1_DCS_EN**	**HS1_ON**	**HS1_PWM**	**HS1_EN**
rw	r	r	r	rw	rwhir	rwhir	rw	rw	r	r	r	rw	rwhir	rwhir	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LS2_OC_DIS	**LS2_OC_STS**	**LS2_SUPERR_STS**	**LS2_DS_STS**	**LS2_DCS_EN**	**LS2_ON**	**LS2_PWM**	**LS2_EN**	**LS1_OC_DIS**	**LS1_OC_STS**	**LS1_SUPERR_STS**	**LS1_DS_STS**	**LS1_DCS_EN**	**LS1_ON**	**LS1_PWM**	**LS1_EN**
rw	r	r	r	rw	rwhir	rwhir	rw	rw	r	r	r	rw	rwhir	rwhir	rw

Field	Bits	Type	Description
HS2_OC_DIS	31	rw	**High Side Driver Overcurrent Shutdown Select** 0_H **Global Shutdown**, all bridges will be shut down in case of overcurrent 1_H **Local Shutdown**, only local bridge will be shut down in case of overcurrent
HS2_OC_STS	30	r	**External High Side 2 FET Over-current Status** 0_B **no Over-current**, no over-current Condition occurred. 1_B **Over-current**, over-current occurred; switch is automatically shut down.
HS2_SUPERR_STS	29	r	**High Side Driver 2 Supply Error Status** 0_B **NORMAL**, supply is in required range. 1_B **SUPPLY ERROR**, detected; this flag is an OR of the VSD_x_STS and VCP_x_STS flags.

Field	Bits	Type	Description
HS2_DS_STS	28	r	**High Side Driver 2 Drain Source Monitoring Status in OFF-State** *Note: the status of the bit is valid only after HSx_DCS_EN bit was set to one.* 0_B **no short on external FET**, no short detected. 1_B **short on external FET detected**, short detected.
HS2_DCS_EN	27	rw	**High Side Driver 2 Diagnosis Current Source Enable** *Note: the status of the bit is valid only after LS2_DCS_EN bit was set to one.* 0_H **DISABLE**, disable current source 1_H **ENABLE**, enable current source; short diagnosis can be performed by evaluating the LSx/HSx_DS_STS Flag
HS2_ON	26	rwhir	**High Side Driver 2 On** 0_B **OFF**, Driver off 1_B **ON**, Driver on
HS2_PWM	25	rwhir	**High Side Driver 2 PWM Enable** 0_B **DISABLE**, disables control by PWM input 1_B **ENABLE**, enables control by PWM input
HS2_EN	24	rw	**High Side Driver 2 Enable** 0_B **DISABLE**, Driver circuit power off 1_B **ENABLE**, Driver circuit power on
HS1_OC_DIS	23	rw	**High Side Driver Overcurrent Shutdown Select** 0_H **Global Shutdown**, all bridges will be shut down in case of overcurrent 1_H **Local Shutdown**, only local bridge will be shut down in case of overcurrent
HS1_OC_STS	22	r	**External High Side 1 FET Over-current Status** 0_B **no Over-current**, no over-current Condition occurred. 1_B **Over-current**, over-current occurred; switch is automatically shut down.
HS1_SUPERR_STS	21	r	**High Side Driver 1 Supply Error Status** 0_B **NORMAL**, supply is in required range. 1_B **SUPPLY ERROR**, detected; this flag is an OR of the VSD_LOTH_STS and VCP_LOTH_STS for lower threshold, or, VSD_UPTH_STS and VCP_UPTH_STS for upper threshold.

Field	Bits	Type	Description
HS1_DS_STS	20	r	**High Side Driver 1 Drain Source Monitoring Status in OFF-State** *Note: the status of the bit is valid only after HSx_DCS_EN bit was set to one.* 0_B **no short on external FET**, no short detected. 1_B **short on external FET detected**, short detected.
HS1_DCS_EN	19	rw	**High Side Driver 1 Diagnosis Current Source Enable** 0_H **DISABLE**, disable current source 1_H **ENABLE**, enable current source; short diagnosis can be performed by evaluating the LSx/HSx_DS_STS Flag
HS1_ON	18	rwhir	**High Side Driver 1 On** 0_B **OFF**, Driver off 1_B **ON**, Driver on
HS1_PWM	17	rwhir	**High Side Driver 1 PWM Enable** 0_B **DISABLE**, disables control by PWM input 1_B **ENABLE**, enables control by PWM input
HS1_EN	16	rw	**High Side Driver 1 Enable** 0_B **DISABLE**, Driver circuit power off 1_B **ENABLE**, Driver circuit power on
LS2_OC_DIS	15	rw	**Low Side Driver Overcurrent Shutdown Select** 0_H **Global Shutdown**, all bridges will be shut down in case of overcurrent 1_H **Local Shutdown**, only local bridge will be shut down in case of overcurrent
LS2_OC_STS	14	r	**External Low Side 2 FET Over-current Status** 0_B **no Over-current**, no over-current Condition occurred. 1_B **Over-current**, over-current occurred; switch is automatically shut down.
LS2_SUPERR_STS	13	r	**Low Side Driver 2 Supply Error Status** 0_B **NORMAL**, supply is in required range. 1_B **SUPPLY ERROR**, detected; this flag is an OR of the VSD_LOTH_STS and VCP_LOTH_STS for lower threshold, or, VSD_UPTH_STS and VCP_UPTH_STS for upper threshold.
LS2_DS_STS	12	r	**Low Side Driver 2 Drain Source Monitoring Status in OFF-State** *Note: the status of the bit is valid only after HSx_DCS_EN bit was set to one.* 0_B **no short on external FET**, no short detected. 1_B **short on external FET detected**, short detected.

Field	Bits	Type	Description
LS2_DCS_EN	11	rw	**Low Side Driver 2 Diagnosis Current Source Enable** *Note: this bit has no effect as a low side driver diagnosis current source is not implemented.* 0_H **DISABLE**, disable current source 1_H **ENABLE**, enable current source; short diagnosis can be performed by evaluating the LSx/HSx_DS_STS Flag
LS2_ON	10	rwhir	**Low Side Driver 2 On** 0_B **OFF**, Driver off 1_B **ON**, Driver on
LS2_PWM	9	rwhir	**Low Side Driver 2 PWM Enable** 0_B **DISABLE**, disables control by PWM input 1_B **ENABLE**, enables control by PWM input
LS2_EN	8	rw	**Low Side Driver 2 Enable** 0_B **DISABLE**, Driver circuit power off 1_B **ENABLE**, Driver circuit power on
LS1_OC_DIS	7	rw	**Low Side Driver 1 Overcurrent Shutdown Select** 0_H **Global Shutdown**, all bridges will be shut down in case of overcurrent 1_H **Local Shutdown**, only local bridge will be shut down in case of overcurrent
LS1_OC_STS	6	r	**External Low Side 1 FET Over-current Status** 0_B **no Over-current**, no over-current Condition occurred. 1_B **Over-current**, over-current occurred; switch is automatically shut down.
LS1_SUPERR_STS	5	r	**Low Side Driver 1 Supply Error Status** 0_B **NORMAL**, supply is in required range. 1_B **SUPPLY ERROR**, detected; this flag is an OR of the VSD_LOTH_STS and VCP_LOTH_STS for lower threshold, or, VSD_UPTH_STS and VCP_UPTH_STS for upper threshold.
LS1_DS_STS	4	r	**Low Side Driver 1 Drain Source Monitoring Status in OFF-State** *Note: the status of the bit is valid only after HSx_DCS_EN bit was set to one.* 0_B **no short on external FET**, no short detected. 1_B **short on external FET detected**, short detected.

Field	Bits	Type	Description
LS1_DCS_EN	3	rw	**Low Side Driver 1 Diagnosis Current Source Enable** *Note: this bit has no effect as a low side driver diagnosis current source is not implemented.* 0_H **DISABLE**, disable current source 1_H **ENABLE**, enable current source; short diagnosis can be performed by evaluating the LSx/HSx_DS_STS Flag
LS1_ON	2	rwhir	**Low Side Driver 1 On** 0_B **OFF**, Driver off 1_B **ON**, Driver on
LS1_PWM	1	rwhir	**Low Side Driver 1 PWM Enable** 0_B **DISABLE**, disables control by PWM input 1_B **ENABLE**, enables control by PWM input
LS1_EN	0	rw	**Low Side Driver 1 Enable** 0_B **DISABLE**, Driver circuit power off 1_B **ENABLE**, Driver circuit power on

B6-Bridge Driver Control Register 2

The register is reset by RESET_TYPE_3.

