MOTIX™ TLE984xQX

Microcontroller with LIN and power switches for automotive applications

About this document

This user manual is addressed to embedded hardware and software developers. It provides the reader with detailed descriptions about the behavior of the MOTIX™ TLE984xQX functional units and their interaction.

The manual describes the functionality of the superset device (TLE9844QX) of the MOTIX™ TLE984xQX Embedded Power IC family. For the available functionality (features) of a specific MOTIX™ TLE984xQX derivative (derivative device), please refer to the respective datasheet. For simplicity, the various device types are referenced by the collective term MOTIX™ TLE984xQX throughout this manual.

Table of contents

Table of contents

Table of contents

Table of contents

Table of contents

Table of contents

Table of contents

1 Overview

Summary of features

- 32-bit Arm®[1)] Cortex®-M0 core
  - Up to 25 MHz or 40 MHz (product variant dependent) clock frequency
  - One clock per machine cycle architecture
  - Single cycle multiplier
- On-chip memory
  - 36, 40, 48, 52 or 64 KB (product variant dependent) flash (including EEPROM)
  - 4 KB EEPROM (emulated in flash)
  - 768 bytes 100 time programmable memory (100TP)
  - 2 or 4 KB (product variant dependent) RAM
  - Boot ROM for start-up firmware and flash routines
- On-chip OSC
- 2 low-side switches incl. PWM functionality, can be used e.g. as relay driver
- 1 or 2 (product variant dependent) high-side switches with cyclic sense option and PWM functionality, e.g. for supplying LEDs or switch panels (min. 150 mA)
- 4 or 5 (product variant dependent) high-voltage monitor input pins for wake-up and with cyclic sense with analog measurement option
- 10 general-purpose I/O ports (GPIO)
- 6 analog input ports
- 10-bit A/D converter with 6 analog inputs and VBAT_SENSE and VS and 4 or 5 (product variant dependent) high voltage monitoring inputs
- 8-bit A/D converter with 7 inputs for voltage and temperature supervision
- Measurement unit with 12 channels together with the on-board 10-bit A/D converter and data post processing
- 16-bit timers – GPT12, Timer2 and Timer21
- Capture/compare unit for PWM signal generation (CCU6)
- 2 full-duplex serial interfaces (UART1, UART2), UART1 with LIN support
- 2 synchronous serial channels (SSC1, SSC2)
- Usage as P/N-channel power MOSFET driver (half-bridge application) supported by four additional differential channels in ADC1 (product variant dependent, only TLE9845QX)
- On-chip debug support via 2-wire SWD
- LIN bootstrap loader to program the flash via LIN (LIN BSL)
- Single power supply from 3.0 V to 28 V
- Low-dropout voltage regulators (LDO)
- 5 V voltage supply VDDEXT 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:
  - Microcontroller unit slow-down mode
  - Sleep mode with cyclic sense option
  - Cyclic wake-up during sleep mode
  - Stop mode with cyclic sense option
- Power-on and undervoltage/brownout reset generator

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

- Overtemperature protection
- Short circuit protection for all voltage regulators and actuators (high-side, low-side)
- Loss of clock detection with fail safe mode for power switches
- Temperature range T_j: -40°C up to 150°C
- Package VQFN-48-31 with LTI feature
- Green package (RoHS compliant)
- AEC qualified

MOTIX™ TLE984xQX product variants

Following the product family concept, some features or parameters differ between products.
The devices types for the product family are summarized in the following table.

Table 1 MOTIX™ TLE984xQX product variants

Product name	Flash size	RAM size	Max. operating frequency	High-side switches	High voltage monitor inputs	PN MOS driver
TLE9842QX	36 KB	2 KB	25 MHz	1 HS	4 MON	No
TLE9842-2QX	40 KB	2 KB	40 MHz	2 HS	5 MON	No
TLE9843QX	48 KB	4 KB	25 MHz	1 HS	4 MON	No
TLE9843-2QX	52 KB	4 KB	40 MHz	2 HS	5 MON	No
TLE9844QX	64 KB	4 KB	25 MHz	1 HS	4 MON	No
TLE9844-2QX	64 KB	4 KB	40 MHz	2 HS	5 MON	No
TLE9845QX	48 KB	4 KB	40 MHz	2 HS	5 MON	Yes

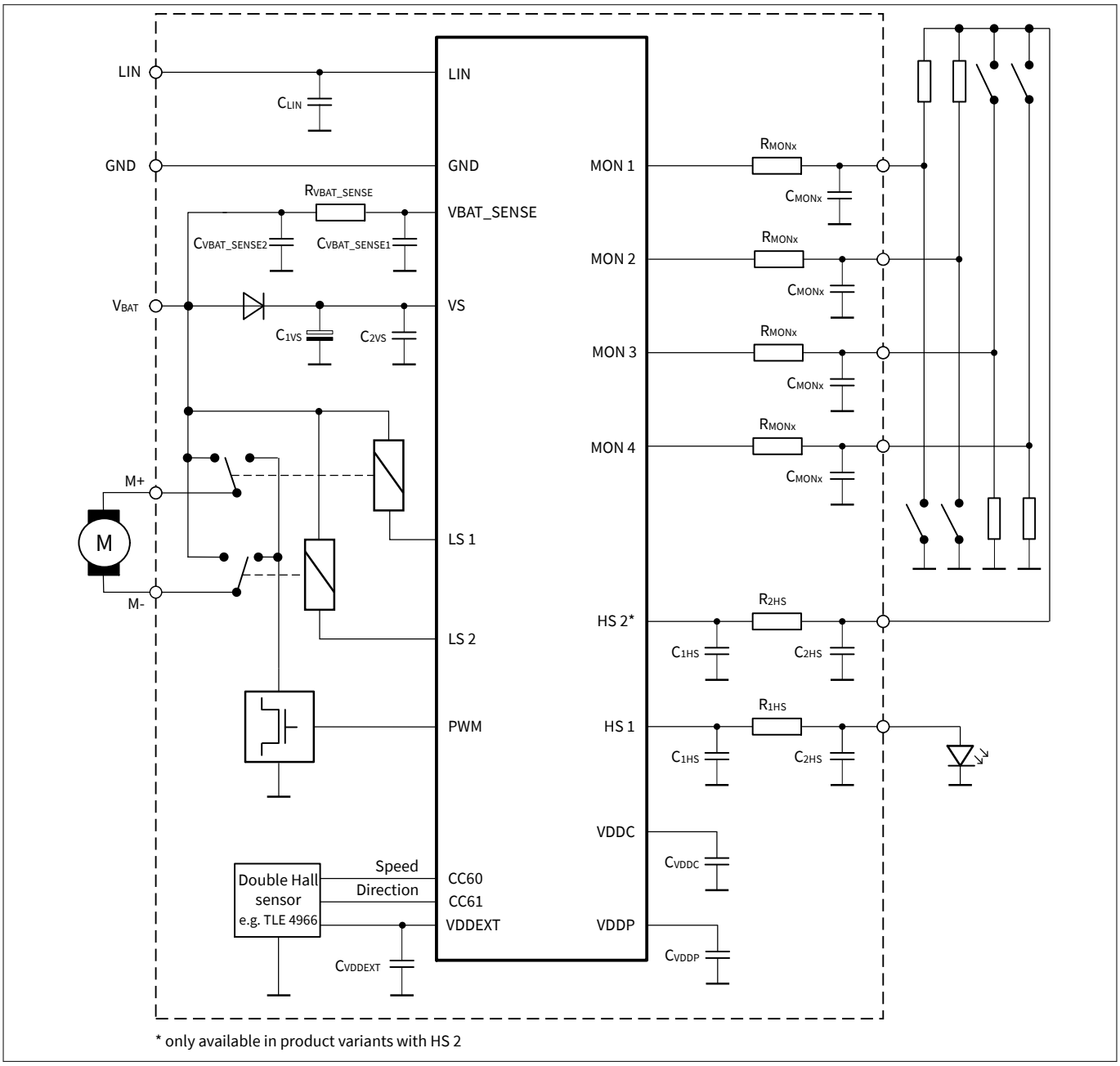


Figure 1 Typical application

1.1 Abbreviations

The following acronyms and terms are used within this document.

Table 2 Acronyms

Acronyms	Name
AHB	Arm® advanced high-performance bus
CCU6	Capture compare unit 6
CGU	Clock generation unit
CLKMU	Clock management unit
CMU	Cyclic management unit
DPP	Data post processing
ECC	Error correction code
EEPROM	Electrically erasable programmable read only memory
GPIO	General purpose input output
HV	High voltage
ICU	Interrupt control unit
LDO	Low dropout voltage regulator
LIN	Local interconnect network
LSB	Least significant bit
LTI	Lead tip inspection
LV	Low voltage
MCU	Microcontroller unit
MF	Measurement functions
MPU	Memory protection unit
MRST	Master receive/slave transmit, corresponds to MISO in SPI
MSB	Most significant bit
MTSR	Master transmit/slave receive, corresponds to MOSI in SPI
MU	Measurement unit
NMI	Non-maskable interrupt
NVIC	Nested vector interrupt controller
OSC	Oscillator
OTP	One time programmable
PBA	Peripheral bridge
PC	Program counter
PCU	Power control unit
PD	Pull down
PGU	Power supply generation unit

(table continues...)

Table 2 (continued) Acronyms

Acronyms	Name
PLL	Phase locked loop
PMU	Power management unit
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	System control unit
SOW	Short open window (for WDT1)
SPI	Serial peripheral interface
SSC	Synchronous serial channel
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 band gap
VCO	Voltage controlled oscillator
WDT1	Watchdog timer in SCU-PM (system control unit – power modules)
WMU	Wake-up management unit
100TP	100 times programmable

2 Block diagram

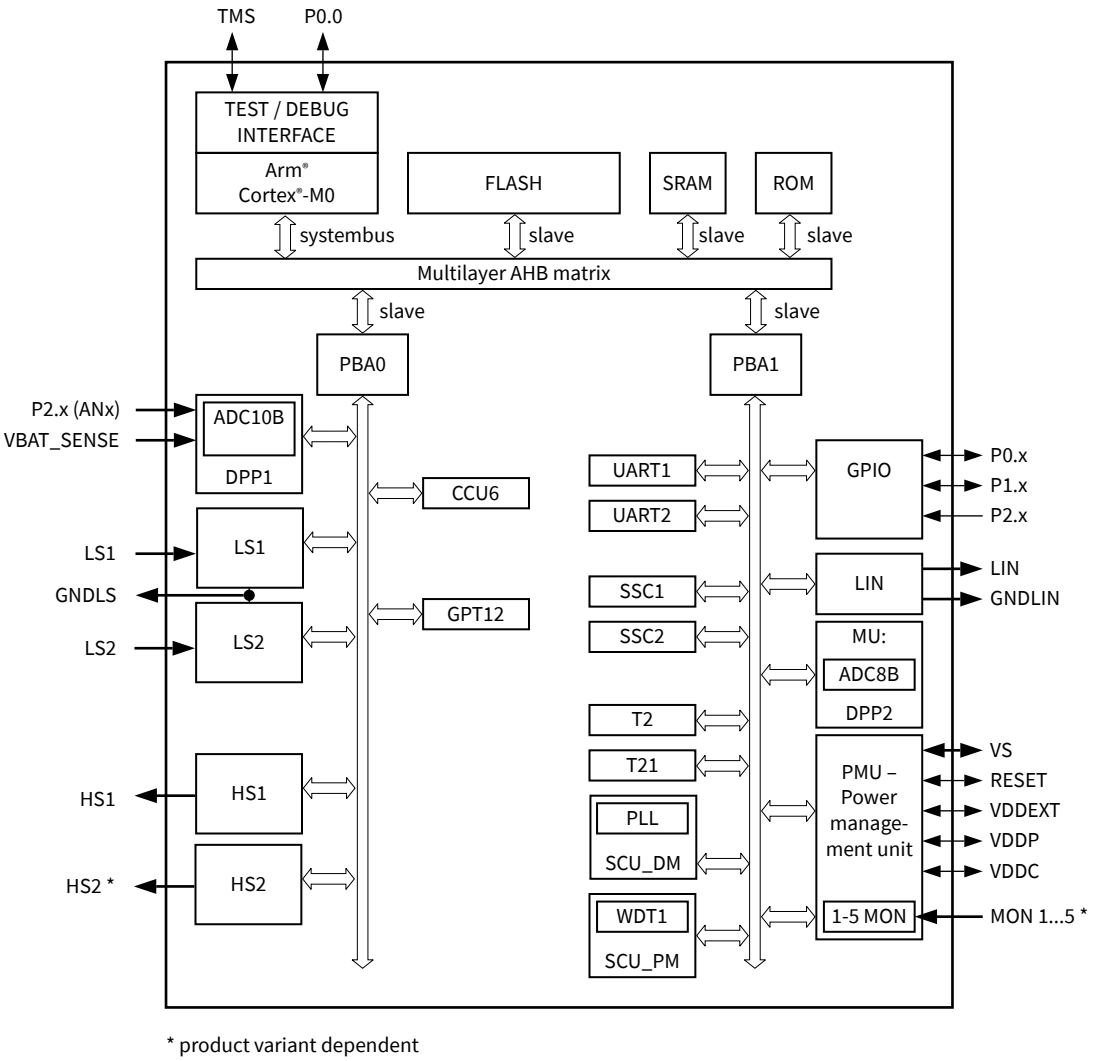


Figure 2 Block diagram, MOTIX™ TLE984xQX

3 Device pinout and pin configuration

3.1 Device pinout

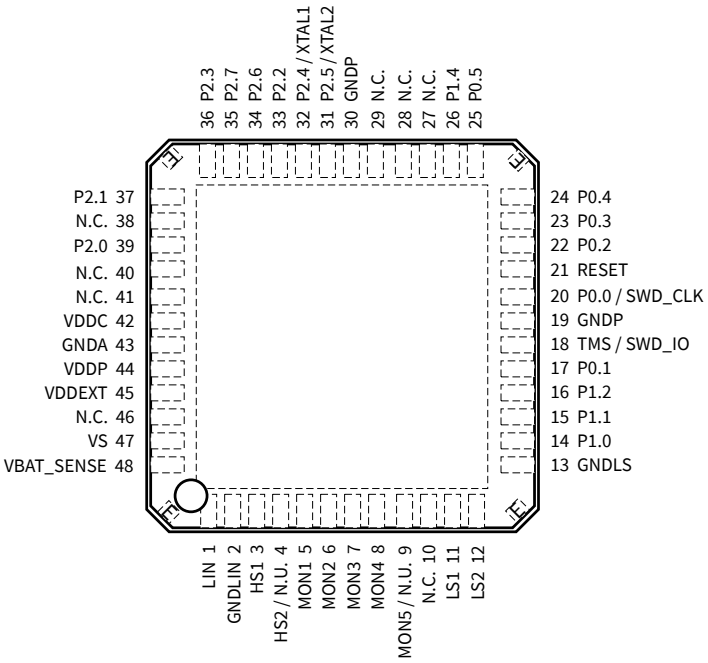


Figure 3 Device pinout, MOTIX™ TLE984xQX

3.2 Pin configuration

After reset, all pins are configured as input (except the power supply pins and the LIN pin). The table below shows the device pin types, functions and reset states. Not all alternate functions are listed, for more details see Chapter 14.