CTRL2
B6-Bridge Driver Control 2 **(04_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DLY_DIAG_DIRSEL	DLY_DIAG_CHSEL			DLY_DIAG_STS	DLY_DIAG_SCLR	DLY_DIAG_TIM									
rw	rw			r	w	r									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HS3_OC_DIS	HS3_OC_STS	HS3_SUPER_STS	HS3_DS_STS	HS3_DCS_EN	HS3_ON	HS3_PWM	HS3_EN	LS3_OC_DIS	LS3_OC_STS	LS3_SUPER_STS	LS3_DS_STS	LS3_DCS_EN	LS3_ON	LS3_PWM	LS3_EN
rw	r	r	r	rw	rwhir	rwhir	rw	rw	r	r	r	rw	rwhir	rwhir	rw

Field	Bits	Type	Description
DLY_DIAG_DIRSEL	31	rw	**Ext. power diag timer on / off select** 0_B **TURN-OFF**, measure turn-off time 1_B **TURN-ON**, measure turn-on time
DLY_DIAG_CHSEL	30:28	rw	**Ext. power diag timer channel select** 000_B **DISABLE**, diag timer deactivated. 001_B **HB1 LS select**, measure LS1 on/off delay time. 010_B **HB2 LS select**, measure LS2 on/off delay time. 011_B **HB3 LS select**, measure LS3 on/off delay time. 100_B **DISABLE**, diag timer deactivated. 101_B **HB1 HS select**, measure HS1 on/off delay time. 110_B **HB2 HS select**, measure HS2 on/off delay time. 111_B **HB3 HS select**, measure HS3 on/off delay time.
DLY_DIAG_STS	27	r	**Ext. power diag timer valid flag** *Note: Clear flag to start a measurement.* 0_B **Diag timer invalid**, diag timer measurement ongoing 1_B **Diag timer valid**, diag timer measurement finished
DLY_DIAG_SCLR	26	w	**Ext. power diag timer valid flag clear** 0_B **Diag timer valid not clear**, 1_B **Diag timer valid clear**,
DLY_DIAG_TIM	25:16	r	**Ext. power diag timer result register**
HS3_OC_DIS	15	rw	**High Side Driver 3 Over-current Shutdown Select** 0_H **Global Shutdown**, all bridges will be shut down in case of overcurrent 1_H **Local Shutdown**, only local bridge will be shut down in case of overcurrent

Field	Bits	Type	Description
HS3_OC_STS	14	r	**External High Side 3 FET Over-current Status** 0_B **no Over-current**, no over-current Condition occurred. 1_B **Over-current**, over-current occurred; switch is automatically shut down.
HS3_SUPERR_STS	13	r	**High Side Driver 3 Supply Error Status** 0_B **NORMAL**, supply is in required range. 1_B **SUPPLY ERROR**, detected; this flag is an OR of the VSD_LOTH_STS and VCP_LOTH_STS for lower threshold, or, VSD_UPTH_STS and VCP_UPTH_STS for upper threshold.
HS3_DS_STS	12	r	**High Side Driver 3 Drain Source Monitoring Status** *Note: the status of the bit is valid only after HSx_DCS_EN bit was set to one.* 0_B **no short on external FET**, no short detected. 1_B **short on external FET detected**, short detected.
HS3_DCS_EN	11	rw	**High Side Driver 3 Diagnosis Current Source Enable** 0_H **DISABLE**, disable current source 1_H **ENABLE**, enable current source; short diagnosis can be performed by evaluating the LSx/HSx_DS_STS Flag
HS3_ON	10	rwhir	**High Side Driver 3 On** 0_B **OFF**, Driver off 1_B **ON**, Driver on
HS3_PWM	9	rwhir	**High Side Driver 3 PWM Enable** 0_B **DISABLE**, disables control by PWM input 1_B **ENABLE**, enables control by PWM input
HS3_EN	8	rw	**High Side Driver 3 Enable** 0_B **DISABLE**, Driver circuit power off 1_B **ENABLE**, Driver circuit power on
LS3_OC_DIS	7	rw	**Low Side Driver 3 Over-current Shutdown Select** 0_H **Global Shutdown**, all bridges will be shut down in case of overcurrent 1_H **Local Shutdown**, only local bridge will be shut down in case of overcurrent
LS3_OC_STS	6	r	**External Low Side 3 FET Over-current Status** 0_B **no Over-current**, no over-current Condition occurred. 1_B **Over-current**, over-current occurred; switch is automatically shut down.

Field	Bits	Type	Description
LS3_SUPERR_STS	5	r	**Low Side Driver 3 Supply Error Status** 0_B **NORMAL**, supply is in required range. 1_B **SUPPLY ERROR**, detected; this flag is an OR of the VSD_LOTH_STS and VCP_LOTH_STS for lower threshold, or, VSD_UPTH_STS and VCP_UPTH_STS for upper threshold.
LS3_DS_STS	4	r	**Low Side Driver 3 Drain Source Monitoring Status in OFF-State** *Note: the status of the bit is valid only after HSx_DCS_EN bit was set to one.* 0_B **no short on external FET**, no short detected. 1_B **short on external FET detected**, short detected.
LS3_DCS_EN	3	rw	**Low Side Driver 3 Diagnosis Current Source Enable** *Note: this bit has no effect as a low side driver diagnosis current source is not implemented.* 0_H **DISABLE**, disable current source 1_H **ENABLE**, enable current source; short diagnosis can be performed by evaluating the LSx/HSx_DS_STS Flag
LS3_ON	2	rwhir	**Low Side Driver 3 On** 0_B **OFF**, Driver off 1_B **ON**, Driver on
LS3_PWM	1	rwhir	**Low Side Driver 3 PWM Enable** 0_B **DISABLE**, disables control by PWM input 1_B **ENABLE**, enables control by PWM input
LS3_EN	0	rw	**Low Side Driver 3 Enable** 0_B **DISABLE**, Driver circuit power off 1_B **ENABLE**, Driver circuit power on

B6-Bridge Driver Control 3

The register is reset by RESET_TYPE_3.

This register is being initialized by the BootROM during the boot-up process to a value of 0003_1F1F_H

CTRL3

B6-Bridge Driver Control 3 **(08_H)** **Reset Value: $0001\ 1111_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		RES			DRV_CCP_DIS	DRV_CCP_TIMSEL		RES					DSMONVTH		
r		r			rw	rw				r			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFF_SEQ_EN	IDISCHARGEDIV2_N	RES	IDISCHARGE_TRIM					ON_SEQ_EN	ICHARGEDIV2_N	RES	ICHARGE_TRIM				
rw	rw	r	rw					rw	rw	r	rw				

Field	Bits	Type	Description
RES	31:30	r	**Reserved** Always read as 0b10
RES	29:27	r	**Reserved** Always read as 1
DRV_CCP_DIS	26	rw	**Dynamic cross conduction protection Disable** *Note: The cross conduction protection consists of two stages. This flag disables the first stage which is the activation of the delayed gate clamp after the configured cross conduction protection time. The second stage which is represented by the delayed gate clamp is still active and will be activated as soon as the opposite MOSFET within a inverter stage is activated.* 0_B **CCP Enable**, dynamic ccp is active. 1_B **CCP Disable**, dynamic ccp is disabled, delayed gate clamp remains active.
DRV_CCP_TIMSEL	25:24	rw	**Minimum cross conduction protection time setting**[1)] 00_B **0.2us**, 200ns cross conduction protection time 01_B **0.4us**, 400ns cross conduction protection time 10_B **0.8us**, 800ns cross conduction protection time 11_B **1.6us**, 1.6us cross conduction protection time
RES	23:19	r	**Reserved** Always read as 0

Field	Bits	Type	Description
DSMONVTH	18:16	rw	**Voltage Threshold for Drain-Source Monitoring of external FETs** 000_B **0.25_V**, Threshold 0 for VDS at 0.25 V 001_B **0.5_V**, Threshold 1 for VDS at 0.5 V 010_B **0.75_V**, Threshold 2 for VDS at 0.75 V 011_B **1.0_V**, Threshold 3 for VDS at 1.0 V 100_B **1.25_V**, Threshold 4 for VDS at 1.25 V 101_B **1.5_V**, Threshold 5 for VDS at 1.5 V 110_B **1.75_V**, Threshold 6 for VDS at 1.75 V 111_B **2.0_V**, Threshold 7 for VDS at 2.0 V
OFF_SEQ_EN	15	rw	**Turn-Off Slewrate Sequencer enable** 0_B **Disabled**, Turn-off Slewrate Sequencer disabled 1_B **Enabled**, Turn-off Slewrate Sequencer enabled
IDISCHARGEDIV2_N	14	rw	**Discharge current range select** *Note: For the BE, BF, and A design steps, the value for ICHARGEDIV2_N is also applied to IDISCHARGEDIV2_N.* 0_B **Half Range**, max. discharge current $I_{Sinkmax_HS}$ / 2 and $I_{Sinkmax_LS}$ / 2 1_B **Full Range**, max. discharge current $I_{Sinkmax_HS}$ and $I_{Sinkmax_LS}$
RES	13	r	**Reserved** Always read as 0
IDISCHARGE_TRIM	12:8	rw	**Trimming of the internal driver discharge current** 00000_B**HiZ**, Slew Rate Control is inactive 00001_B**min discharge current**, lowest gate discharge current selected (do not use this setting for gate driving!) *Note: This setting is for off-diagnosis and open-load detection only* 11111_B**max charge current**, see IDISCHARGEDIV2_N
ON_SEQ_EN	7	rw	**Turn On Slewrate Sequencer enable** 0_B **Disabled**, Turn-on Slewrate Sequencer disabled 1_B **Enabled**, Turn-on Slewrate Sequencer enabled
ICHARGEDIV2_N	6	rw	**Charge current range select** 0_B **Half Range**, max. charge current I_{Soumax_HS} / 2 and I_{Soumax_LS} / 2 1_B **Full Range**, max. charge current I_{Soumax_HS} and I_{Soumax_LS}
RES	5	r	**Reserved** Always read as 0

Field	Bits	Type	Description
ICHARGE_TRIM	4:0	rw	**Trimming of the internal driver charge current** 00000_B**HiZ**, Slew Rate Control is inactive 00001_B**min charge current**, lowest gate charge current selected (do not use this setting for gate driving!) 11111_B**max charge current**, see ICHARGEDIV2_N

1) If MI_CLK = 20 MHz.