Table 3 Pin definitions and functions

Symbol	Pin number	Type	Reset state	Function	
P0				Port 0 is a 6-bit bidirectional general purpose I/O port. Alternate functions can be assigned and are listed in the port description section. Only main functions are listed below.	
P0.0	20	I/O	I/PU	SWD_CLK GPIO	Serial wire debug clock General purpose IO Alternate function mapping see Table 73
P0.1	17	I/O	I/PU	GPIO	General purpose IO Alternate function mapping see Table 73
P0.2	22	I/O	I/PD	GPIO	General purpose IO Alternate function mapping see Table 73
P0.3	23	I/O	I/PU	GPIO	General purpose IO Alternate function mapping see Table 73
P0.4	24	I/O	I/PU	GPIO	General purpose IO Alternate function mapping see Table 73
P0.5	25	I/O	I/PU	GPIO	General purpose IO Alternate function mapping see Table 73
P1				Port 1 is a 4-bit bidirectional general purpose I/O port. Alternate functions can be assigned and are listed in the port description section. Only main functions are listed below.	
P1.0	14	I/O	I	GPIO	General purpose I/O Alternate function mapping see Table 75
P1.1	15	I/O	I	GPIO	General purpose I/O Alternate function mapping see Table 75
P1.2	16	I/O	I	GPIO	General purpose I/O Alternate function mapping see Table 75
P1.4	26	I/O	I	GPIO	General purpose I/O Alternate function mapping see Table 75
P2				Port 2 is a 8-bit general purpose I/O port. Alternate functions can be assigned and are listed in the port description section. Only the main functions are listed below.	
P2.0	39	I	I	AN0	ADC1 analog input channel 12 Alternate function mapping see Table 77
P2.1	37	I	I	AN1	ADC1 analog input channel 7 Alternate function mapping see Table 77

(table continues...)

Table 3 (continued) Pin definitions and functions

Symbol	Pin number	Type	Reset state	Function	
P2.2	33	I	I	AN2	ADC1 analog input channel 8 Alternate function mapping see Table 77
P2.3	36	I	I	AN3	ADC1 analog input channel 9 Alternate function mapping see Table 77
P2.4	32	I	I	XTAL1	External oscillator input Alternate function mapping see Table 77
P2.5	31	I/O	I	XTAL2	External oscillator output Alternate function mapping see Table 77
P2.6	34	I	I	AN6	ADC1 analog input channel 10 Alternate function mapping see Table 77
P2.7	35	I	I	AN7	ADC1 analog input channel 11 Alternate function mapping see Table 77
Power supply					
VS	47	P	–	Battery supply input	
VDDP	44	P	–	I/O port supply (5.0 V). Do not connect external loads. For buffer and bypass capacitors	
VDDC	42	P	–	Core supply (1.5 V during active mode, 0.9 V during stop mode). Do not connect external loads. For buffer/bypass capacitor	
VDDEXT	45	P	–	External voltage supply output (5.0 V, 20 mA)	
GNDLS	13	P	–	Low-side ground LS1, LS2	
GNDP	19, 30	P	–	Core supply ground	
GNDA	43	P	–	Analog supply ground	
GNDLIN	2	P	–	LIN ground	
Monitor inputs					
MON1	5	I	I	High voltage monitor input 1	
MON2	6	I	I	High voltage monitor input 2	
MON3	7	I	I	High voltage monitor input 3	
MON4	8	I	I	High voltage monitor input 4	
MON5/N.U.	9	I	I	High voltage monitor input 5 (product variant dependent)	
High-side switch/low-side switch outputs					
LS1	11	O	Hi-Z	Low-side switch output 1	
LS2	12	O	Hi-Z	Low-side switch output 2	
HS1	3	O	Hi-Z	High-side switch output 1	
HS2/N.U.	4	O	Hi-Z	High-side switch output 2 (product variant dependent)	

(table continues...)

Table 3 (continued) Pin definitions and functions

Symbol	Pin number	Type	Reset state	Function	
LIN interface					
LIN	1	I/O	PU	LIN bus interface input/output	
Others					
TMS	18	I	I/PD	TMS SWD_IO	Test mode select input Serial wire debug input/output
RESET	21	I/O	I/O/PU	Reset input/output, not available during sleep mode	
VBAT_SENSE	48	I	I	Battery supply voltage sense input	
N.C.	27, 28, 29, 38, 40, 41	–	–	No internal connection, should be connected to GND	
	10, 46	–	–	No internal connection, should be connected to GND or left open	
EP	–	–	–	Exposed pad, connect to GND	

Type and default state abbreviations used in the table above:

- I/O: Input/output
- I: Input only
- O: Output only
- P: Power supply
- PU: Pull-up enabled
- PD: Pull-down enabled
- Hi-Z: High-impedance

4 Introduction

This highly integrated circuit contains analog and digital functional blocks. For system and interface control an embedded 32-bit Arm® Cortex®-M0 microcontroller is included. For internal and external power supply purposes, on-chip low drop-out regulators are existent. An internal oscillator (no external components necessary) provides a cost effective and suitable clock in particular for LIN slave nodes. As communication interface, a LIN transceiver and several high-voltage monitor inputs with adjustable threshold and filters are available. Furthermore, one (or two, depending on the device variant) high-side switch(es) (e.g. for driving LEDs or powering switches), two low-side switches (e.g. for relays) and several general purpose input/outputs (GPIO) with pulse-width modulation (PWM) capabilities are available.

The microcontroller unit supervision and system protection including reset feature is controlled 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 terminal 30 connected automotive applications. A wake-up from the power saving mode is possible via a LIN bus message, via the monitoring inputs or repetitive with a programmable time period (cyclic wake-up).

The integrated circuit is available in a package with 0.5 mm pitch and is designed to withstand the challenging conditions of automotive applications.

4.1 SOC system power modes overview

The MOTIX™ TLE984xQX has several operational modes mainly to support low power consumption requirements. For more details see Chapter 5.4.

Active mode

In Active mode all modules are activated and the MOTIX™ TLE984xQX is fully operational.

Stop mode

The Stop mode is one out of two major low-power modes. The transition to the low-power modes is done by setting the respective bits in the mode control register. In Stop mode the embedded microcontroller is still powered allowing faster wake-up reaction times, but not clocked. A wake-up from this mode is possible by LIN bus activity, the high-voltage monitor input pins or the respective 5 V GPIOs.

Sleep mode

The Sleep mode is a major low-power mode. The transition to the low-power modes is done by setting the respective bits in the microcontroller unit mode control register. The sleep time is configurable. In Sleep mode the embedded microcontroller power supply is deactivated, allowing the lowest system power consumption, but the wake-up time is longer compared to the Stop mode. In this mode a 64-bit wide buffer for data storage is available. A wake-up from this mode is possible by LIN bus activity or the high-voltage monitor input pins and cyclic wake. A wake-up from Sleep mode behaves similar to a power-on reset. While changing into Sleep mode, no incoming wake-requests are lost (that means no dead-time). It is possible to enter Sleep mode even with LIN dominant.

Cyclic wake-up

The cyclic wake-up is a special feature of the Sleep and Stop mode. The enabling of cyclic wake-up is done by first setting the respective bits in the mode control register followed by the SLEEP or STOP command. Additional to the cyclic wake-up behavior (wake-up after a programmable time period), the wake-up sources of the normal Stop mode and Sleep mode are available.

Cyclic sense

The cyclic sense is a special feature of the Sleep mode and the Stop mode. The enabling to the cyclic is done by first setting the respective bits in the mode control register followed by the STOP or SLEEP command. For example, in cyclic sense the high-side switch can be switched on periodically for biasing some switches. The wake-up condition is configurable, when the sense result of defined monitor inputs at a window of interest changed compared to the previous wake-up period or reached a defined state respectively. In this case the Active mode is entered immediately.

The following table shows the possible power mode configurations of each major module or function respectively.

Table 4 Power mode configurations

Module/function	Active mode	Sleep mode	Stop mode	Comment
VPRE, VDDP, VDDC	ON	OFF	ON	–
VDDEXT	ON/OFF	OFF	Cyclic/ON/OFF	–
HSx	ON/OFF	Cyclic	Cyclic	Cyclic sense
LSx	ON/OFF	OFF	OFF	–
LIN TRX	ON/OFF	Wake-up only/OFF	Wake-up only/OFF	–
MONx (wake-up)	n.a.	Disabled/static/cyclic		

(table continues...)

Table 4 (continued) Power mode configurations

Module/function	Active mode	Sleep mode	Stop mode	Comment
			Disabled/static/ cyclic	Cyclic: combined with HS=on
MONx (measurement)	ON/OFF	OFF	OFF	Available on all channels
VS sense	ON/OFF brownout detection	Brownout detection	Brownout detection	Brownout detection done in PCU
VBAT_SENSE	ON/OFF	OFF	OFF	–
GPIO 5V	ON	OFF	ON	–
WDT1	ON	OFF	OFF	–
CYCLIC WAKE	n.a.	Cyclic wake-up/cyclic sense/OFF	Cyclic wake-up/ cyclic sense/OFF	Cyclic sense with HS; wake-up needs MC for enter sleep mode again
Measurement	ON[1)]	OFF	OFF	–
Microcontroller unit	ON/slow-down/STOP	OFF	OFF	–
CLOCK GEN (MC)	ON	OFF	OFF	–
LP_CLK (f_{LP_CLK})	ON	OFF	OFF	WDT1
LP_CLK2 (f_{LP_CLK2})	ON	ON	ON	For cyclic wake-up

1) May not be switched off due to safety reasons.

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. Only the software can clear the wake-up source flags. It is ensured, that no wake-up event is lost since the wake-up events are captured at anytime (Active mode, transitions, Sleep mode, Stop mode).

As default wake-up sources, MON inputs and cyclic wake are activated after power-on reset, LIN is disabled as wake-up source by default.

Wake-up levels and transitions

The wake-up can be triggered by rising, falling or both signal edges for each monitor input and GPIOs individually.

4.2 Device register types

The following register types are used within this device.

Table 5 Register types

Bit-type	Meaning	Read access from SW (bus)	Write access from SW (bus)	Bit-set by HW	Bit-clear by HW	Bit-clear from SW
r	read-only of HW signal from SW , no register	yes	n/a	n/a	n/a	n/a
rc rh	bit set by HW, sticky, clear on read by SW	yes	no	yes	no	Yes, by SW read
rw	read-write from SW, not influenced by HW	yes	yes	no	no	(Covered by SW write)
rwh	written by HW, read by SW, cleared by writing 0	yes	no	yes	yes	Yes, by SW write 0
rwh1	write-only from SW. Set by SW for 1 cycle, then reset by HW	no	yes	no	yes	No
rwhir	rw from SW, can be reset by HW	yes	yes	no	yes	(Covered by SW write)
rwhis	rw from SW, can be set by HW	yes	yes	yes	no	(Covered by SW write)
rwhrs	read-write by SW, HW can also set and reset	yes	yes	yes	yes	(Covered by SW write)
rwhxr	interrupt register, level sensitive. Set by HW, sticky, read-write from SW, reset from SW only via external bit and only when HW is 0 again (HW level has priority)	yes	yes	yes – by level	no	Yes, by SW write 1 to external register-clear-bit
rwhxre	interrupt register, rising edge sensitive. Set by HW on rising edge, sticky, read-write from SW, reset from SW only via external bit	yes	yes	yes – by rising edge	no	Yes, by SW write 1 to external register-clear-bit
rwpt	read-only in user mode (only writable in testmode and by firmware)	yes	(not in user mode)	no	no	(Not in user mode)
rwpw	only writable from SW after PASSWD-protection has been opened (for 32 cycles)	yes	yes, but gated by PASSWD-protection	no	no	(Covered by SW write)
w	clear on write 1, for interrupts (interrupt status clear bit) and	no	no	no	no	Yes, by SW write 1

(table continues...)

Table 5 (continued) Register types

Bit-type	Meaning	Read access from SW (bus)	Write access from SW (bus)	Bit-set by HW	Bit-clear by HW	Bit-clear from SW
	sticky status registers (status clear)					

4.3 Device reset masks

The reset mask as defined in the register reset table indicates the bit fields affected by the corresponding RESET_TYPE.

Below example illustrates a RESET_TYPE_3 case. Similar principle applies to the other RESET_TYPE.

Table 6 Reset mask example

Reset type	Reset value	Note
RESET_TYPE_3	$0000\ 0000_H$	Reset mask: 0x00000044

Reset type = RESET_TYPE_3

Reset mask = 0x00000044 = $0000\ 0044_H$ = $0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0100\ 0100_B$.

A RESET_TYPE_3 event resets bit 2 and 6 to their default values (i.e. reset value), in this example to 0.

5 Power management unit (PMU)

5.1 Features

- System modes control (start-up, sleep, stop and active)
- Power management (cyclic wake)
- 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 TRX, MON, GPIOs)
- System error logging

5.2 Introduction

The power management unit is responsible for generating all needed voltage supplies for the embedded MCU (VDDC, VDDP) and the external supply (VDDEXT). 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 functionalities especially the reset behavior of the embedded MCU. Another purpose of the power management unit is to ensure the fail safe behavior of the system IC.

Therefore the power management unit controls all system modes including the corresponding transitions. To ensure this system-master role of the PMU, an independent logic supply and system clock are internally implemented and they work independently from the MCU clock.

5.2.1 Block diagram

The following figure shows the structure of the power management unit. The table below describes the submodules more detailed.