Turn-off Slewrate Sequencer Control

The register is reset by RESET_TYPE_3.

OFF_SEQ_CTRL
Turn-off Slewrate Sequencer Control **(10_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRV_OFF_I_1					DRV_OFF_t_1			DRV_OFF_I_2					DRV_OFF_t_2		
rw					rw			rw					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRV_OFF_I_3					DRV_OFF_t_3			DRV_OFF_I_4					DRV_OFF_t_4		
rw					rw			rw					rw		

Field	Bits	Type	Description
DRV_OFF_I_1	31:27	rw	**Slew rate sequencer off phase 1 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_OFF_t_1	26:24	rw	**Slew rate sequencer off phase 1 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration
DRV_OFF_I_2	23:19	rw	**Slew rate sequencer off phase 2 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_OFF_t_2	18:16	rw	**Slew rate sequencer off phase 2 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration
DRV_OFF_I_3	15:11	rw	**Slew rate sequencer off phase 3 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_OFF_t_3	10:8	rw	**Slew rate sequencer off phase 3 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration
DRV_OFF_I_4	7:3	rw	**Slew rate sequencer off phase 4 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_OFF_t_4	2:0	rw	**Slew rate sequencer off phase 4 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration

Turn-on Slewrate Sequencer Control

The register is reset by RESET_TYPE_3.

ON_SEQ_CTRL
Turn-on Slewrate Sequencer Control **(14_H)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRV_ON_I_1					DRV_ON_t_1			DRV_ON_I_2					DRV_ON_t_2		
rw					rw			rw					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRV_ON_I_3					DRV_ON_t_3			DRV_ON_I_4					DRV_ON_t_4		
rw					rw			rw					rw		

Field	Bits	Type	Description
DRV_ON_I_1	31:27	rw	**Slew rate sequencer on phase 1 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_ON_t_1	26:24	rw	**Slew rate sequencer on phase 1 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration
DRV_ON_I_2	23:19	rw	**Slew rate sequencer on phase 2 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_ON_t_2	18:16	rw	**Slew rate sequencer on phase 2 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration
DRV_ON_I_3	15:11	rw	**Slew rate sequencer on phase 3 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_ON_t_3	10:8	rw	**Slew rate sequencer on phase 3 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration
DRV_ON_I_4	7:3	rw	**Slew rate sequencer on phase 4 current** 0_H **Disabled**, Slew Rate Control is disabled $1f_H$ **Maximum**, maximum output discharge current
DRV_ON_t_4	2:0	rw	**Slew rate sequencer on phase 4 time** 0_H **50ns**, 50ns phase duration 7_H **400ns**, 400ns phase duration

27.6.2 Driver Trimming Register

Trimming Driver

This register is password protected, see **PASSWD**. Writing to it is only possible if password is set. Write access without opened password scheme will cause a HardFault.

TRIM_DRVx
Trimming of Driver **(18_H)** **Reset Value: $8000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**RES**					**CPLOW_TFILT_SEL**		**HS3DRV_OCSDN_DIS**	**HS2DRV_OCSDN_DIS**	**HS1DRV_OCSDN_DIS**	**HS3DRV_FDISCHG_DIS**	**HS2DRV_FDISCHG_DIS**	**HS1DRV_FDISCHG_DIS**	**HSDRV_DS_TFILT_SEL**	
r	r					rwp2		rwp2	rwp2	rwp2	rwp2	rwp2	rwp2	rwp2	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LS3DRV_OCSDN_DIS	**LS2DRV_OCSDN_DIS**	**LS1DRV_OCSDN_DIS**	**LS3DRV_FDISCHG_DIS**	**LS2DRV_FDISCHG_DIS**	**LS1DRV_FDISCHG_DIS**	**LSDRV_DS_TFILT_SEL**		**RES**	**DRV_CCPTIMMUL**		**RES**			**LS_HS_BT_TFILT_SEL**	
rwp2	rwp2	rwp2	rwp2	rwp2	rwp2	rwp2		r	rwp2		r			rwp2	

Field	Bits	Type	Description
RES	31	r	**Reserved** Always read as 1
RES	30:26	r	**Reserved** Always read as 0
CPLOW_TFILT_SEL	25:24	rwp2	**Filter Time for Charge Pump Voltage Low Diagnosis** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 00_B **4_µs**, 4 µs filter time 01_B **8_µs**, 8 µs filter time 10_B **16_µs**, 16 µs filter time 11_B **32_µs**, 32 µs filter time
HS3DRV_OCSDN_DIS	23	rwp2	**High Side 3 Predriver in overcurrent situation disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown in overcurrent situation enable 1_B **Disable**, Predriver shutdown in overcurrent situation disable

Field	Bits	Type	Description
HS2DRV_OCSDN_DIS	22	rwp2	**High Side 2 Predriver in overcurrent situation disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown in overcurrent situation enable 1_B **Disable**, Predriver shutdown in overcurrent situation disable
HS1DRV_OCSDN_DIS	21	rwp2	**High Side 1 Predriver in overcurrent situation disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown in overcurrent situation enable 1_B **Disable**, Predriver shutdown in overcurrent situation disable
HS3DRV_FDISCHG_DIS	20	rwp2	**High Side 3 Predriver fast discharge disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown fast discharge enable 1_B **Disable**, Predriver shutdown fast discharge disable
HS2DRV_FDISCHG_DIS	19	rwp2	**High Side 2 Predriver fast discharge disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown fast discharge enable 1_B **Disable**, Predriver shutdown fast discharge disable
HS1DRV_FDISCHG_DIS	18	rwp2	**High Side 1 Predriver fast discharge disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown fast discharge enable 1_B **Disable**, Predriver shutdown fast discharge disable

Field	Bits	Type	Description
HSDRV_DS_TFILT_SEL	17:16	rwp2	**Filter Time for Drain-Source Monitoring of High Side Drivers** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 00_B **1_µs**, 1 µs filter time 01_B **2_µs**, 2 µs filter time 10_B **4_µs**, 4 µs filter time 11_B **8_µs**, 8 µs filter time
LS3DRV_OCSDN_DIS	15	rwp2	**Low Side 3 Predriver in overcurrent situation disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown in overcurrent situation enable 1_B **Disable**, Predriver shutdown in overcurrent situation disable
LS2DRV_OCSDN_DIS	14	rwp2	**Low Side 2 Predriver in overcurrent situation disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown in overcurrent situation enable 1_B **Disable**, Predriver shutdown in overcurrent situation disable
LS1DRV_OCSDN_DIS	13	rwp2	**Low Side 1 Predriver in overcurrent situation disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown in overcurrent situation enable 1_B **Disable**, Predriver shutdown in overcurrent situation disable

Field	Bits	Type	Description
LS3DRV_FDISCHG_DIS	12	rwp2	**Low Side 3 Predriver fast discharge disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown fast discharge enable 1_B **Disable**, Predriver shutdown fast discharge disable
LS2DRV_FDISCHG_DIS	11	rwp2	**Low Side 2 Predriver fast discharge disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown fast discharge enable 1_B **Disable**, Predriver shutdown fast discharge disable
LS1DRV_FDISCHG_DIS	10	rwp2	**Low Side 1 Predriver fast discharge disable** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 0_B **Enable**, Predriver shutdown fast discharge enable 1_B **Disable**, Predriver shutdown fast discharge disable
LSDRV_DS_TFILT_SEL	9:8	rwp2	**Filter Time for Drain-Source Monitoring of Low Side Drivers** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 00_B **1_µs**, 1 µs filter time 01_B **2_µs**, 2 µs filter time 10_B **4_µs**, 4 µs filter time 11_B **8_µs**, 8 µs filter time
RES	7	r	**Reserved** Always read as 0
DRV_CCPTIMMUL	6:5	rwp2	**Multiplier bits for cross conduction time settings in register DRV_CCP_TIMSEL, BF-Step only** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 00_B **MUL1**, DRV_CCP_TIMSEL value is multiplied by 1 01_B **MUL2**, DRV_CCP_TIMSEL value is multiplied by 2 10_B **MUL4**, DRV_CCP_TIMSEL value is multiplied by 4 11_B **MUL4**, DRV_CCP_TIMSEL value is multiplied by 4

Field	Bits	Type	Description
RES	4:2	r	**Reserved** Always read as 0
LS_HS_BT_TFILT_SEL	1:0	rwp2	**Blanking Time for Drain-Source Monitoring of Low / High Side Drivers** *Note: this SFR can only be written if the corresponding SCU_DM password register is written!* 00_B **1_µs**, 1 µs filter time 01_B **2_µs**, 2 µs filter time 10_B **4_µs**, 4 µs filter time 11_B **8_µs**, 8 µs filter time

27.6.3 Charge Pump Control and Status Register

Charge Pump Control and Status Register

The register is reset by RESET_TYPE_3.