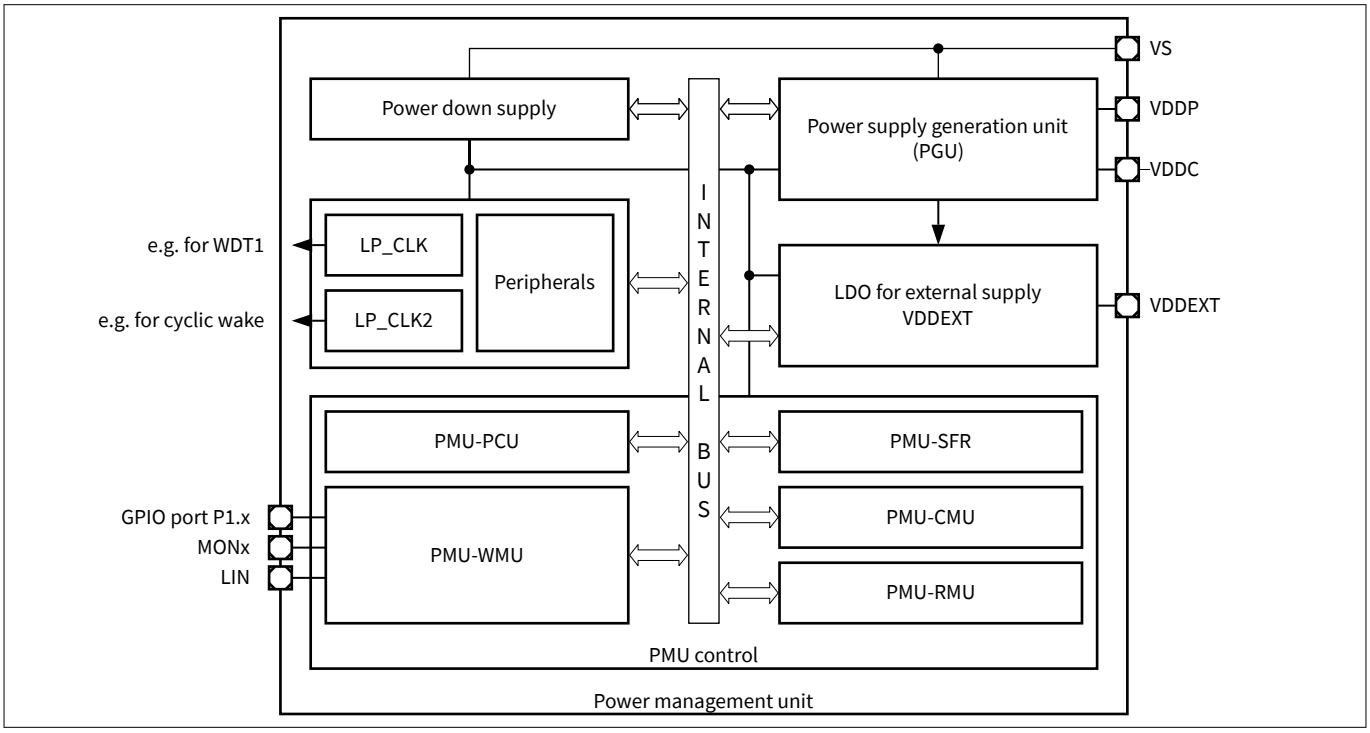


Figure 4 Power management unit block diagram

Table 7 Description of PMU submodules

Mod. name	Modules	Functions
Power down supply	Independent supply voltage generation for PMU	This supply is dedicated to the PMU to ensure an independent operation from generated power supplies (VDDP, VDDC)
LP_CLK (= f_{LP_CLK})	• Clock source for all PMU submodules • Backup clock source for system • Clock source for WDT1	This ultra low power oscillator generates the clock for the PMU This clock is also used as backup clock for the system in case of PLL clock failure and as independent clock source for WDT1
LP_CLK2 (= f_{LP_CLK2})	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 start-up and operation (bandgap, bias)
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)

(table continues...)

Table 7 (continued) Description of PMU submodules

Mod. name	Modules	Functions
VDDEXT	Voltage regulator for VDDEXT to supply external modules (e.g. sensors)	This voltage regulator is a dedicated supply for external modules
PMU-SFR	All PMU relevant extended special function registers	This module contains all PMU relevant registers, which are 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 diagnosis like under- and overvoltage detection, overcurrent and short circuit diagnoses
PMU-WMU	Wake-up management unit of the PMU	This block is responsible for controlling all wake-up related actions within the PMU module
PMU-CMU	Cyclic management unit of the PMU	This block is responsible for controlling all actions within cyclic mode
PMU-RMU	Reset management unit of the PMU	This block generates resets triggered by the PMU like undervoltage or short circuit reset, and passes all resets to the relevant modules and their register. A reset status register with every reset source is available

5.2.2 PMU modes overview

The PMU offers a set of four operating modes:

- Active mode: The embedded system supplies, VDDP and VDDC, are operational and VDDEXT can be enabled by the user software
- Sleep mode: All embedded system supplies, VDDC, VDDP and VDDEXT, are turned off
- Fail Sleep mode: Same as for Sleep mode
- Stop mode: VDDP, VDDEXT and VDDC are operational. VDDEXT can be kept off by the user or be used for cyclic sense (see Chapter 5.6.1)

Active mode is the PMU state that allows full operation of the embedded system, so is the one that allows for the user software and application to run.

In order to decrease power consumption during idle application instances, the user can set the system and the power supply generation in Stop mode or Sleep mode. In case of predefined fail safe scenarios the PMU will be able to lead the system in Fail Sleep mode, which also allows for low power consumption. In Chapter 5.4.6.1 a detailed description of these mechanisms can be found.

5.3 Power supply generation (PGU)

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

Submodules of PSG

- Power down supply: independent analog supply voltage generation for power control unit logic, for VDDP regulator and for VDDC regulator
- VPRE: analog supply voltage pre-regulator. Purpose of this regulator is the power dissipation reduction for the following regulator stages and to provide for superior line regulation
- 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.
- PCU: Power control unit responsible for supervising and controlling 5 V regulator and 1.5 V regulator

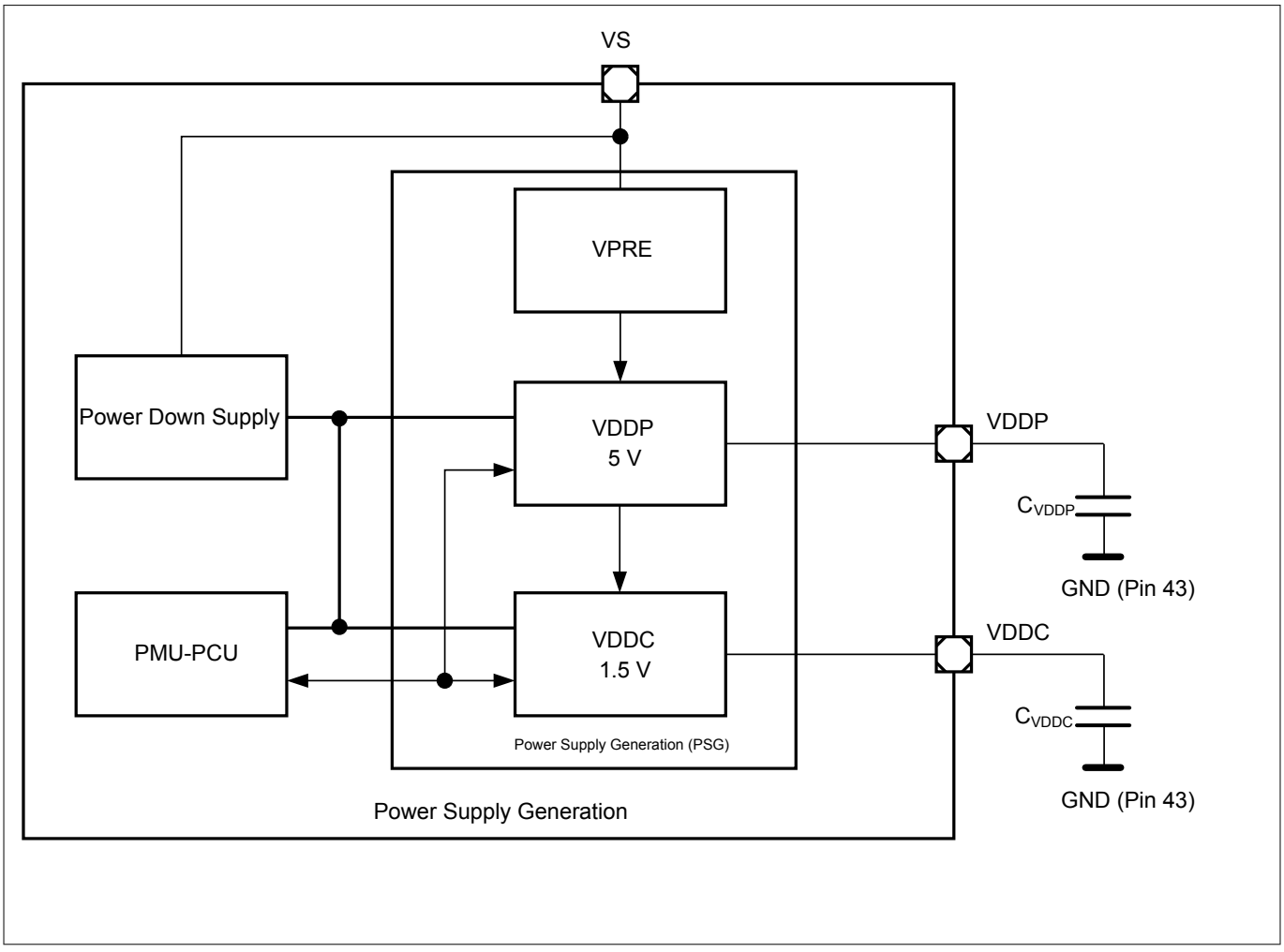


Figure 5 Power supply generation block diagram

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})
- Overtemperature shutdown with MCU signaling (interrupt)
- Pre-regulator for 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 a proper regulator functionality.

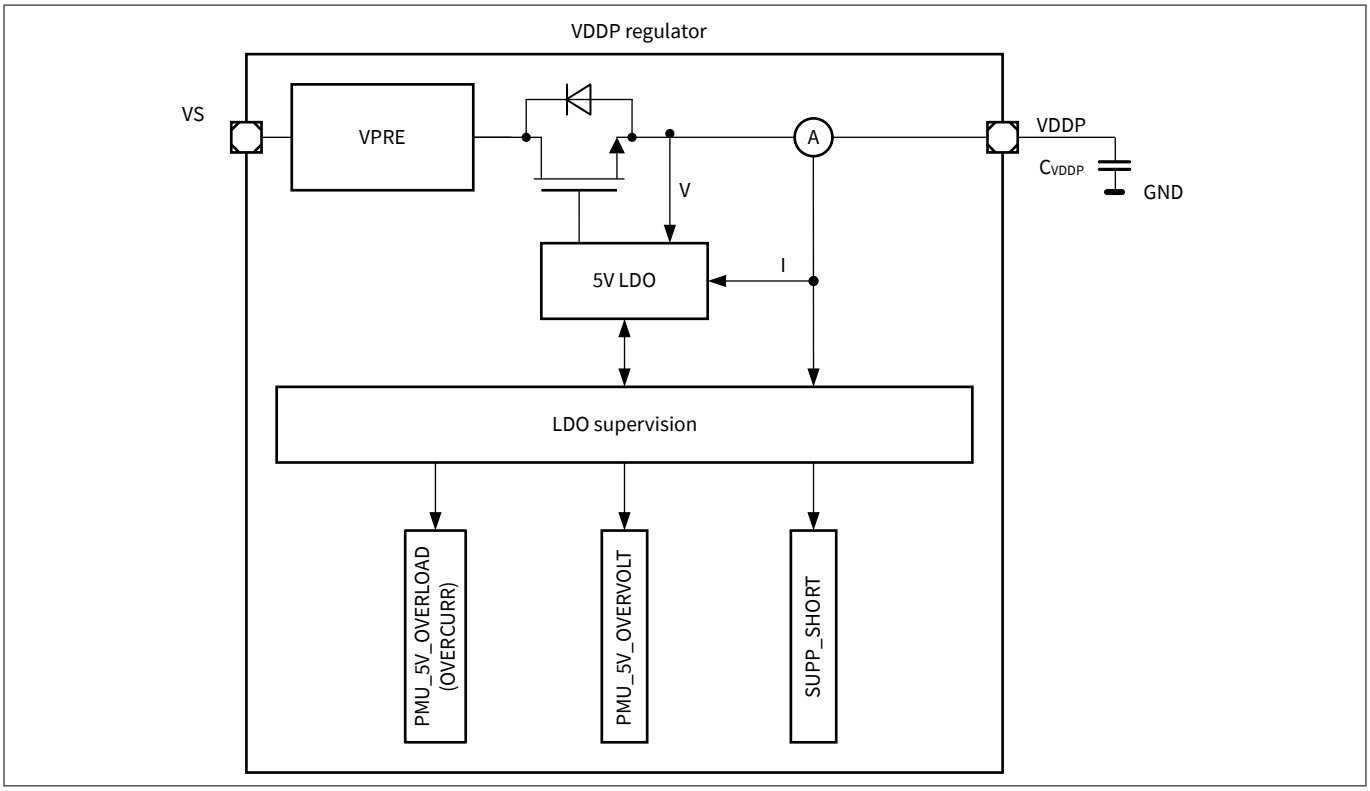


Figure 6 Module block diagram of VDDP voltage regulator

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, digital peripherals and other chip internal analog 1.5 V functions (e.g. ADC).

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
- Overtemperature shutdown with MCU signaling (interrupt)
- 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.

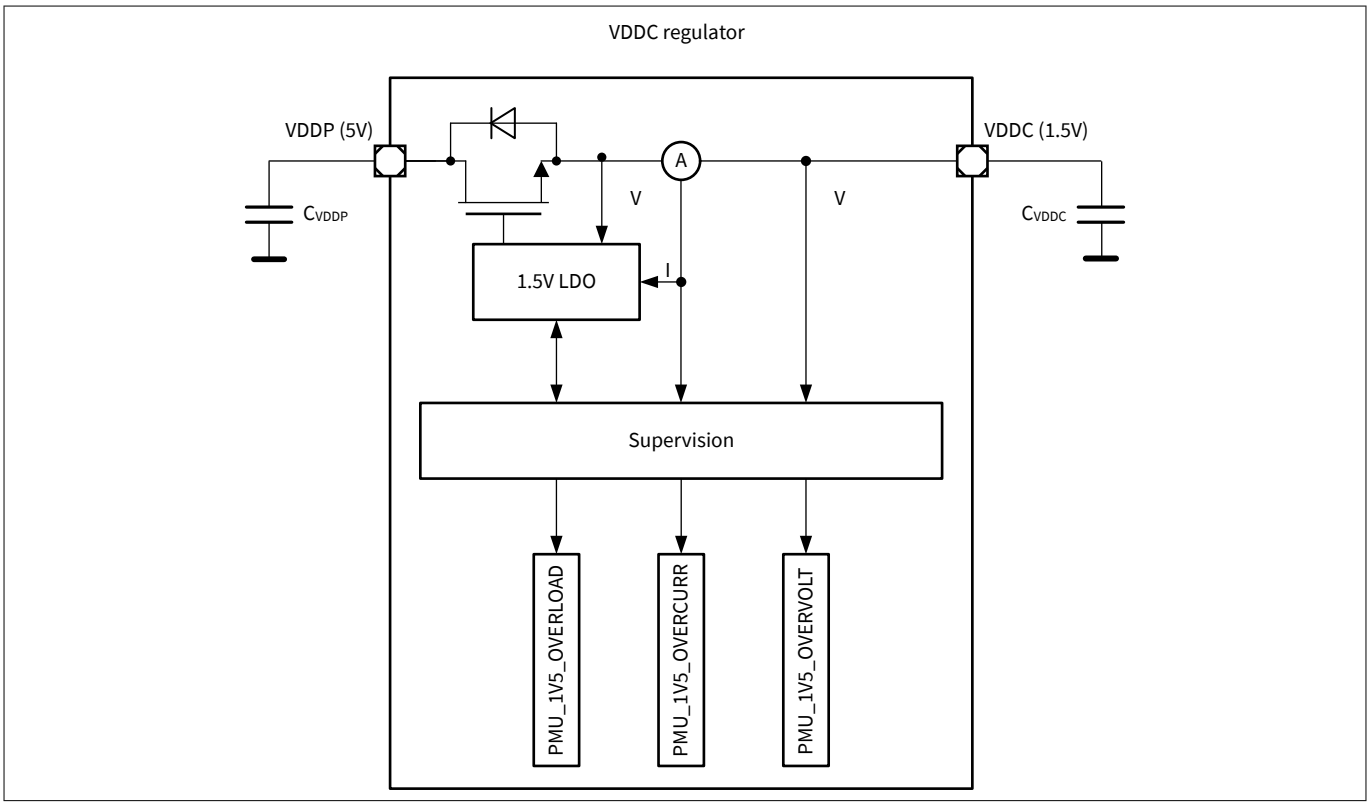


Figure 7 Module block diagram of VDDC voltage regulator

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.