CP_CTRL_STS
Charge Pump Control and Status Register **(20_H)** **Reset Value: $0000\ 0004_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				VTHVCP9V_TRIM		VCP9V_SET	CPLOPWRM_EN	VSD_UPTH_STS	DRVx_VSDUP_DIS	VSD_LOTH_STS	DRVx_VSDLO_DIS	VCP_UPTH_STS	DRVx_VCPUP_DIS	VCP_LOTH1_STS	DRVx_VCPLO_DIS
r				rw		rw	rw	r	rw	r	rw	r	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					VCP_LOWTH2			RES		VCP_LOTH2_STS	RES		CP_RDY_EN	RES	CP_EN
r					rw			r		r	r		rw	r	rw

Field	Bits	Type	Description
RES	31:28	r	**Reserved** Always read as 0
VTHVCP9V_TRIM	27:26	rw	**Charge Pump Output Voltage 9V Trimming** 00_B **7.5_V**, Threshold 0 01_B **8.0_V**, Threshold 1 10_B **8.5_V**, Threshold 2 11_B **9.0_V**, Threshold 3
VCP9V_SET	25	rw	**Charge Pump 9 V Output Voltage Set** 0_B **14V Set**, output voltage set to 14V 1_B **9V Set**, output voltage set to 9V
CPLOPWRM_EN	24	rw	**Charge Pump Low Power Mode Enable** 0_B **Low Power Mode Disable**, low power mode inactive 1_B **Low Power Mode Enable**, low power mode active
VSD_UPTH_STS	23	r	**Driver Supply MU High Status** 0_B **Driver Supply Voltage ok**, no overvoltage detected 1_B **Driver Supply Voltage too high**, overvoltage on VSD Pin detected
DRVx_VSDUP_DIS	22	rw	**Driver shutdown in case of VSD overvoltage** 0_B **Driver Disabled**, DRVx shutdown in case of VSD overvoltage 1_B **Driver Enabled**, DRVx not shutdown

Field	Bits	Type	Description
VSD_LOTH_STS	21	r	**Driver Supply MU Low Status** 0_B **Driver Supply Voltage ok**, no undervoltage detected. 1_B **Driver Supply Voltage too low**, undervoltage on VSD Pin detected.
DRVx_VSDLO_DIS	20	rw	**Driver shutdown in case VSD undervoltage** 0_B **Driver Disabled**, DRVx shutdown in case of VSD undervoltage 1_B **Driver Enabled**, DRVx not shutdown
VCP_UPTH_STS	19	r	**Charge Pump MU High Status** 0_B **Charge Pump Output Voltage ok**, no overvoltage detected 1_B **Charge Pump Output Voltage too high**, overvoltage on charge pump output detected
DRVx_VCPUP_DIS	18	rw	**Driver shutdown in case of VCP overvoltage** 0_B **Driver Disabled**, DRVx shutdown in case of VCP overvoltage 1_B **Driver Enabled**, DRVx not shutdown
VCP_LOTH1_STS	17	r	**Charge Pump MU Low Status** 0_B **Charge Pump Output Voltage ok**, no undervoltage detected. 1_B **Charge Pump Output Voltage too low**, undervoltage on chargepump output detected.
DRVx_VCPLO_DIS	16	rw	**Driver shutdown in case VCP undervoltage** 0_B **Driver Disabled**, DRVx shutdown in case of VCP undervoltage 1_B **Driver Enabled**, DRVx not shutdown
RES	15:11	r	**Reserved** Always read as 0
VCP_LOWTH2	10:8	rw	**Charge Pump Output Voltage Lower Threshold Detection Level** 000_B **7.325_V**, Threshold 0 001_B **7.654_V**, Threshold 1 010_B **7.982_V**, Threshold 2 011_B **8.309_V**, Threshold 3 100_B **8.638_V**, Threshold 4 101_B **8.966_V**, Threshold 5 110_B **9.293_V**, Threshold 6 111_B **9.620_V**, Threshold 7
RES	7:6	r	**Reserved** Always read as 0
VCP_LOTH2_STS	5	r	**Charge Pump Low Status**[1)] 0_B **Charge Pump Output Voltage ok**, no undervoltage detected. 1_B **Charge Pump Output Voltage too low**, undervoltage on chargepump output detected.

Field	Bits	Type	Description
RES	4:3	r	**Reserved** Always read as 0
CP_RDY_EN	2	rw	**Bridge Driver on Charge Pump Ready Enable** 0_B **OFF**, Bridge Driver can be enabled immediately 1_B **ON**, Bridge Driver can only be enabled when Charge Pump is ready
RES	1	r	**Reserved** Always read as 0
CP_EN	0	rw	**Charge Pump Enable** 0_B **DISABLE**, Charge Pump, circuit power off 1_B **ENABLE**, Charge Pump, circuit power on

1) VCP_LOW: a potential root cause for VCP undervoltage is an external short of a gate driver output. Reactivation of the bridge driver should only be done when the short circuit condition is removed to avoid repetitive stress of the affected output stage.

Charge Pump Clock Control Register

The register is reset by RESET_TYPE_3.

CP_CLK_CTRL
Charge Pump Clock Control Register **(24_H)** **Reset Value: 0000 $CA16_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPCLK_EN	F_CP		DITH_UPPER					RES			DITH_LOWER				
rw	rw		rw					r			rw				

Field	Bits	Type	Description
RES	31:16	r	**Reserved** Always read as 0
CPCLK_EN	15	rw	**Charge Pump Clock Enable** 0_B **DISABLE**, Charge Pump Clock is switched off and has value of 0 1_B **ENABLE**, Charge Pump Clock is running
F_CP	14:13	rw	**MSB of CP_CLK divider** CP_CLK frequency is defined by the concatenation of f_cp and dith_upper for the upper boundary during dithering, and dith_lower for the lower boundary
DITH_UPPER	12:8	rw	**CP_CLK upper frequency boundary during dithering** Must be equal or less than DITH_LOWER, see definition of F_CP
RES	7:5	r	**Reserved** Always read as 0
DITH_LOWER	4:0	rw	**CP_CLK lower frequency boundary during dithering** Must be equal or greater than DITH_UPPER, see definition of F_CP

28 Current sense amplifier

28.1 Features

Main features

- Programmable gain settings: G (see data sheet for values)
- Differential input voltage: V_{IX} (see data sheet for values)
- Wide common-mode input range: V_{CM} (see data sheet for values)
- Low setting time: T_{SET} (see data sheet for values)

28.2 Introduction

The current sense amplifier in the following figure can be used to measure near-ground differential voltages via the 10-bit ADC. Its gain is digitally programmable through internal control registers.

Linear calibration has to be applied to achieve high gain accuracy, e.g., end-of-line calibration using the shunt resistor.

The following figure shows how the current sense amplifier can be used as a low-side current sense amplifier where the motor current is converted to a voltage by means of a shunt resistor R_{SH}. A differential amplifier input is used to eliminate measurement errors caused by a voltage drop across the stray resistance R_{Stray} and differences between the external and internal grounds. If the voltage at one or both inputs is outside the operating range, the input circuit is overloaded and requires a certain specified **recovery time**.

In general, an external low-pass filter should suppress of EMI.

The CSA is able to measure positive as well as negative currents through the shunt. Since the ADC1 only provides single-ended inputs the output of the CSA will be offset by Vzero. Vzero is 0.4 * VAREF, approx. 2V. In order to measure the real CSA output at 0A input the user has to perform an ADC1 Ch1 measurement with idealy 0A through the shunt, with **CTRL**.VZERO bit set to ‘0’. The desired gain has to be set before that. The offset might vary from one gain setting to another, therefore if the gain has been changed the offset measurement has to be repeated.

The bit **CTRL**.VZERO switches between the CSA output and the Vzero voltage reference, as displayed in the following figure. The CSA is not ideal, it shows an input offset, therefore the Vzero voltage reference, if measured by the ADC1, may differ from the CSA output value at 0A current input by +/-~1mV * Gain.

28.2.1 Block diagram

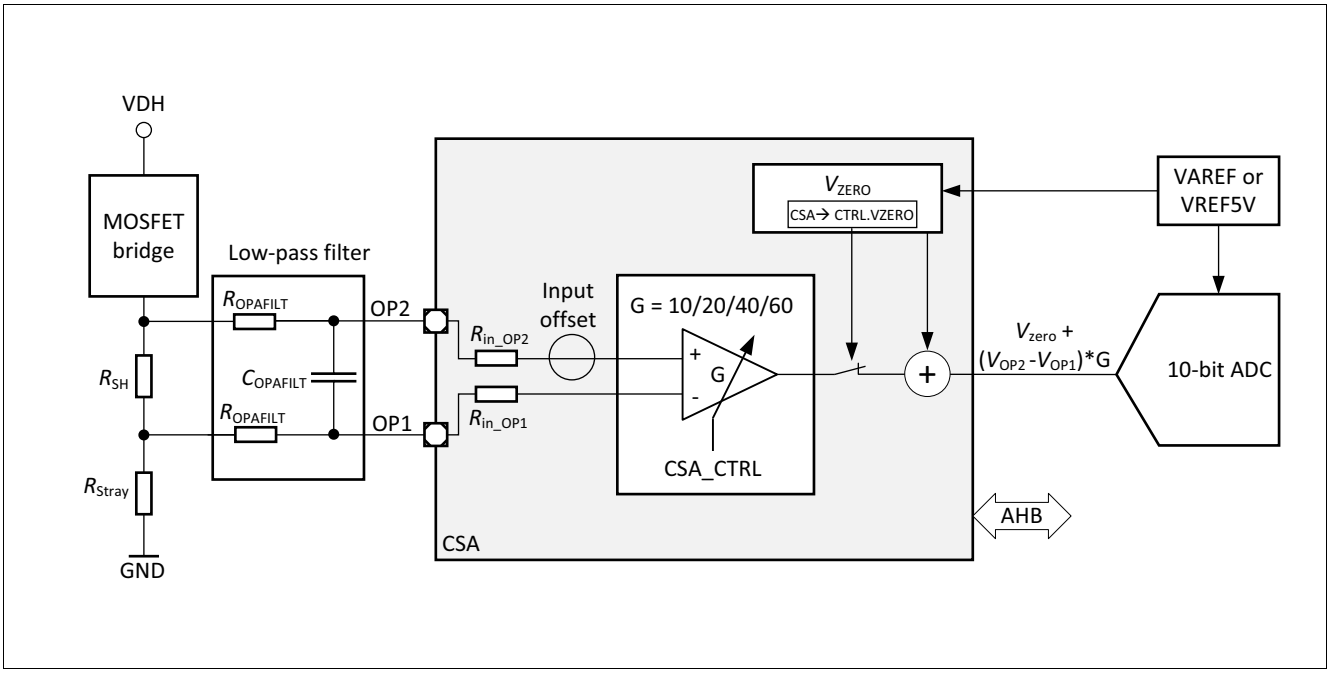


Figure 232 Current sense amplifier block diagram

28.2.2 Recommended offset calibration

In order to perform a correct offset calibration the complete path from shunt resistor through the CSA to the ADC1 should be measured. For this purpose it should be ensured that there is no current flow (0A) through the shunt. The following list collects the steps to perform a recommended offset calibration:

- connect the CSA output to the ADC1 Channel 1 input by closing the switch, CSA->**CTRL**.VZERO=0
- set desired CSA gain, CSA->**CTRL**.GAIN=x
- ensure 0A current flowing through the shunt, therefore 0V differential voltage at the CSA input
- perform an ADC1 Channel 1 measurement, using the software mode, would be suitable
- the ADC1 Channel 1 digital output, **RES_OUT1**, holds the CSA output offset including the CSA input offset
- repeat the measurement for other CSA gain settings if needed

Note: By applying the recommended calibration procedure the violated output offset is compensated.

Figure 233 displays the effective signal path for the recommended offset calibration.

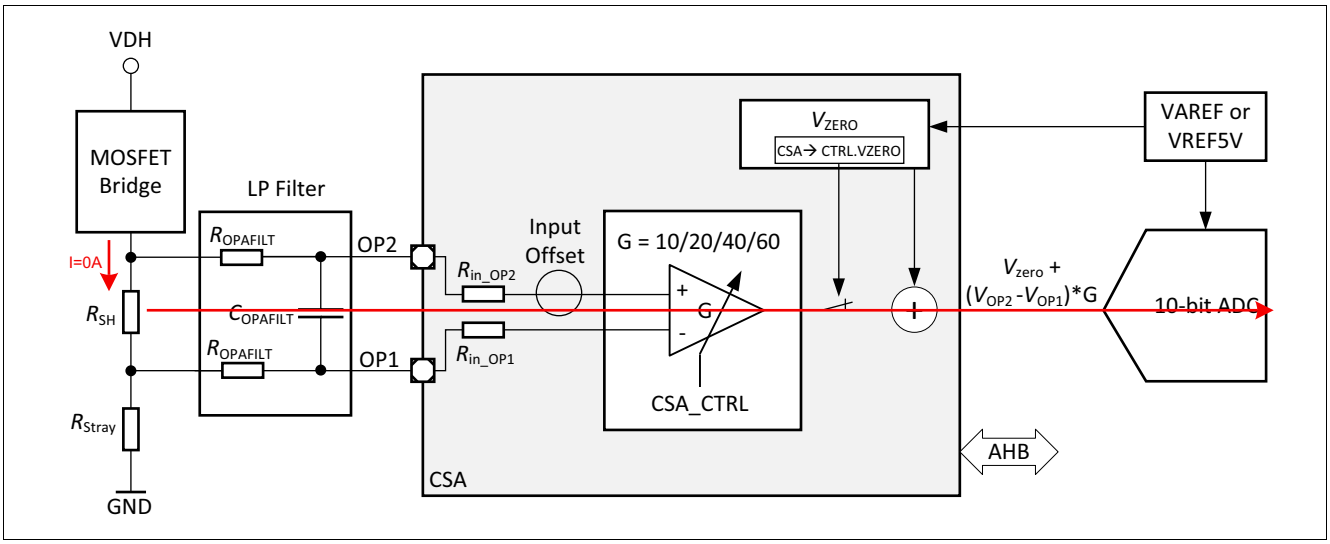


Figure 233 CSA simplified internal structure - offset calibration path

28.3 Functional Description

28.3.1 ADC Code Calculation

The differential input voltage V_{OP2} - V_{OP1} of the embedded Current Sense Amplifier (CSA) is converted to an ADC code by the following equation:

(28.1)

$$\mathrm{ADC1out} = \mathrm{floor}\left(\frac{V_{zero} + (V_{OP2} - V_{OP1}) * G}{V_{LSB}}\right)$$

wherein the parameter V_{OP1} *and* V_{OP2} are the voltages at the inputs of the amplifier and G is the configured gain.

The CSA output voltage V_{CSAout} = (V_{OP2} - V_{OP1}) * G is centered around an offset voltage V_{Zero} which has the following dependency on the reference voltage V_{AREF} of ADC1:

(28.2)

$$V_{Zero} = 0.4 * V_{AREF}$$

The typ. value of V_{AREF} is 5V @ 27 °C. V_{LSB} defined as follows:

(28.3)

$$V_{LSB} = \frac{V_{AREF}}{1023}$$

28.4 Register Definition

The next chapter lists the configuration possibilities of the Current Sense Amplifier (CSA) which can be used for external current sensing.

Table 194 Amplifier Module Base Address List

Module	Base Address
CSA	48018000_H

The base address of the module is the same as for the measurement unit (MU) as the current sense amplifier is a sub-block of the MU.

Table 195 Register Overview

Register Short Name	Register Long Name	Offset Address	Reset Value
CTRL	Operational Amplifier Control and Status	$0C_H$	$0000\ 0000_H$

The registers are addressed wordwise.

Operational Amplifier Control and Status

The following register consists of control and status bits. This Register is cleared by every reset.

The register is reset by RESET_TYPE_3.

CTRL

Operational Amplifier Control and Status **($0C_H$)** **Reset Value: $0000\ 0000_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES							VZERO	RES					GAIN		EN
r							rw	r					rw		rw

Field	Bits	Type	Description
RES	31:9	r	**Reserved** Always read as 0
VZERO	8	rw	**Current Sense Output Selection** 0_B **VOUT**, CSA output connected to ADC1 Ch1 1_B **VZERO**, voltage reference connected to ADC1 Ch1
RES	7:3	r	**Reserved** Always read as 0
GAIN	2:1	rw	**Operational Amplifier Gain Setting** 00_B **10**, Gain Factor 10 01_B **20**, Gain Factor 20 10_B **40**, Gain Factor 40 11_B **60**, Gain Factor 60
EN	0	rw	**OPA Enable** 0_B **DISABLE**, OPA switched off 1_B **ENABLE**, OPA switched on

29 Revision History

Revision	Date	Changes
Rev. 1.81	2023-07-10	
System control unit – digital modules (SCU-DM)		• Editorial changes
System control unit – power modules (SCU-PM)		• Editorial changes
Rev. 1.8	2022-03-08	
Front cover		• Editorial change in table “Grade 1 derivatives, Product name”: Changed the order of “TLE9872QXA40” and “TLE9872-2QXA40”.
Overview and Device register types		• Editorial changes
Power Management Unit (PMU)		• Editorial changes • Editorial change in chapter “Data storage registers”: Changed the reset values from “0000 0000_H” to “00_H”.
System control unit – digital modules (SCU-DM)		• Editorial change: Deleted “In the case of watchdog resets, the requestor can be identified via the reset indicator bits WDT1RST and WDTRST.”
Memory control unit		• Features: Changed “Memory protection functions for all system memory types (D-flash, P-flash, RAM)” to “Memory protection for D-flash and P-flash system memory”. • Chapter "Memory Protection Unit (MPU)”: Updated the description of the MPU.
High-speed synchronous serial interface (SSC1/SSC2)		• Chapter “Error Detection Mechanisms: Added figure “Example of an SSC phase error”.
Measurement unit		• Added details of the ADC1 reference voltage (V_{AREF}) – Table “Measurement functions and associated modules”, Module name: Added “Reference voltage (V_{AREF})”. – Added chapter “ADC1 reference voltage (V_{AREF})”.
14-Bit Sigma Delta ADC (ADC3 / ADC4)		• Removed chapter “Interpretation of SDADC output code”.