Features

- Switchable (by software) +5 V, low-drop voltage regulator
- Switch-on undervoltage blanking time in order to drive small capacitive loads
- Intrinsic current limitation
- Undervoltage monitoring and shutdown with MCU signaling (interrupt)
- Overtemperature shutdown with MCU signaling (interrupt)
- Pull-down current source at the output for Sleep mode only (typ. 100 µA)
- Cyclic sense option together with GPIOs (Stop mode only)
- Low current mode available to ensure reduced stop mode current consumption. In this mode current capability is reduced to I_{VDDEXT_LCM}
- VDDEXT automatically resumes operation after clearing the undervoltage status flag PMU_VDDEXT_CTRL.VDDEXT_UV_ISC

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

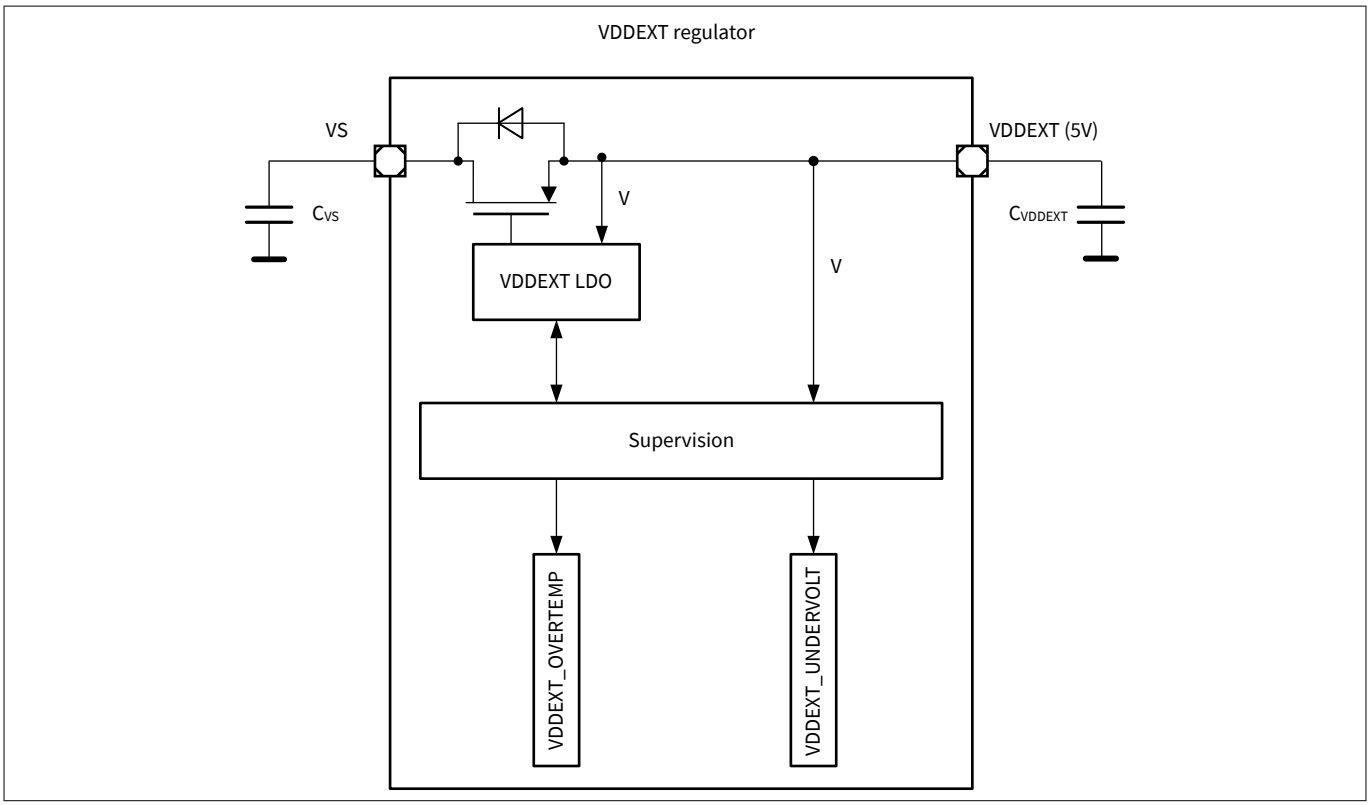


Figure 8 Module block diagram

5.3.3.1 VDDEXT internal diagnosis

When VDDEXT is used as a supply e.g. for external pull-up switches, overtemperature and/or undervoltage may occur because of external causes. These conditions 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 revive the system from Stop mode (see Chapter 5.4.3) and will be signaled in the PMU_WAKE_STATUS register.

Overtemperature

This diagnosis is available in Active mode. However the event will also act as wake-up source when:

- PCU is transitioning to and from Stop mode
- VDDEXT is powered on for cyclic sense (see PMU_VDDEXT_CTRL register for VDDEXT cyclic sense setup)

When this event happens, VDDEXT will be turned off and locked off (VDDEXT_OT_STS=1 and VDDEXT_OT_IS=1). To remove the lock the software must follow this sequence:

1. Clear the interrupt flags: VDDEXT_OT_ISC=1 and VDDEXT_OT_SC=1
2. Enable VDDEXT: PMU_VDDEXT_CTRL.VDDEXT_ENABLE=1

Undervoltage

When this event happens, VDDEXT will be turned off and locked off (VDDEXT_UV_IS=1). To remove the lock the software must follow this sequence:

1. Clear the interrupt flag: VDDEXT_UV_ISC=1
2. Enable VDDEXT: PMU_VDDEXT_CTRL.VDDEXT_ENABLE=1

5.3.4 Low-V_S operation

The integrated VDDP regulator will enter dropout operation as the VS pin voltage is dropping below the min. 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:

- Low-side, high-side switches
- GPIOs
- Transceiver interface
- VDDEXT regulator

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

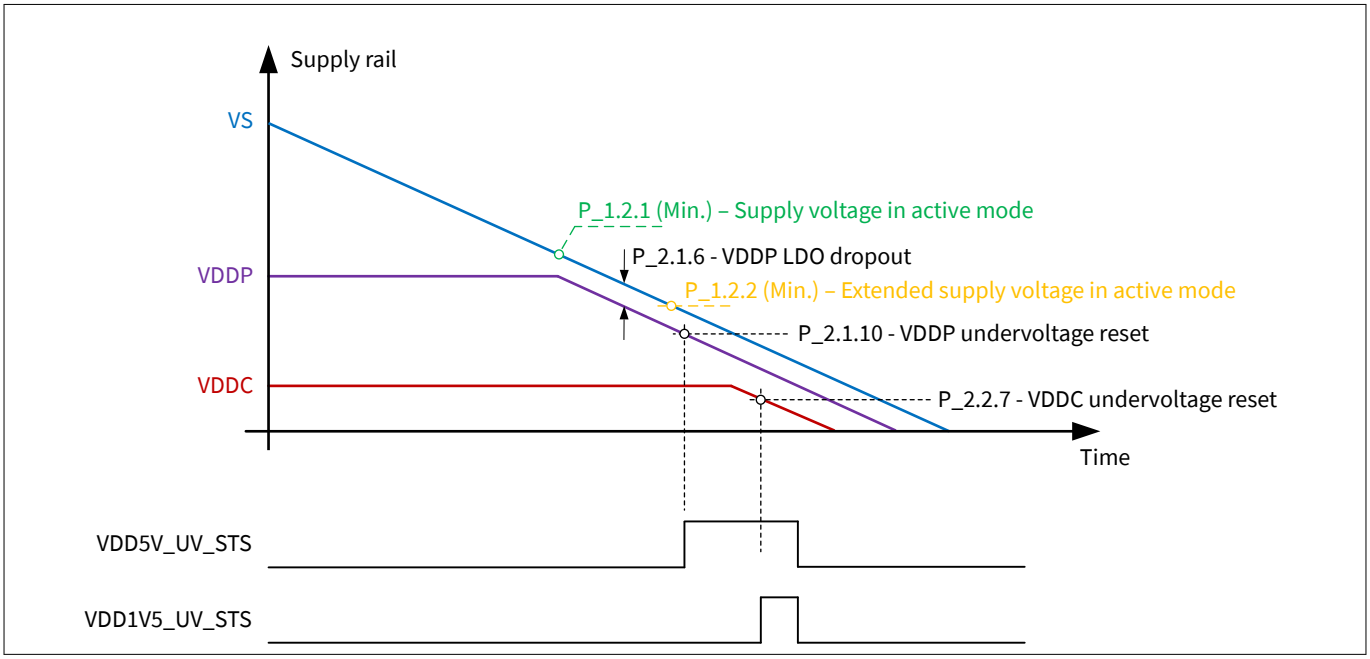


Figure 9 Low-V_S operation[2)]

2 P_*: See the datasheet for details.

5.3.5 PGU ADC2 monitoring

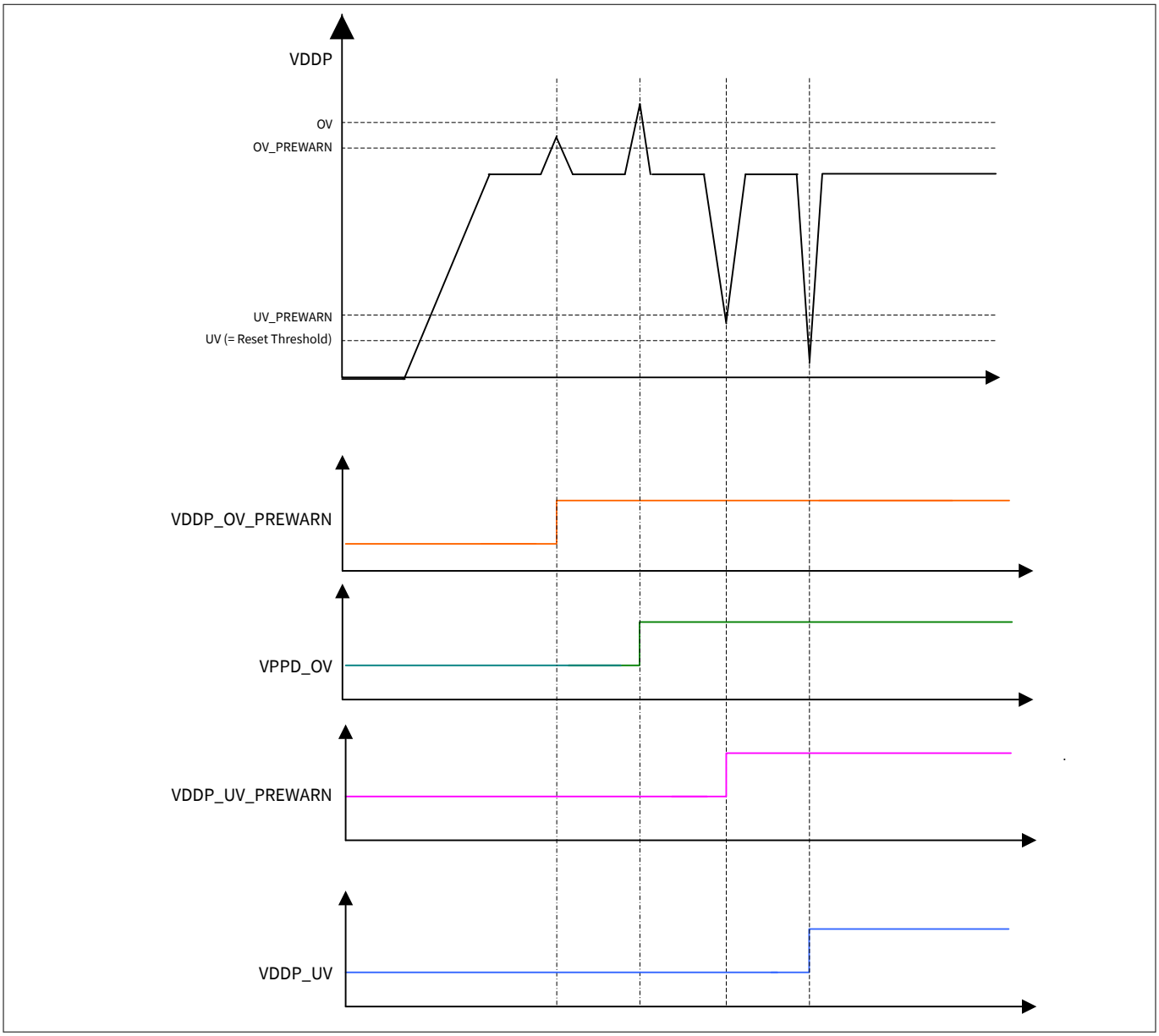


Figure 10 VDDP monitoring with ADC2 CH2

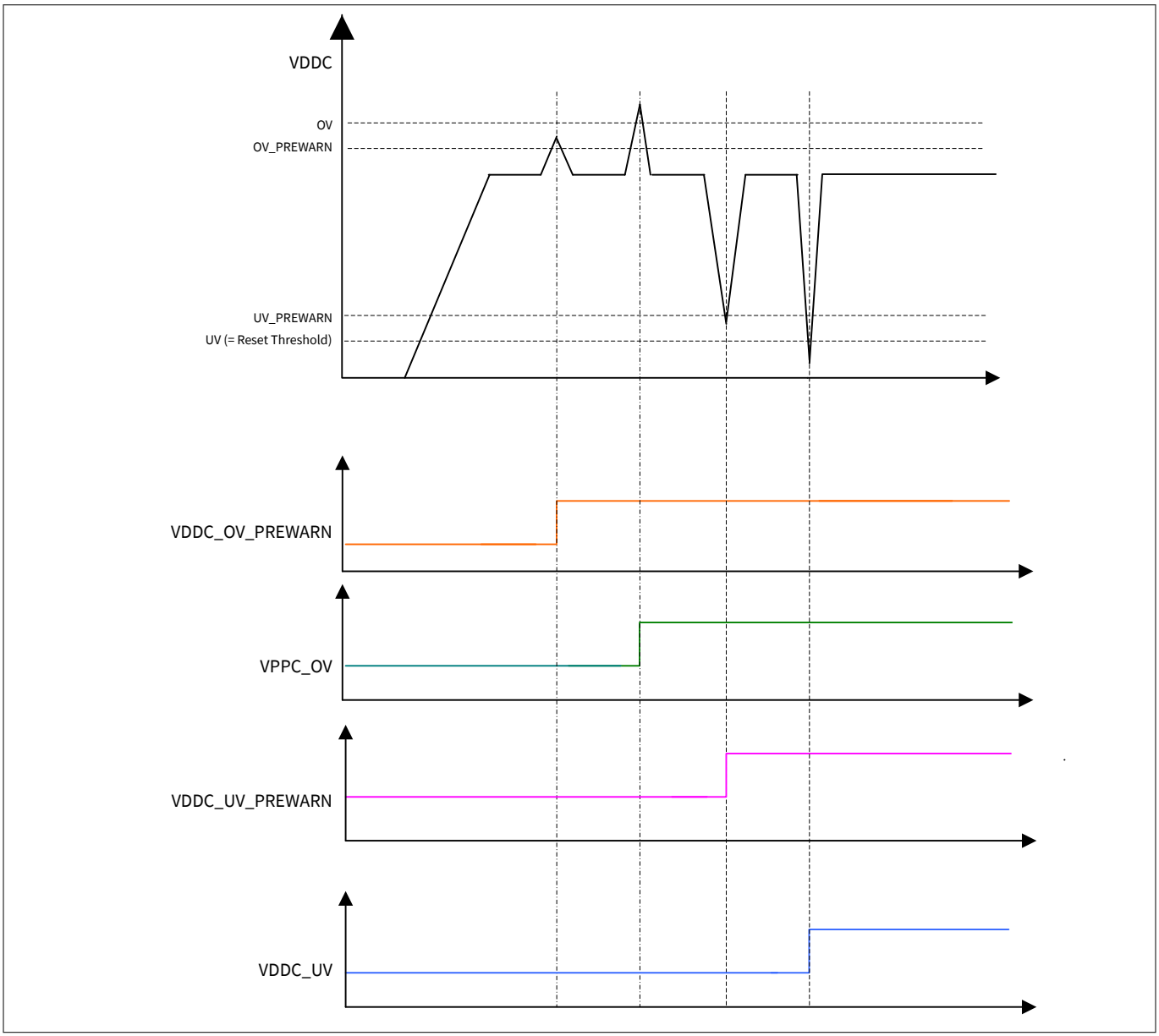


Figure 11 VDDC monitoring with ADC2 CH4

5.3.6 Power supply generation (PGU) registers

The registers are addressed wordwise.

5.3.6.1 Register overview - Power Supply Generation registers (ascending offset address)

Table 8 Register overview - Power Supply Generation registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_SUPPLY_STS	Voltage reg status register	0008_H	79
PMU_VDDEXT_CTRL	VDDEXT control register	$000C_H$	81

5.3.6.2 Power supply generation register

The PMU_SUPPLY_STS register is dedicated to control the voltage regulators VDDP and VDDC.
It provides an overview about the status of the two voltage supplies.

5.3.6.3 VDDEXT control register

The VDDEXT can be fully controlled by the PMU_VDDEXT_CTRL register, including all diagnosis functions.

5.4 Power control unit (PCU)

The power control unit is the controlling instance of the system power generation unit (PGU). It offers important fail safe features which will be described in the next chapters.

The state diagram in the figure below is a functional representation of the PMU state machine. In particular the following properties are highlighted:

- Power modes and states
- Fail, application, reset and internal events
- Reset types (see Chapter 5.5 for details)
- Supply and watchdog error counters operations

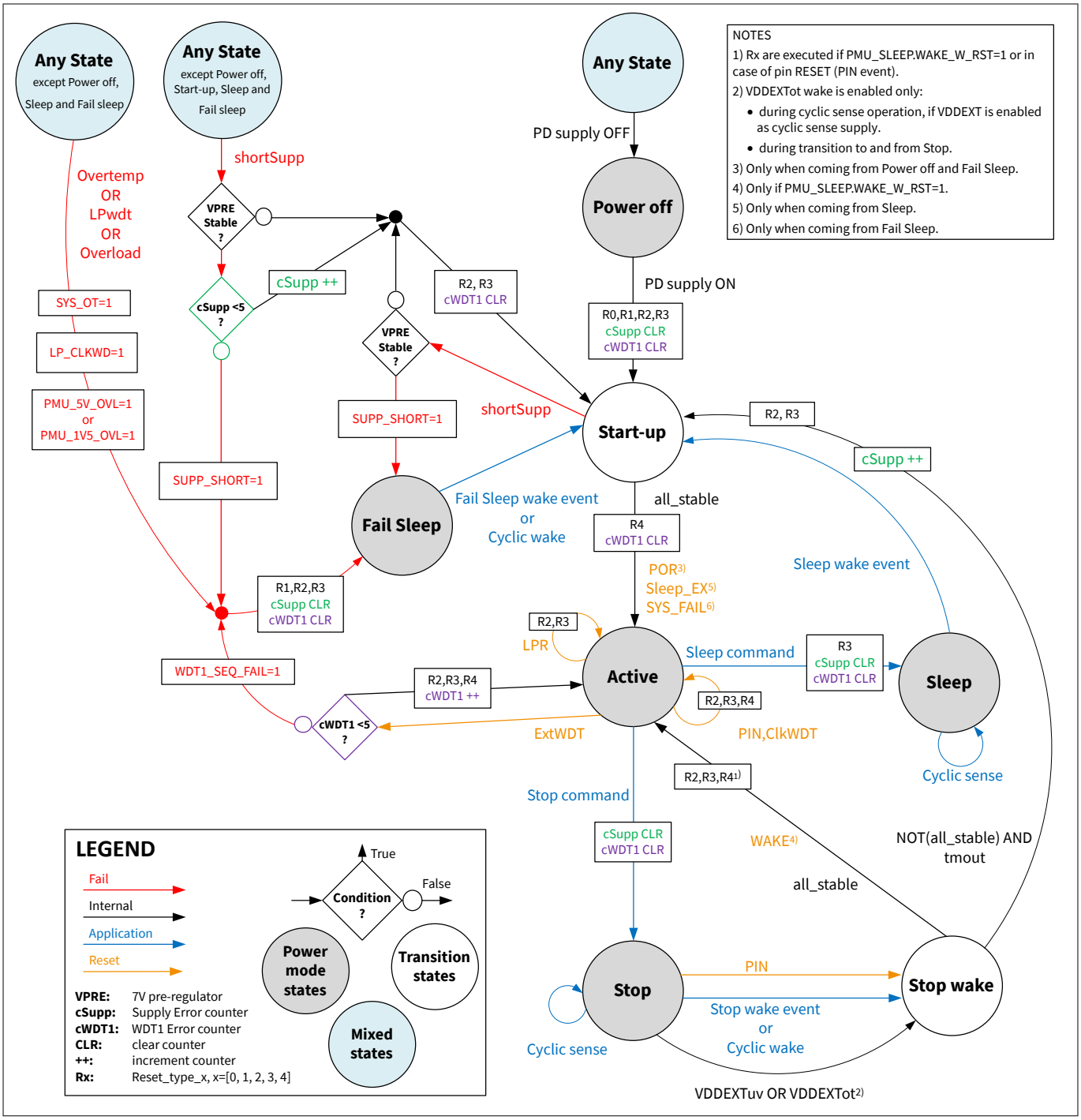


Figure 12 Power control unit state diagram

Table 9 PCU state diagram events

Event	Short description	Details see
PD supply ON/OFF	Power-down supply turned on/off	Chapter 5.4.1
all_stable	VDDP and VDDC ramp up successful	Chapter 5.4.1 Chapter 5.4.4
VDDEXTuv	VDDEXT undervoltage	Chapter 5.3.3.1 Chapter 5.4.3
VDDEXTot	VDDEXT overtemperature	Chapter 5.3.3.1 Chapter 5.4.3
tmout	280 µs timeout (based on LP_CLK)	Chapter 5.4.4
shortSupp	VDDP and VDDC supply error	Chapter 5.4.6.1
Overload	VDDP and VDDC overcurrent fail	Chapter 5.4.6.1
Overtemp	System overtemperature	Chapter 5.4.6.1
LPwdt	LP_CLK watchdog	Chapter 5.4.6.1
SYS_FAIL	Fail Safe event reset	Chapter 5.5 Chapter 5.4.6.1
POR	Power on reset	Chapter 5.4.1 Chapter 5.5
PIN	RESET (pin 21) reset	Chapter 5.5
LPR	Low priority resets (PMU_SOFT OR LOCKUP)	Chapter 5.5
ClkWDT	Clock watchdog (SCU_PM) reset	Chapter 5.5
ExtWDT	WDT1 error reset	Chapter 5.5
SleepEX	Sleep wake reset	Chapter 5.5
WAKE	Stop wake reset	Chapter 5.5
Stop Command	User software sends system to Stop mode	Chapter 5.4.2
Sleep Command	User software sends system to Sleep mode	Chapter 5.4.2
Stop wake event	Application, fail or reset event	Chapter 5.4.3
Sleep wake event	Application event	Chapter 5.4.5
Fail sleep wake event	Application event	Chapter 5.4.6
Cyclic sense	Periodic sense of device pins	Chapter 5.6
Cyclic wake	Wake up after time interval expires	Chapter 5.6

5.4.1 Power-off and start-up

In power-off state, the PMU is held in a reset state. This state is kept until the main supply voltage V_S reaches the minimum value for the power down supply to operate. In this state all internal supplies are turned off. When V_S is higher than its minimum operating value, the PMU internal reset is released and the power down supply is turned on.

Start-up is a phase dedicated to the transition from Power off, Fail Sleep, Sleep or Stop (see Chapter 5.4.3 for details) to Active mode. In start-up the PMU ramps up the voltages VDDP and VDDC in a predefined sequence, which ends with the release of the MCU reset (see Chapter 5.5). When this status is reached the system will transit to Active mode. This sequence is illustrated in Figure 13.