Revision	Date	Changes
Bridge driver (incl. charge pump)		• Chapter “Channel turn on/off delay measurement”: Added: "Note: DLY_DIAG_TIM[9:2] in CTRL2 is mapped to the most significant 8 bits of the 10-bit delay measurement counter value while DLY_DIAG_TIM[0] = DLY_DIAG_TIM[1] = DLY_DIAG_TIM[2]. For MI_CLK = 20 MHz the effective delay measurement resolution is 200 ns." • Chapter “Bridge Driver Diagnosis, Preparation, B”: Changed from "For a detection of a motor, CTRL3.DSMONVTH shall be set to the maximum value. The CTRL3.IDISCHARGE_TRIM value, which sets the current driven by the DS-Current sources, shall be set to a very small value. A value of 1 would be sufficient." to "The CTRL3.IDISCHARGE_TRIM value, which sets the current driven by the DS current sources, must be set to 1." • Chapter "Preparation, C, Enable Drain-Source Current Sources on the High-Side (HSx) drivers": Changed figure "CTRL2". • Chapter “Bridge Driver Diagnosis”: Changed from "all the DS-Status flags of each phase can be cleared now, and shall stay cleared." to “all the DS-Status flags of each phase can be cleared now through SCUPM_BDRV_ISCLR, and should stay cleared.” • Added chapter “SHx pin voltage level". • Register CP_CLK_CTRL, changed descriptions from – DITH_UPPER: Changed from rom "legal values are equal or greater than DITH_LOWER, see definition of f_cp" to "Must be equal or less than DITH_LOWER, see definition of F_CP" – DITH_LOWER: Changed from "legal values are equal or less than DITH_UPPER, see definition of f_cp" to "Must be equal or greater than DITH_UPPER, see definition of F_CP"

Revision History

Revision	Date	Changes
Rev. 1.7	2022-05-11	
Front cover		• Title: Added the A and UH steps • Tables: Added orderable part number (OPN), design step, and package
Overview		• Table “Acronyms”: Added OPN (orderable part number)
Power Management Unit (PMU)		• Sleep mode: Changed “Cyclic Sense” to “Cyclic Wake mode”. • Figure "Configuration Flow of cyclic sense mode": Changed "Sleep Mode" to "Stop mode" • CNF_RST_TFB: Updated description, Added “BE, BF, and A design steps”
System control unit – power modules (SCU-PM)		• System failures handled by SCU_PM: Deleted "automatic shutdown of power modules in case of VS Overvoltage" • VSD-Overvoltage System Shutdown: Deleted "LIN" • Register “ID”: Changed Reset Value from 82_H to $8x_H$
DMA controller		• Deleted “BITS Definition”.
Measurement unit		• Functional description: Added equations • Changed “build out of” to “consists of”
High-voltage monitor input		• Register “CNF”, changed: – “CYC” to “RES” – Description from “MON for Cycle Sense Enable, Works only if MON1_EN is enabled, 0B DISABLED, Cycle Sense disabled, 1B ENABLED, Cycle Sense enabled” to “Reserved, Default value: 0B DISABLED, Reading or writing this bit has no effect.” – Deleted footnote
Bridge driver		• Chapter “Preparation”, changed: From "For a detection of a motor, CTRL3.DSMONVTH shall be set to the maximum value. The CTRL3.IDISCHARGE_TRIM value, which sets the current driven by the DS-Current sources, shall be set to a very small value. A value of 1 would be sufficient." to "The CTRL3.IDISCHARGE_TRIM value. which sets the current driven by the DS current sources, must be set to 1." • Register “CTRL3”, field "IDISCHARGEDIV2_N", added note: "For the BE, BF, and A design steps, the value for ICHARGEDIV2_N is also applied to IDISCHARGEDIV2_N."

Revision History

Revision	Date	Changes
Rev. 1.6	2021-06-18	
All		• Changed “variant” to “product”. • "Reserved bits" and "bit fields”: Changed to consistent spelling: "RES". • "Timer X": Changed to consistent spelling: "TimerX".
Overview		• TLE987x: Summary of Features: "40 MHz clock frequency" changed to "**Up to** 40 MHz clock frequency". • Abbreviations: MCU acronym explanation in the glossary: Changed “**Memory Control** Unit” to “**Microcontroller** Unit”. • Abbreviations, added: “rfu: Reserved for future use.” • Changed "cyclic wake-up is activated with 1 s of **wake** time" to "cyclic wake-up is activated with 1 s of **wait** time". • Changed from “Followed by the Sleep and Stop Mode command“ to “Followed by the Sleep Mode command“. • Changed from “Additionally, the device is in Cyclic Wake-Up mode with the maximum configurable dead time setting” to “Additionally, the device is in Cyclic Wake-Up mode with the configurable dead time setting”. • Changed “Module/function: LP_CLK2 (100 kHz), Active mode: **ON**” to “Module/function: LP_CLK2 (100 kHz), Active mode: **ON/OFF**” • Added “GPIOs”: The wake-up can be triggered by rising, falling, or both signal edges for the monitor input, **GPIOs**, by LIN, or by cyclic wake-up.
Power Management Unit (PMU)		• Added: Figure PWM-wake: “Power Management Unit System Modes” and footnote “Product dependent. See the datasheet for details.” • Replaced figure “Sleep Mode Entry Timing” and figure “Sleep Mode LIN Wake-Up Timing”. • Added chapter “Low-VS operation”. • Deleted in “VDDEXT Control Register”: “(range 1 is VDDEXT ± 220 mV, VDDEXT_OK according to P_2.3.5)”. • Wake-Up Management Unit (WMU), Functional description, Wake-Up over MON Pins: Removed “Cyclic sense”: "This setup can be used to wake-up the device from Stop Mode, but also a wake-up from Sleep Mode." • Changed “to be switched **of**” to “to be switched **off**”
System Control Unit - Digital Modules (SCU-DM)		• Figure “System control unit”: Changed from "INTISR<**9**:0>" to "INTISR<**15**:0>". • “Interrupt Vector Table”, Service Request: Changed: “MU- ADC8/T3” to “MU- ADC2/T3”. • Register “DMAIEN2”: Discrepancy in the table figure “SDADCIE” adjusted. “r” changed to “rw”, according to the register table.
Arm® Cortex®-M3 core		• ISRPENDING: deleted "maskable". • Register “AFSR:”: Discrepancy in the table figure “AUXFAULT” adjusted. “rw” changed to “r”, according to the register table.
DMA		• Changed “Table x shows the <Module> module” to “Table x shows the DMA module”.

Revision History

Revision	Date	Changes
Memory Control Unit		• Deleted “MCU“: Changed “The memory control unit (MCU) is divided into the following submodules.” to “The memory control unit is divided into the following submodules.”
Interrupt System		• Changed “Before enabling an interrupt, all corresponding interrupt status flags **should** be cleared” to “Before enabling an interrupt, all corresponding interrupt status flags **must** be cleared”. • Table "Interrupt vector table", Service request: "Changed MU- ADC**8**/T3" to "MU-ADC**2**/T3". • Register “DMAIRC2CLR:”: Discrepancy in table figure “SDADCC” adjusted. “r” changed to “w”, according to the register table.
GPIO Ports and Peripheral I/O		• Register “P2_DIR”: Discrepancy in the register table “” adjusted. “rh” changed to “rw”, according to the table figure. • Register “P2_PUDSEL”, “P2_PUDEN”: Discrepancy in the register table adjusted. “rh” changed to “rw”, according to the table figure.
General-purpose timer units (GPT12)		• Register "PISEL", "IST5IN", “1B”: Changed from “Signal T5INB is selected” to “Reserved”
Timer2 and Timer21		• Register “T2H:”: Discrepancy in the table “T2H” adjusted. “rw” changed to “rwh”, according to the register figure.
Capture/compare unit 6 (CCU6)		• Changed in table “Multi-Input Capture Mode”: “11111” to “1111”.
UART1/UART2		• Table “Connection options”: Changed “UART1.**T**x” to “UART1.**R**x”. Added “(UART1.Rx)
LIN Transceiver		• Figure “LIN transceiver block diagram”: Changed the text at the LP filter output to: "RxD_1 to UART and TIMER2 pin T2EX (T2EXCON=0, T2EXIS=0)”
Measurement unit		• Port 2 ADC Selection Control Register: Added "for products with CSA. In the product without CSA the reset value is $0000\ 0000_H$." P2_ADCSEL_CTRL: Changed reset value from "$0000\ 000_H$" to "$0000\ 0400_H$" accordingly.
10-bit analog-to-digital converter (ADC1)		• Deleted “8-bit”: Changed “8-bit and 10-bit resolution” to “10-bit resolution”. • Features: Up to 8 analog input channels. Deleted: “(channel 7 reserved for future use).”

Revision History

Revision	Date	Changes
High-voltage monitor input		• Introduction: Changed "The input is sensitive to a input level monitoring, this is available when the module is switched to active mode with the SFR bit EN." to "The input level can be monitored if the module is enabled (PMU_MON_CNF)." • Deleted “As the system provides the functionality of cyclic sense, the IN can be configured as a wake-up source for this mode. This is done by setting the bit CYC.”
Bridge Driver (incl. Charge Pump)		• Deleted figures "BRDV_IRQ_CTRL" and “BDRV_ISCLR” • Figure "Driver Module Block Diagram": Added an asterisk into the High and Low side Drivers blocks and related note "The VGSx limiter and fast discharge functions are implemented in this block.". • Replaced "this flag is an OR of the VDS_x_STS and VCP_x_STS flags" by "this flag is an OR of the VSD_LOTH_STS and VCP_LOTH_STS for lower threshold, or, VSD_UPTH_STS and VCP_UPTH_STS for upper threshold."