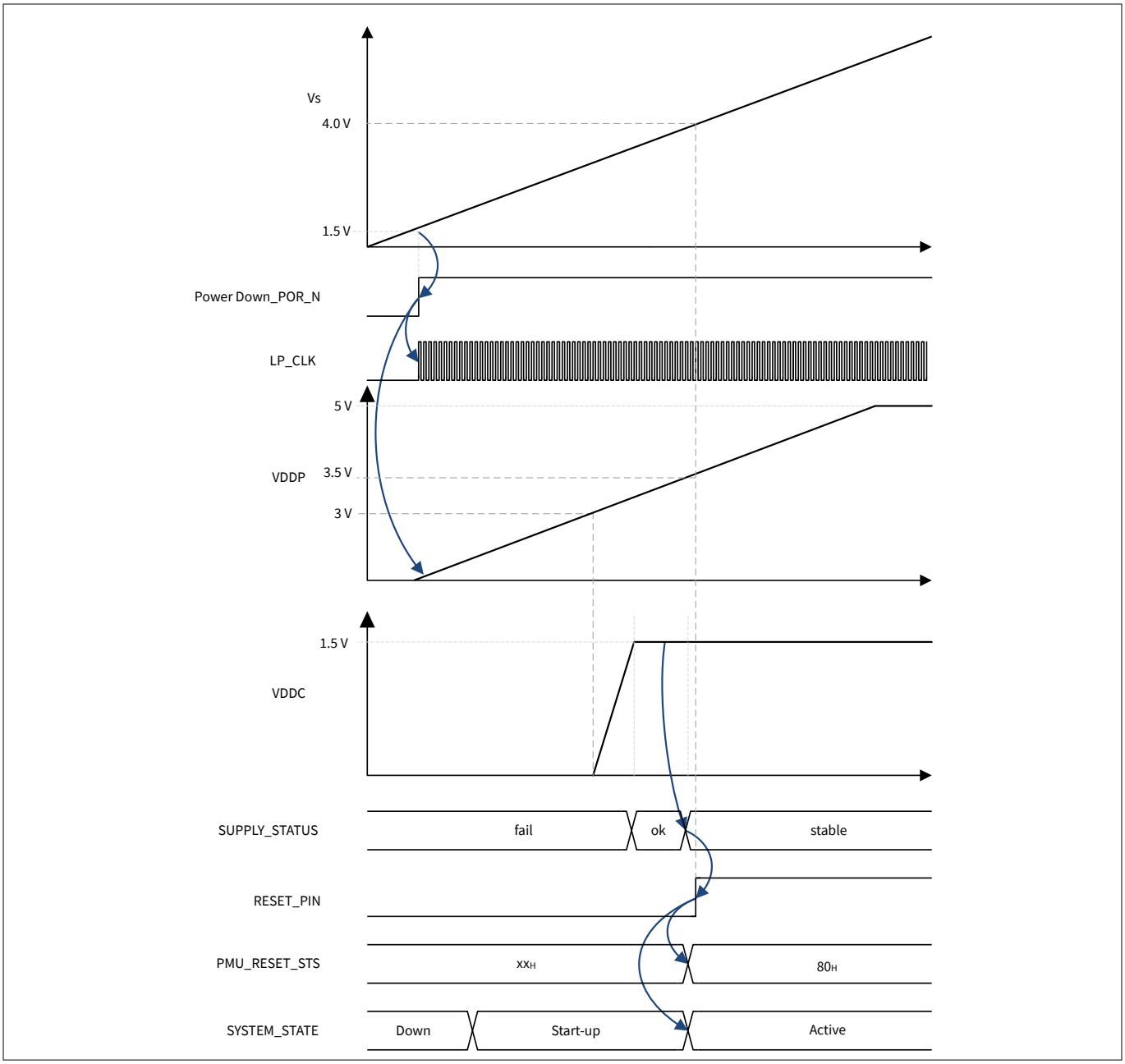


Figure 13 Power-on and start-up behavior of reset

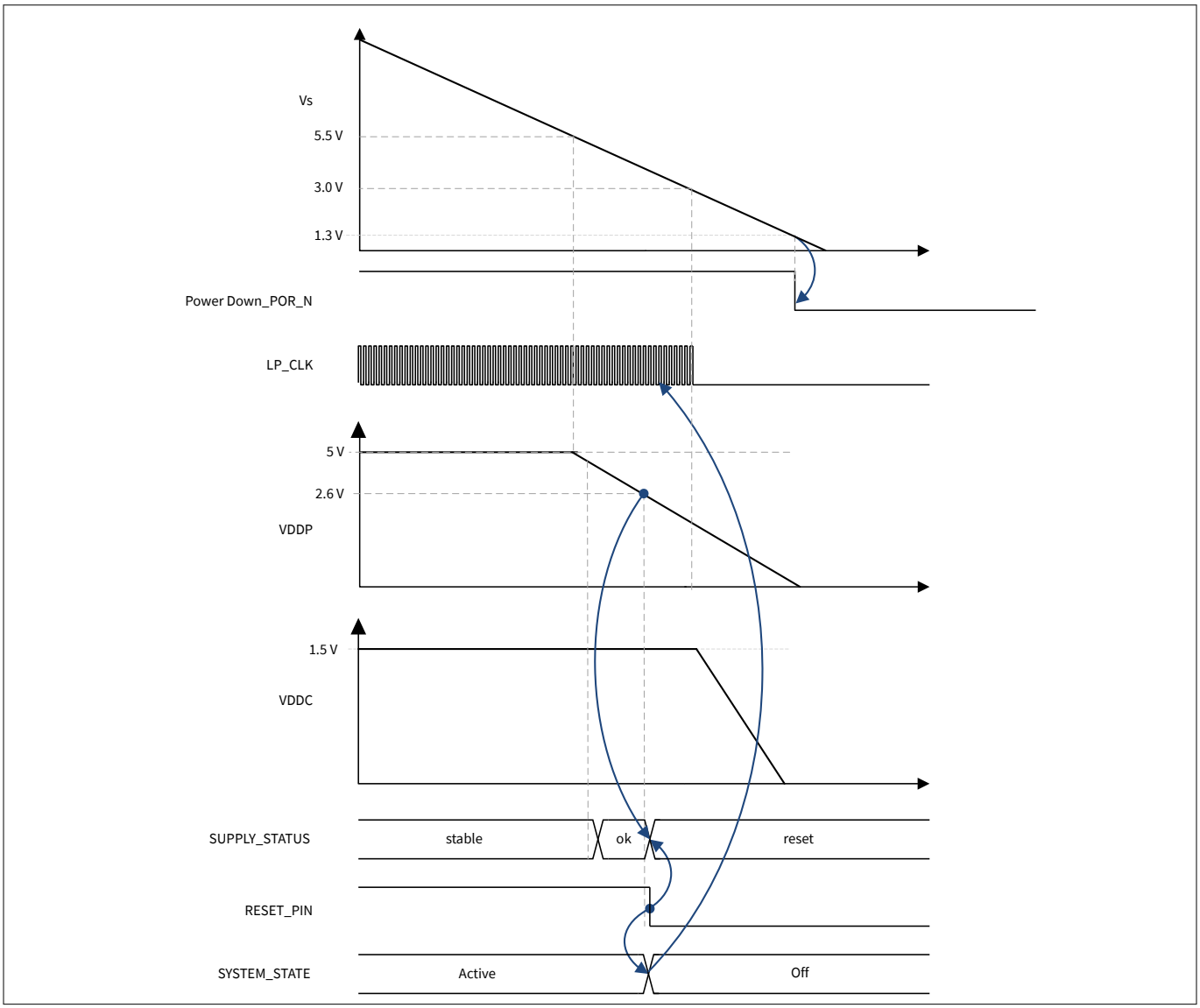


Figure 14 Power-down and power loss sequence

5.4.2 Active mode

In Active mode the reset of the embedded MCU is released and the application software takes control of the system. In this mode the PMU is responsible for supplying and supervising the embedded system.

The supervision functionality of the PMU monitors the output voltage/current of the generated supplies and the status information of the system watchdog (WDT1).The behavior in case of failure events coming from these supervision functionalities is described in Chapter 5.4.6.1.

From Active mode, Sleep and Stop modes can only be set by the user's software. The user software has to write the correspondent value in the power mode control register (SCU_PMCON0):

- For Stop mode: SCU_PMCON0.PD=1
- For Sleep mode: SCU_PMCON0.SL=1

5.4.3 Stop mode

The objective of the Stop mode is to provide a data retention feature for the embedded MCU and the special function registers (XSFRs) while reducing current consumption.

In Stop mode, VDDC goes by default from 1.5 V to reduced voltage. This feature can be disabled by mean of the register PMU_SLEEP.EN_0V9_N. 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 supervision functionality of the PMU monitors the output voltage/current of the generated supplies. The behavior in case of failure events coming from these supervision functionalities is described in Chapter 5.4.6.1.

The wake-up from Stop mode sources are listed in Table 10.

Table 10 Stop mode wake-up sources

Wake-up source	Condition	Activation	Configuration
Cyclic wake	Elapsed time	Continuous	Chapter 5.6.2
MONx	Rising or falling edge	Continuous or cyclic sense	Chapter 24
P1.x	Rising or falling edge	Continuous or cyclic sense	Chapter 5.7
LIN	LIN dominant pulse	Continuous	Chapter 5.7
RESET (pin 21)	Pin pulled down	Continuous	Reset sources
VDDEXTuv	VDDEXT undervoltage	Continuous	Not configurable
VDDEXTot	VDDEXT overtemperature	Cyclic sense[1)]	Not configurable

1) VDDEXTot wake is enabled only during:

- Cyclic sense operation, if VDDEXT is enabled as cyclic sense supply
- Transition to and from Stop mode

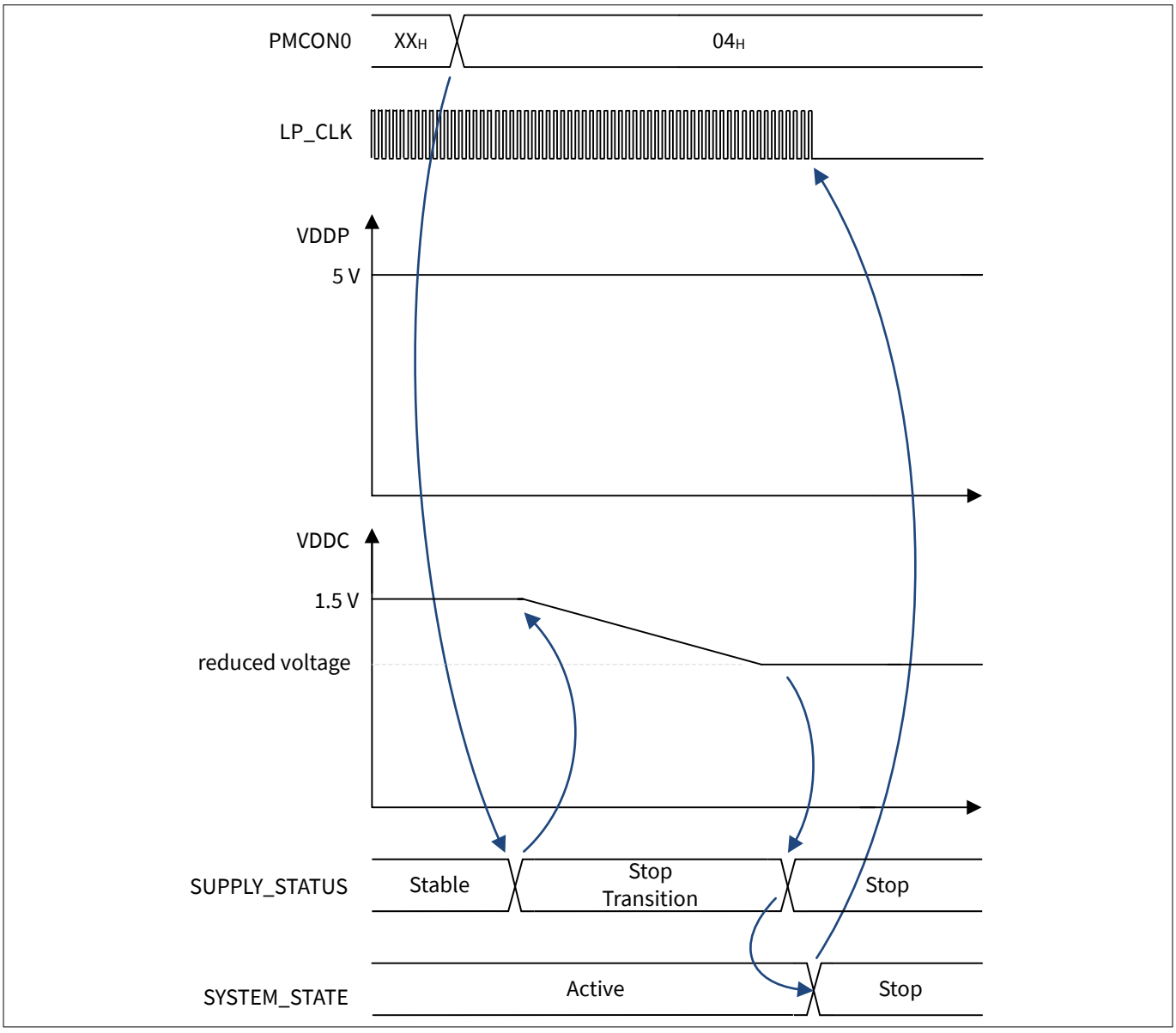


Figure 15 Stop mode entry timing

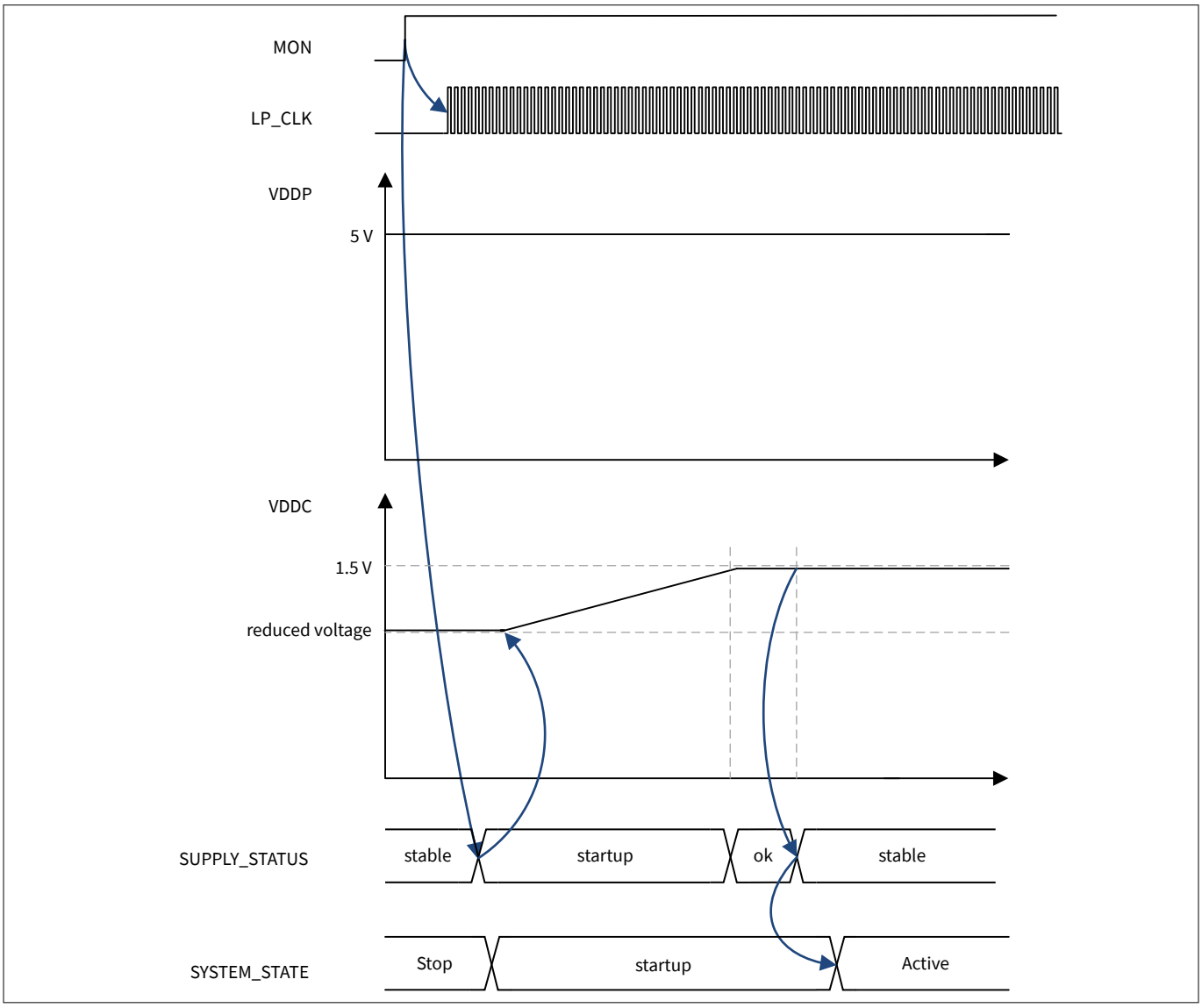


Figure 16 Stop mode wake-up timing

5.4.4 Stop wake

Stop wake is a transition state during which the PGU gets prepared to enter Active mode. If VDDC was operating on reduced voltage it will be ramped up to the default 1.5 V.

If within a timeout of 280 µs (based on LP_CLK) the supplies will not reach stable operation, the PCU will restart the system from the start-up phase.

5.4.5 Sleep mode

The Sleep mode is the power saving mode with the lowest power consumption. In this mode the PMU switches off all voltage supplies (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 17 and Figure 18 show the Sleep mode entry and exit procedures.

The Sleep mode can be exit by the events described in Table 11.

Table 11 Sleep mode wake-up sources

Wake-up source	Condition	Activation	Configuration
Cyclic wake	Elapsed time	Continuous	Chapter 5.6.2
MONx	Rising or falling edge	Continuous or cyclic sense	Chapter 24
LIN	LIN dominant pulse	Continuous	Chapter 5.7

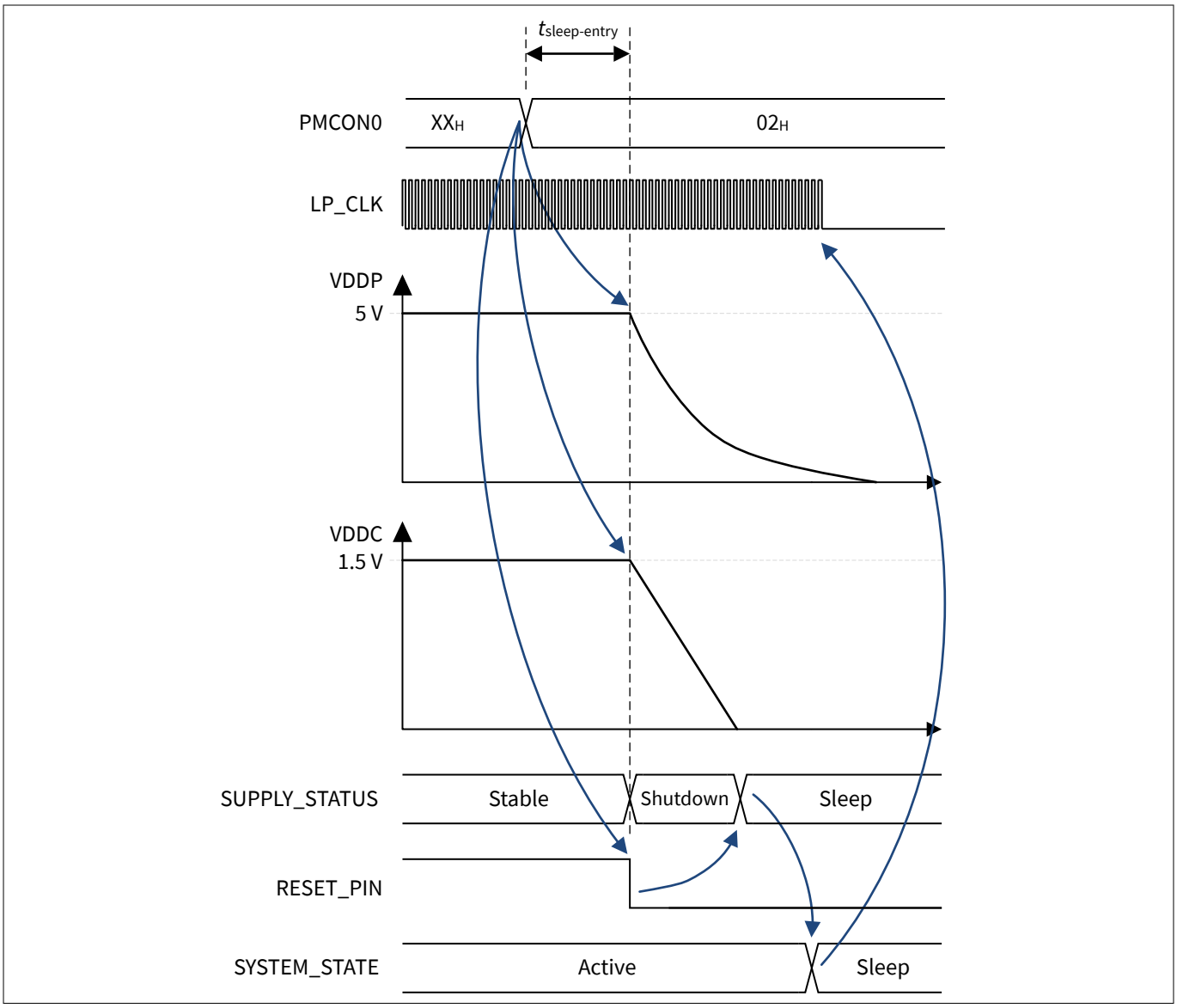


Figure 17 Sleep mode entry timing

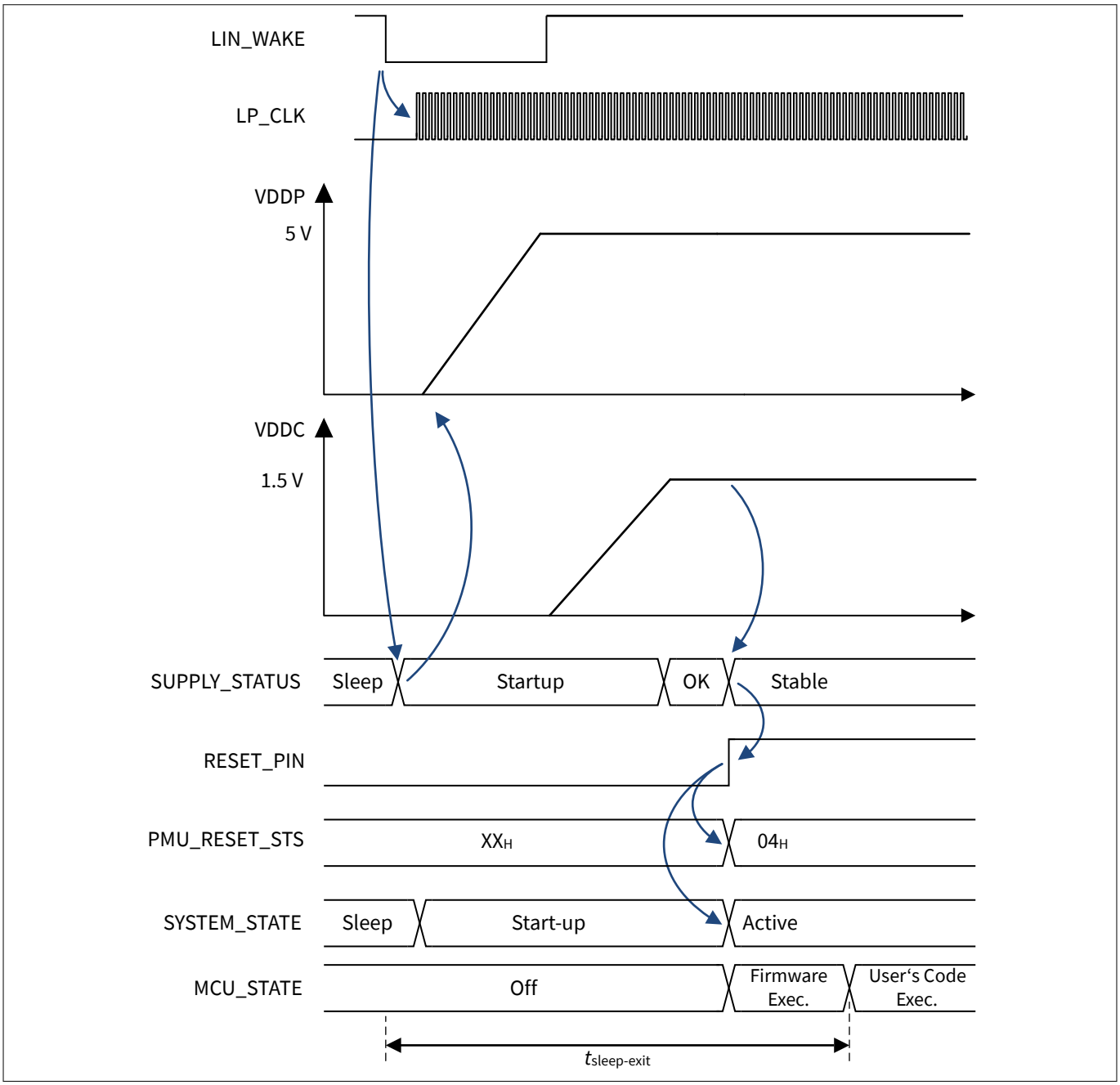


Figure 18 Sleep mode LIN wake-up timing

5.4.6 Fail Sleep mode

From power consumption and system start-up point of view, Fail Sleep and Sleep mode are equivalent. The main differences are the type of events that will trigger this mode, so called Fail safe scenarios. These will be described in Chapter 5.4.6.1.

Another peculiarity is that because of the different reset types executed in this mode, the wake-up sources are slightly different from Sleep mode. The Fail Sleep wake-up sources are listed in Table 12.

Table 12 Fail Sleep mode wake-up sources

Wake-up source	Condition	Activation	Configuration
Cyclic wake	Elapsed time	Continuous	Enabled by default: Chapter 5.6.2
MONx	Rising or falling edge	Continuous or cyclic sense	Enabled by default: Chapter 24

5.4.6.1 Fail safe scenarios

The PCU handles several hardware/software supervision features which enable the application to detect a failure situation.

Eventually, the PCU is able to trigger Fail Sleep mode, which allows low current consumption and predictable system behavior. This is possible from any state, except Power-off, Fail sleep and Sleep.

The failure events that have triggered Fail Sleep mode can be recognized at a successful wake up by observing the content of the PMU_WFS register. Furthermore, at every Fail Sleep entry a reset is performed (see Chapter 5.5) and will be signaled in the flag PMU_RESET_STS.SYS_FAIL at the next successful start-up.

Following are the descriptions of the supervision events that can trigger a transition to Fail Sleep mode. In the following table a summary of these events and their flags in the PMU_WFS register are listed

Table 13 Fail safe scenarios

Trigger	Description	PMU_WFS flag	Occurrence
shortSupp	VDDP/VDDC supply error	SUPP_SHORT	5
ExtWDT	Watchdog timer WDT1 error	WDT1_SEQ_FAIL	5
Overtemp	System overtemperature	SYS_OT	1
Overload (VDDP)	VDDP overcurrent	PMU_5V_OVL	1
Overload (VDDC)	VDDC overcurrent	PMU_1V5_OVL	1
LPwdt	LP_CLK clock watchdog error	LP_CLKWD	1

5.4.6.1.1 VDDP/VDDC supply error

The power supervision feature of the PCU is mainly responsible for monitoring the voltage regulators VDDP and VDDC. In case of voltage regulator malfunction (shortSupp in Figure 12), the PCU restarts the voltage regulators (VDDP and VDDC) by entering the start-up phase. Each time this happens a dedicated supply error counter (cSupp in Figure 12) is incremented.

5.4.6.1.2 Watchdog timer WDT1 error

The PCU supervises the failure information of the system watchdog (WDT1). In case the watchdog is not serviced or serviced in a wrong way (ExtWDT in Figure 12) the MCU is reset and a dedicated WDT1 error counter (cWDT1 in Figure 12) is increased by one (only if cWDT1 < 5). Until cWDT1 < 5, the system stays in the Active mode and after a reset the application software takes over control.

5.4.6.1.3 WDT1 and supply error counters

If the cSupp reaches the value 5, the PCU supervision function will set the device into Fail Sleep mode (instead of restarting the system from start-up). After a successful wake-up from Fail Sleep mode the user can recognize the occurred failure scenario by checking the corresponding PMU_WFS.SUPP_SHORT flag.

If cWDT1 reaches the value 5, the PCU sends the embedded system to Fail Sleep mode (instead of just issuing a reset while in Active mode). If the system can be successfully restarted, this cause of failure can be recognized by reading the PMU_WFS.WDT1_SEQ_FAIL flag.

Both the WDT1 and supply error counters are automatically cleared when the system enters the following states:

- Start-up
- Fail Sleep
- Sleep
- Stop

5.4.6.1.4 VDDP/VDDC overcurrent

Another internal supervision feature of the PCU is to monitor the current sourced by the regulators VDDP and VDDC. In case an overcurrent detection occurs on VDDP and/or VDDC (Overload in Figure 12), the PCU will send the system in Fail Sleep mode.

When the overcurrent condition is gone, a wake-up can be invoked. After a successful wake-up the user can recognize the occurred failure scenario by checking the corresponding PMU_WFS.PMU_1V5_OVL and/or PMU_WFS.PMU_5V_OVL flag.

5.4.6.1.5 System overtemperature

The PCU handles the supervision of the system overtemperature that comes from the ADC2 channel 6 post-processing unit.

The ADC2 channel 6 measures the voltage of the system internal temperature sensor. This temperature sensor includes two sensing elements which monitor the chip and PMU regulator temperatures. Details about their working operation can be found in Chapter 21.5. In Table 162 the default working operation and thresholds of the ADC2 ch6 can be identified.

If the ADC2 ch6 upper threshold is triggered, MF_TEMPSENSE_CTRL.SYS_OT_STS will be set. Also an overtemperature condition will be triggered in the PCU (Overtemp in Figure 12). This will start a transition to Fail Sleep mode.

When the overtemperature condition is gone, the system can be successfully restarted from this state. This cause of failure can be recognized by reading the PMU_WFS.SYS_OT flag.

5.4.6.1.6 LP_CLK clock watchdog

The PMU supervises also the correct functionality of its internal clock LP_CLK. In case of fail of this one, the PCU issues a transition (LPwdt in Figure 12) to Fail Sleep. After a successful wake this cause of failure can be recognized by reading the PMU_WFS.LP_CLKWD flag.

5.4.7 Power control unit registers

The PMU_WFS register is dedicated for the control of the PMU peripherals.

The registers are addressed wordwise.

5.4.7.1 Register overview - Power Control Unit registers (ascending offset address)

Table 14 Register overview - Power Control Unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_HIGHSIDE_CTRL	High-side control register	$005C_H$	83
PMU_WFS	WFS system fail register	0070_H	84

5.5 Reset management unit (RMU)

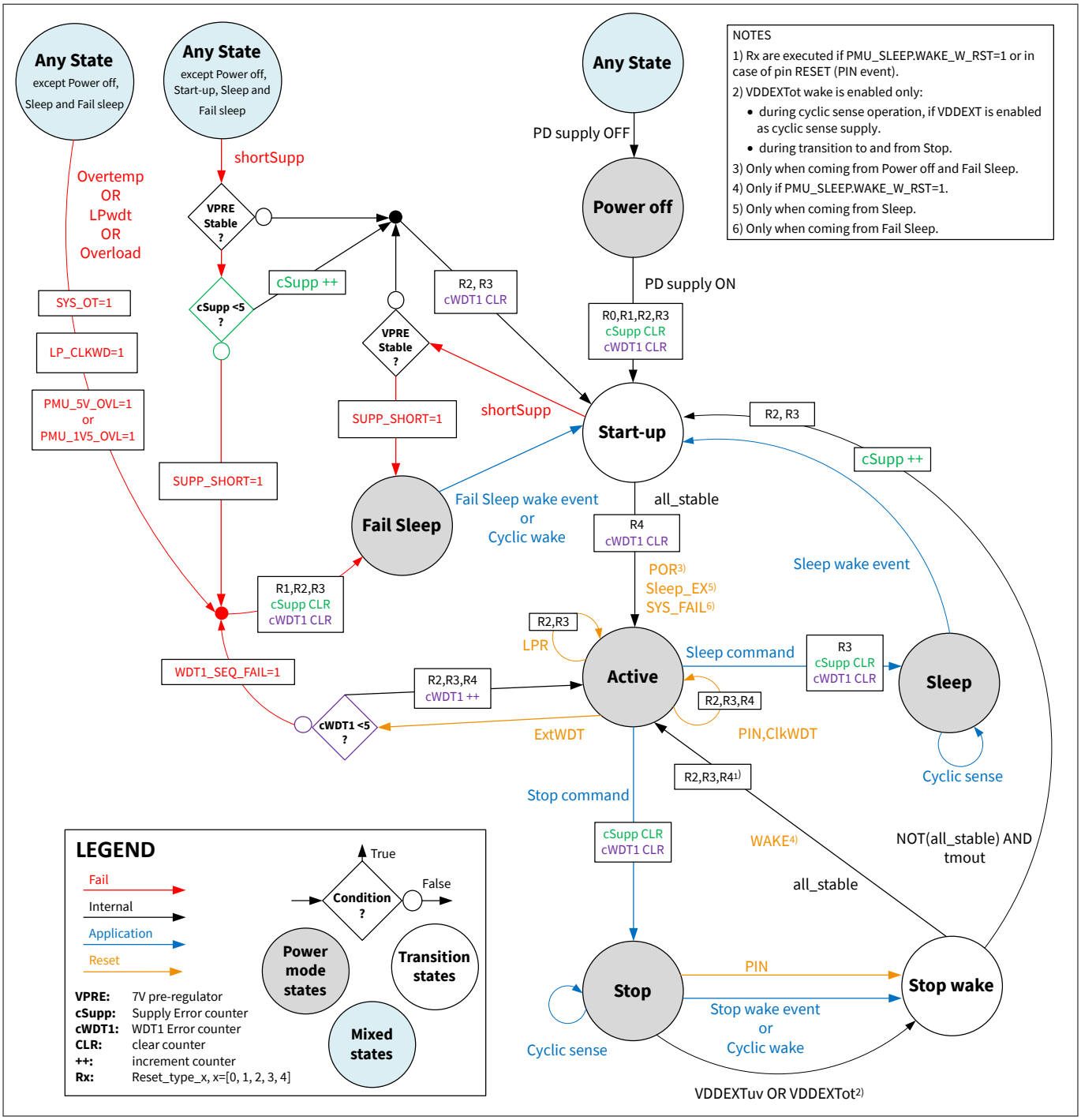


Figure 19 Power control unit state diagram (same as Figure 12)

The RMU, together with the PCU, controls the reset behavior of the entire device. Indeed depending on the events that trigger the PCU state transitions, different reset sources are triggered. Furthermore, different reset sources will result in one or more reset types. The reset types will eventually perform the reset to the reset value of the SoC SFRs.