Revision	Date	Changes
Rev. 1.5.3	2020-10-07	Table 2 supported device list updated
		Chapter 6.3.3.6 Switching PLL Parameters list updated
		Table 17 OSCDISC setting for Prescaler Mode corrected
		Chapter 6.3.3.2 OSCDISC settings in Prescaler Mode corrected
Rev. 1.5.2	2020-01-28	Table 1, Table 2 updated: TLE9872QXA40, TLE9872-2QXA40, TLE9872QXW40, TLE9872-2QXW40, TLE9872QTW40, TLE9872-2QTW40 added
		References to flash size extended to up to 256 KByte
		References to RAM size extened to up to 8 KByte
Rev. 1.5.1	2020-01-28	Table 2 updated, TLE9877QTW40 and TLE9879QTW40 added
Z8F80155847 Rev. 1.5	2020-01-28	Introduction: Added registered trademarks: Arm® Cortex®, changed Cortex to Arm® Cortex®
		SDADC marked to be variant dependent, check datasheet for availability
		XTAL support marked to be variant dependent, check datasheet for availability
		Table 1, Table 2 updated, temperature range and package removed
		Summary of features: temperature range, package and qualification information removed
		Table 4 updated
		Table 5 updated
		Figure 2 updated
		Chapter 5.3.3, physical values replaced by electrical characteristics symbols
		PMU_SUPPLY_STS reset types updated
		Chapter 5.3.5.2, physical values replaced by electrical characteristics symbol
		VDDEXT_CTRL reset types updated
		Figure 15, Cyclic Sense Mode flow updated
		Figure 19 updated, f_{OSC} to f_{OSC_int}, footnote for XTAL variant dependency

Revision History

Revision	Date	Changes
		Table 15 headline updated, reset value to page number
		Chapter 6.3.3.6 updated
		Figure 23 added for variants without XTAL
		Figure 25, clock tree, updated, BGCLK added
		Figure 26 updated
		OSC_CON.Bit5 description updated (erratum id 0000057330-3)
		APCLK_CTRL1.PLL_LOCK bit description updated
		Chapter 6.4.1, Wake-up reset description updated
		Chapter 6.4.2, warm reset description added
		RSTCON.LOCKUP_EN access type updated
		Table 35, NMI vector updated
		former Chapter “Implementation of DMA Request inside the SCU” removed
		P0_POCON0.PDM1 (P0.1) strong driver options added
		P0_POCON1.PDM3 (P0.3) strong driver options added
		TCCR.TCC bitfield description updated
		NVM_PROT_STS bitfield descriptions updated
		MEM_ACC_STS register updated
		ID.VERID bit description updated
		MEMSTAT.SASTATUS bit description updated
		MEMSTAT.SECTORINFO description updated
		Figure 34 updated
		Chapter 7.5.2 system overtemperature description updated, fail power statge disable description added
		PCU_CTRL_STS reset value and read value of reserved bits updated SYS_OT_PS_DIS and FAIL_PS_DIS description updated
		SHPR2 register address updated (erratum id 0000057330-2)
		SYSTICK_CS bitfield description updated
		SYSTICK_CAL bitfield shortnames added
		NVIC_IPR0, NVIC_IPR1, NVIC_IPR2, NVIC_IPR3 bitfields updated
		ICSR, AIRCR, SCR, CCR, SHCSR, CFSR, HFSR, DFSR, AFSR updated, shortnames added
		Figure 39 DMA transfer request sources added
		STATUS, CHNL_SW_REQUEST, CHNL_USEBURST_SET, CHNL_USEBURST_CLR, CHNL_REQ_MASK_SET, CHNL_REQ_MASK_CLR, CHNL_ENABLE_SET, CHNL_ENABLE_CLR, CHNL_PRI_ALT_SET, CHNL_PRI_ALT_CLR, CHNL_PRIORITY_SET, CHNL_PRIORITY_CLR, ERR_CLR updated, bitfield short names added
		Note on Page 284 added
		Table 77 headline Description updated to Page Number
		Table 80, Table 81 headline Reset Value updated to Page Number

Revision History

Revision	Date	Changes
		P2_DATA, P2_DIR, P2_PUDSEL, P2_PUDEN bitfield updated
		Table 89 P2.3 INP2 (CTRAP) updated (erratum id 0000057330-1)
		Figure 78, Figure 79 updated, basic clock removed
		Figure 109 updated
		Table 110 updated
		T5CON.T5UD note corrected
		PISEL note removed
		Table 113 headline Reset Value updated to Page Number
		Table 135 headline Reset Value updated to Page Number
		Figure 161 updated
		Figure 174 UART1 interface added
		Table 141 UART1 connection options added
		Term UART1 doesn't support Mode0 removed
		SCON, Mode2: $f_{PCLK}/32$ removed
		Table 145, Table 146 headline Reset Value updated to Page Number
		Chapter 20.3.1 LIN modes description updated
		Figure 185 updated
		Table 155 headline Reset Value updated to Page Number
		former Figure 185 removed
		Figure 187 updated
		Chapter 22.5 description updated, Figure 190, Table 160 added
		VMON_SEN_CTRL.VMON_SEN_HRESO_5V description updated
		CTRL2.SEL_TS_COUNT description updated
		Chapter 23.5.1.2 calibration time added
		Chapter 23.6.1.1 IIR filter propagation delay added
		Chapter 23.7.1 hysteresis/counter description added, Figure 203, Figure 204 added
		FILT_UP_CTRL, FILT_LO_CTRL, TH0_3_UPPER, TH4_5_UPPER, TH6_9_UPPER, CNT0_3_UPPER, CNT4_5_UPPER, CNT6_9_UPPER, TH0_3_LOWER, TH4_5_LOWER, TH6_9_LOWER, CNT0_3_LOWER, CNT4_5_LOWER, CNT6_9_LOWER bitfield shortnames added
		Table 174, Table 175 hysteresis and counter values updated
		Chapter 24.3.1 EIM/ESM description updated, Figure 206, Figure 207 added
		GLOBCTR.DIVA description updated
		CHx_EIM.TRIG_SEL description updated
		CHx_EIM.REP description updated
		CHx_ESM.TRIG_SEL description updated
		Chapter 25.1 clock source information added
		CTRL_STS updated

Revision History

Revision	Date	Changes
		CNF bitfield shortnames added
		former Chapter 27.3.1 removed
		Chapter 27.3.3 added
		Figure 221 updated
		CTRL3 bitfield description update
		CP_CTRL_STS.DRVx_VSDUP_DIS description updated
		CP_CTRL_STS.DRVx_VSDLO_DIS description updated
		CP_CTRL_STS.DRVx_VCPUP_DIS description updated
		CP_CTRL_STS.DRVx_VCPLO_DIS description updated
Rev. 1.4	2018-08-03	Added: Introduction “About this document”, Listing of product variants Grade 1 and Grade 0 derivates
		PMU_VDDEXT_CTRL Register, reset types updated
		PMU_SUPPLY_STS Register, reset types updated
		SCU_MEM_ACC_STS Register, reset types updated
		SCU_APCLK_CTRL1.PLL_LOCK bit encoding corrected
		TLE9879-2QXA40, including SDADC (ADC3/4) added
		Chapter “Block diagram” removed, it can be found in the datasheet
		Chapter “Device Pinout and Pin configuration” removed, it can be found in the data sheet
		Chapter 5, editorial changes
		Application diagram removed, it can be found in the data sheet
		Figure 7, System Overtemperature added to transition from active to sleep
		Chapter 4.2.2 - Stop Mode, number values replaced by electrical characteristics symbol
		Chapter 4.3.2, number values replaced by electrical characteristics symbols
		Chapter 4.3.3, hint for ADC3/4 usage added
		Chapter 4.4.1.1, SUPP_TMOUT replaced by SUPP_SHORT
		Table 13, CNF_CYC_WAKE reset value updated
		Figure 28, naming of reset sources updated, following list updated
		Chapter 4.7, added
		Figure 96, Footnote added to XTAL1, XTAL2
		Figure 100, Footnote added to XTAL1, XTAL2, register naming updated
		Figure 102, XTAL frequencies replaced by EC symbol, output renamed to f_{OSC_int}
		Chapter 10.3.3.1, electrical parameters replaced by symbols
		Chapter 10.3.3, f_{OSC} as output of the HP_OSC replaced by f_{OSC_int}
		Chapter 10.3.3.7, editorial changes
		Figure 113, Clock tree updated, SDADC added
		SCU_OSC_CON, reset value set by bootrom added
		SCU_OSC_CON.OSC2L, f_{OSC} replaced by f_{OSC_int}

Revision History

Revision	Date	Changes
		SCU_SYSCON0, reset value set by bootrom added
		SCU_SYSCON0.SYSCLKSEL, f_{OSC} replaced by f_{OSC_int}
		SCU_APCLK_CTRL1, reset value set by bootrom added
		SCU_APCLK_CTRL1.APCLK_SET, SDADCCLK_DIV included
		SCU_APCLK_CTRL2.SDADCCLK_DIV bit field added
		SCU_APCLK1, reset value set by bootrom added
		SCU_APCLK2, reset value set by bootrom added
		Chapter 10.4.1, editorial changes
		Chapter 10.4.2, editorial changes
		Chapter 10.4.4.2, editorial changes
		Table 54, Power Down replaced by Stop Mode
		Table 55, naming of reset sources updated
		Chapter 10.5.2.2, 1.5V replaced by V_{DDCOUT}, 0.9V replaced by $V_{DDCOUT_Stop_Red}$
		SCU_PMCON0, bit field description update, Power Down relaced by Stop Mode
		Table 61, MU-ADC8/T3, SDADC added to node 2
		SCU_MODIEN1, reset value set by bootrom added
		SCU_MODIEN2, reset value set by bootrom added
		SCU_GPT12IEN, reset value set by bootrom added
		Chapter 10.7.3.2, SDADC added
		SCU_DMAIEN2.SDADCIE bit field added
		Chapter 10.7.5, FNMIOCDS replaced by FNMIOT, WDT1RST renamed to WDT1 reset
		SCU_MODPISEL1.T21EXCON/.T2EXCON description updated
		SCU_MODPISEL2, related port pins added to the description
		SCU_MODPISEL3, related port pins added to the description
		SCU_NVM_PROT_STS, reset value set by bootrom added
		Chapter 10.18, added
		SCUPM_AMCLK_FREQ_STS, frequency calculation added to description
		SCUPM_AMCLK_TH_HYS, frequency calculation added to description
		Figure 6, German-style logic symbol replaced by named logic symbol
		SCUPM_SYS_IS, SCUPM_SYS_ISCLR, SCUPM_SYS_IRQ_CTRL, SDADC added (ADC3/4)
		Chapter 4.5.1, editorial changes, figure removed
		Chapter 4.5.2, 150°C replaced by symbol $T_{j,max}$
		SCUPM_PCU_CTRL_STS, reset value set by bootrom added
		Chapter 4.7, added
		Table 8 lists abriviations for Figure 5
		Table 14, ADC34 (SDADC) added, PPORTS replaced by PORT
		Chapter 4.3, number values replaced by electrical characteristics symbols

Revision History

Revision	Date	Changes
		Chapter 4.3.2, number values replaced by electrical characteristics symbols
		Chapter 4.3.2.6, number values replaced by electrical characteristics symbols
		Table 8, SDADC added to Node 2
		Chapter 4.3.1, German-style logic symbol replaced by named logic symbol in all figures
		Table 10, ADC3/4 (SDADC) added to Node 2
		Chapter 4.3.2, Interrupt Structure 1 heading removed
		Table 12, SDADC added to Node 2
		Figure 7, SDADC added to Node 2
		Figure 28, DMA Channel 13 - SDADC added
		EXICON0, reset value set by bootrom added
		DMAIRC2, DMAIRC2CLR, SDADC added
		Chapter 4.1, Low Power Mode replaced by Stop Mode
		SCUPM_WDT1_TRIG, footnote updated
		Figure 5, DIR register added to figure
		Table 20, SDADC added to P2.0, P2.2, P2.4, P2.5
		PORT_P2_DATA, bit 7 and bit 1 updated to be reserved
		PORT_P2_DIR, bit 7 and bit 1 updated to be reserved
		PORT_P2_PUDSEL, bit 7 and bit 1 updated to be reserved
		PORT_P2_PUDEN, bit 7 and bit 1 updated to be reserved
		PORT_P2_DIR, reset value set by bootrom added
		Chapter 4, German-style logic symbol replaced by named logic symbol in all figures
		Figure 10, updated figure
		TIMER3_T3_TRIGG_CTRL, bitfield T3_TRIGG_INP_SEL updated for Mode 3b
		Figure 34, updated figure
		Figure 35, figure title updated
		Figure 3, pull up resistor value has been replaced by electrical characteristics symbol
		LIN_LIN_CTRL_STS, reset value set by bootrom added
		Chapter 4.5, added
		Figure 4, attenuator values and reference voltage values replaced by the electrical characteristics symbol
		Figure 4, added for TLE9879-2QXA40
		Table 10, MF_P2_ADCSEL_CTRL register added to the table
		Chapter 4.3.2, V_{BG} value updated on the formulars
		Former chapter 23.3.3 removed
		Chapter 4.4.2 added
		Chapter 4.5.1, attenuator values and input resistance replaced by the electrical characteristics symbol

Revision History

Revision	Date	Changes
		Figure 18, physical values replaced by electrical characteristics symbol
		MF_VMON_SEN_CTRL.VMON_SEN_SEL bit description updated
		MF_TRIM_BEMFx, register description updated
		Chapter 4.8.1, V_{BG} value updated on the formulars
		Figure 26 updated
		MF_REF1_STS, reset value set by bootrom added
		ADC2_ADC2_CTRL_STS.VS_RANGE bit description updated
		Chapter 4.4.1, conversion time calculation updated
		Figure 9 updated
		ADC2_MEAS_ADC2_CTRL1 bits 11..8, always read as 0 removed
		Table 19, footnotes updated
		Table 20, footnotes updated
		Figure 11, German-style logic symbol replaced by named logic symbol
		Chapter 4, chapter added for TLE9879-2QXA40 only
		Chapter 1.1, physical values replaced by electrical characteristics symbols
		Figure 4, German-style logic symbol replaced by named logic symbol
		Figure 11, updated
		DRV_DRV_CTRL3, bit31 and bit30 set to reserved
		DRV_CP_CTRL_STS.DRVx_VSDUP_DIS description updated
		DRV_CP_CTRL_STS.DRVx_VSDLO_DIS description updated
		DRV_CP_CTRL_STS.DRVx_VCPUP_DIS description updated
		DRV_CP_CTRL_STS.DRVx_VCPLO_DIS description updated
		Chapter 4.1, physical values replaced by electrical characteristics symbols
		“Application Information” chapter removed, it can be found in the datasheet
1.3	2017-06-27	TLE987x product variants chapter added
		SYS_IS description for bit LIN_TMOUT_IS corrected
		In Table 158 the settings for channels 7..9 updated
		Table 140 was added to support proper synch break recognition
		ADC2.FILT_OUT0 and following, description updated
		SFRs Associated with Timer Block GPT1 updated
		SFRs Associated with Timer Block GPT2 updated
		Interrupt Node Assignment all figures updated
		Interrupt Flags Overview table updated
		Table 172, ADC2 default upper/lower thresholds updated (Grade1)
		Table 173, ADC2 default upper/lower thresholds updated (Grade0)
		SCON2 register removed, all information available in SCON register
		CSA Chapter 28.2 updated
		Figure 227 added, simplified internal CSA diagram
		ADC2 Figure 205 register naming updated

Revision History

Revision	Date	Changes
		PMU Sleep Mode Entry Timing figure updated
		PMU Sleep Mode LIN Wake-Up Timing figure updated
		PMU Stop Mode Entry Timing figure updated
		PMU VDDP figure updated
		PMU VDDC figure updated
		PMU Stop Mode Exit Timing figured updated
		PMU Timing Diagram for Cyclic Sense figure updated
		SCU PLL Block Diagram updated
		SCU Clock Tree, DMA added
		SCUPM Block diagram of CGU including Clock Watchdogs updated
		Information about Re-enabling of VAREF updated
		ADC1 Trigger Selection GPT12.T6 selection corrected
		GPT12E Table 97 for Capture Mode added
		P1_OD register population updated
		Alternate Function Map for GPIOs added
		DMA Registers, descriptions updated: DMAIEN1, DMAIEN2, DMASRCCLR, DMASRCSEL2, DMAIRC1, DMAIRC1CLR, DMAIRC2, DMAIRC2CLR
1.2	2016-10-05	ADC2 IIR-Filter updated
		BEMF Comparator Interrupt Structure added
		ADC2.Ch4 (MON) interrupts routing to NMI supply interrupt SYS_SUPPLY_IRQ_CTRL.MON_OV_IE added SYS_SUPPLY_IRQ_CTRL.MON_UV_IE added SYS_SUPPLY_IRQ_STS.MON_OV_IS added SYS_SUPPLY_IRQ_STS.MON_UV_IS added SYS_SUPPLY_IRQ_STS.MON_OV_STS added SYS_SUPPLY_IRQ_STS.MON_OV_STS added
		Structure of PREWARN_SUP MON connection updated
		External Voltage Regulator 5.0V (VDDEXT) editorial changes
		Simplified Application Diagram Example reverse polarity simplified
		Timer 2 Module I/O Interface updated, P2.5 direction corrected
		ADC1 Trigger Selection CCU6.T13 and GPT12.T6 selection added
		UART1 Module I/O Interface updated, P1.4 stated to be input and output
		UART2 Module I/O Interface updated, P1.2/TXD2 replaced by P1.1/TXD2
		Information about Re-enabling of VAREF added
		Bridge Driver Diagnosis editorial changes, IDISCHARGE_TRIM value corrected to 1
		CSA ADC Code Calculation formulars corrected
		Driver Module Block Diagram (incl. system connections) updated
		LIN Baud Rate Detection chapter reworked
		LINSCLR moved into UART chapter

Revision History

Revision	Date	Changes
		In chapter Hardware Reset the register link corrected
1.1	2016-06-14	PLL Functional Description Normal Mode editorial changes
		Clock Tree figure updated
		MODPISEL1.GPT12CAPINB updated
		EXICON0.MON renamed to MON_Trig_Sel
		External Interrupt Control Registers editorial changes
		DMAIRC2.SSC1 renamed to SSC1RDY
		DMAIRC2.SSC2 renamed to SSC2RDY
		GPT12E.PISEL.ISCAPIN editorial changes
		CCU6.PISEL0.ISPOSx BEMF assigment corrected
		Chapter BEMF Comparators updated, figures and registers updated
		Register TRIM_DRVx updated, DRV_CCP_TIMMUL added (BF-Step)
1.0	2015-06-02	Initial Release

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2023-07-10
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2023 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about any aspect of this document?
Email: erratum@infineon.com

Document reference
Z8F80155847

IMPORTANT NOTICE
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics ("Beschaffenheitsgarantie").
With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.
In addition, any information given in this document is subject to customer's compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer's products and any use of the product of Infineon Technologies in customer's applications.
The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer's technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

For further information on technology, delivery terms and conditions and prices, please contact the nearest Infineon Technologies Office (**www.infineon.com**).

WARNINGS
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies' products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.