Which events and which reset types are triggered by the PCU state machine can be determined in Figure 19 and Figure 20.

In general, the triggered reset sources are indicated by the flags in the register PMU_RESET_STS. This register is initialized by RESET_TYPE_1.

	PMU_VS_POR SoC power-on	PMU_SleepEx Wake-up from Sleep Mode	PMU_PIN Hardware reset	PMU_ExtWDT Watchdog timer WDT1	PMU_SOFT Software reset	SYS_FAIL System fail	PMU_ClkWDT Clock watchdog	PMU_Wake Wake-up from Stop Mode [1]	LOCKUP Lockup reset
RESET_TYPE_0	X								
RESET_TYPE_1	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	

1) Only if PMU_SLEEP.WAKE_W_RST = 1.

Figure 20 Reset types issued by the different reset sources

5.5.1 Reset sources

5.5.1.1 Power-on reset

Power-on reset, or POR, is the most powerful reset source. It will issue indeed all the reset types (see Figure 20).

In Chapter 5.4.1, the sequence that triggers this reset is described. This sequence is illustrated in Figure 19. It is important to notice that a POR is performed before entering Active mode and only when coming from Power-off or Fail Safe. This event is signaled by the flag PMU_VS_POR.

5.5.1.2 System fail

This reset source is triggered when any of the fails safe scenarios described in Chapter 5.4.6.1 happens. This event is signaled by the flag SYS_FAIL.

5.5.1.3 Wake-up from Stop mode (with reset)

When a wake-up from Stop mode is performed and the option PMU_SLEEP.WAKE_W_RST is set, a reset source is triggered. This event is signaled by the flag PMU_WAKE.

If the RESET pin is used as wake-up source from Stop mode, a reset is always executed (see Chapter 5.5.1.8).

5.5.1.4 Wake-up from Sleep mode

When a wake-up from Sleep mode is performed a reset source is triggered. This event is signaled by the flag PMU_SleepEX.

5.5.1.5 Low priority resets - software reset or lockup

The software reset can be only triggered by the user software by setting the PMU_LPR CPU_AIRCR.SYSRESETREQ bit. The software related reset is executed within two MCU clock cycles required

by the CPU architecture. 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 to its internal system clock. When this reset source is triggered the PMU_SOFT flag is set.

The lockup reset is caused by a software fault (i.e. Hardfault) and is triggered by the Arm® Cortex®-M0 core[3]. This reset can be enabled by the bit SCU_RSTCON.LOCKUP_EN. When this reset source is triggered, the PMU_RESET_STS.LOCKUP flag is set.

Both these reset sources can only be triggered in active mode. From the system point of view both of these reset sources have the lowest priority. If any of these low priority resets is triggered the PMU_LPR flag is triggered.

5.5.1.6 Clock watchdog

This reset source is triggered by the failure of any of the three clock watchdog CLKWDT1, CLKWDT2, CLKWDT3 implemented in the clock watchdog unit (CWU). For more information see Chapter 7.3. When this reset source is triggered, the flag PMU_ClkWDT is set.

5.5.1.7 Watchdog timer (WDT1)

When WDT1 is not serviced correctly for less than 5 times in a row (see Chapter 5.4.6.1) a reset source will be triggered. Similarly to the LPR. This can happen only in Active mode. When this reset source is triggered, the flag PMU_ExtWDT is set.

5.5.1.8 Hardware reset (RESET pin)

When the RESET pin is pulled down from an external hardware resource (see Chapter 5.5.2 for more details) a reset source is triggered. This can happen only when the system is in Active or Stop mode.

If triggered during Active mode, a reset will be performed and the system will stay in Active mode. This even is signaled by the PMU_PIN flag.

When triggered during Stop mode, the PCU handles it as wake-up source. At the end of the successful wake-up from Stop mode, a reset will be performed. This event is shown in the flags PMU_PIN and PMU_WAKE.

3 For more information on lockup, please refer to the Arm® Cortex®-M0 core documentation.

5.5.2 RESET pin

The RESET pin (pin 21) is a port with internal pull-up supplied by VDDP, hence it is operational only in Active and Stop mode. It works both as an output and input port, supporting different functionalities:

- Input: trigger a reset from an external hardware resource (Hardware reset) by pulling down the pin for a time longer than T_{filt_RESET} (see product specific datasheet)
- Output: signal internal reset events (during Active and Stop mode) to external hardware

In order to avoid any reset deadlock situation no hardware pin reset will be recognized during the blind time set by the RST_TFB register. The functionality of the reset blind time is shown in Figure 21.

Note: A capacitive load connected to the RESET pin (pin 21) will extend the rising and fall time of the pin voltage. If an internal reset is triggered and the rising time is much longer than the blind time, a Hardware reset could be mistakenly triggered.

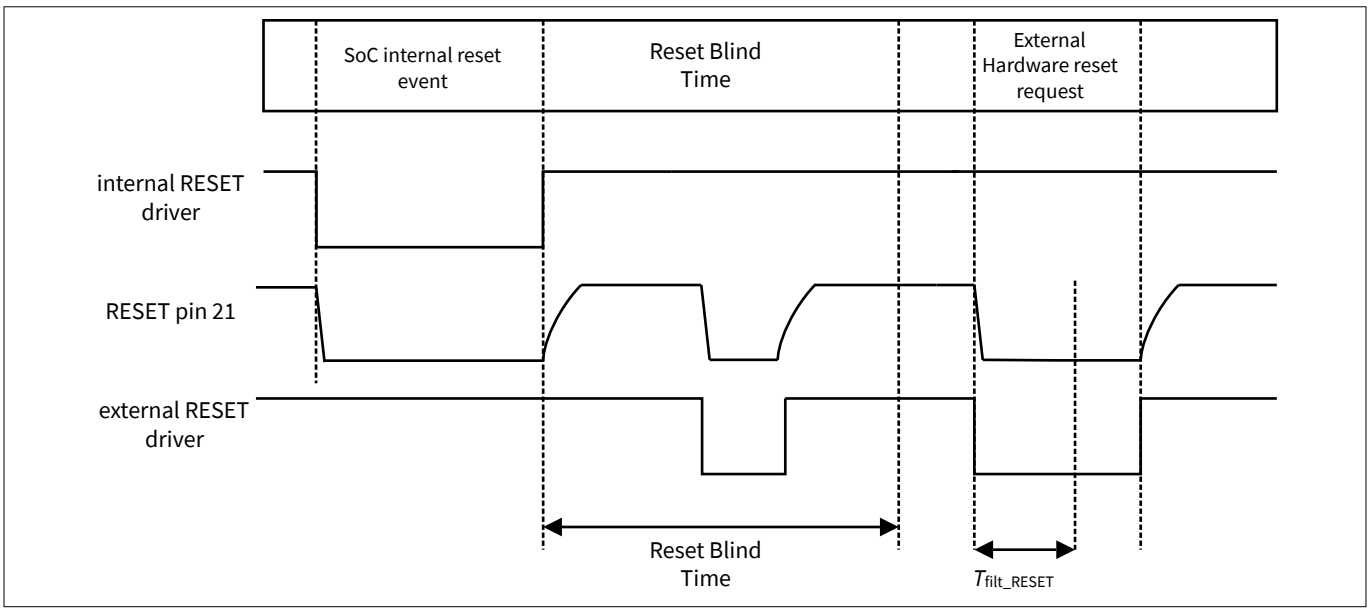


Figure 21 Reset blind time

5.5.3 Reset management unit (RMU) registers

The registers are addressed wordwise.

5.5.3.1 Register overview - Reset Management Unit registers (ascending offset address)

Table 15 Register overview - Reset Management Unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_RESET_STS	Reset status register	0010_H	97
PMU_CNF_RST_TFB	Reset blind time register	$006C_H$	99

5.6 Cyclic management unit (CMU)

The cyclic management unit is responsible for controlling the timing sequence in cyclic sense or cyclic wake operation. The unit operates with the LP_CLK2 clock.

5.6.1 Cyclic sense mode

To select a dedicated MONx pin for cyclic sense mode, the bit MONx_CYC must be set in the corresponding PMU_MON_CNF1.MON[1-4]_CYC and PMU_MON_CNF2.MON5_CYC (product variant dependent) register. In this configuration the wake-up information of this MONx pin is only accepted during the sensing time where the HS_CYC_ON (internal HSx_ON gating signal) is high (see Figure 22). The sensing time where the enable signal is active, will be set in the PMU_SLEEP register. The flags inside PMU_SLEEP are used to configure the dead time (T_{Dead}). PMU_SLEEP.CYC_SENSE_S_DEL is used to program the sample delay of the wake inputs and thus the on-time (T_{On}).

After a valid wake-up event the start-up sequence is similar to the asynchronous wake-up and the system enters the start-up mode automatically too. If the PMU detects a wake-up during cyclic sense then the enable signal of the current source (HS) stays active as long the application software does not disable these signals.

The following figure illustrates the principle of the cyclic sense mode. Here a high-side switch is used as current source together with a MONx pin as a wake-up source. The same timing flow can also be applied for cyclic operation with VDDEXT and all GPIOs from port 1.

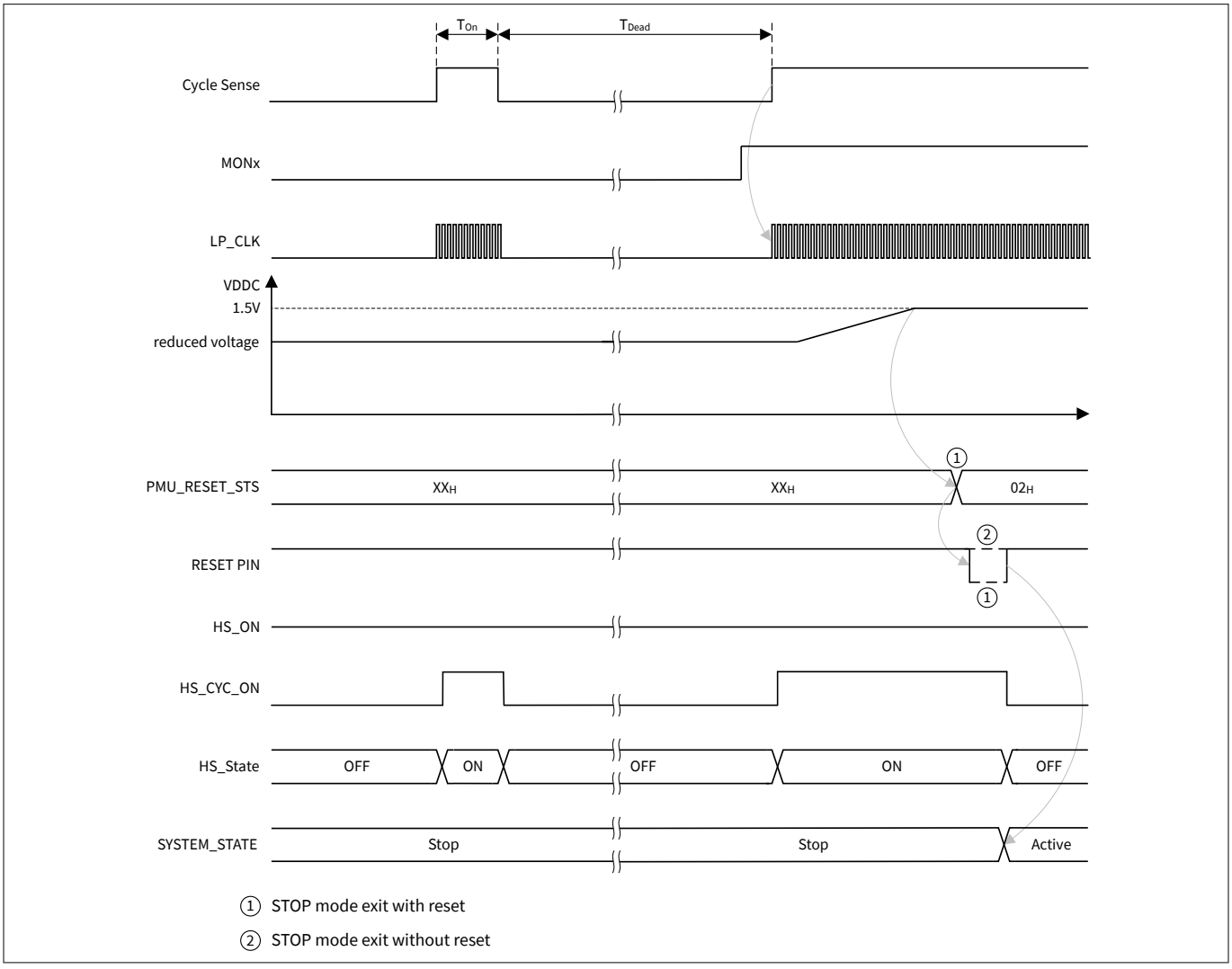


Figure 22 Timing diagram for cyclic sense

5.6.1.1 Configuration of cyclic sense mode

The configuration of cyclic sense mode is shown in the following figure:

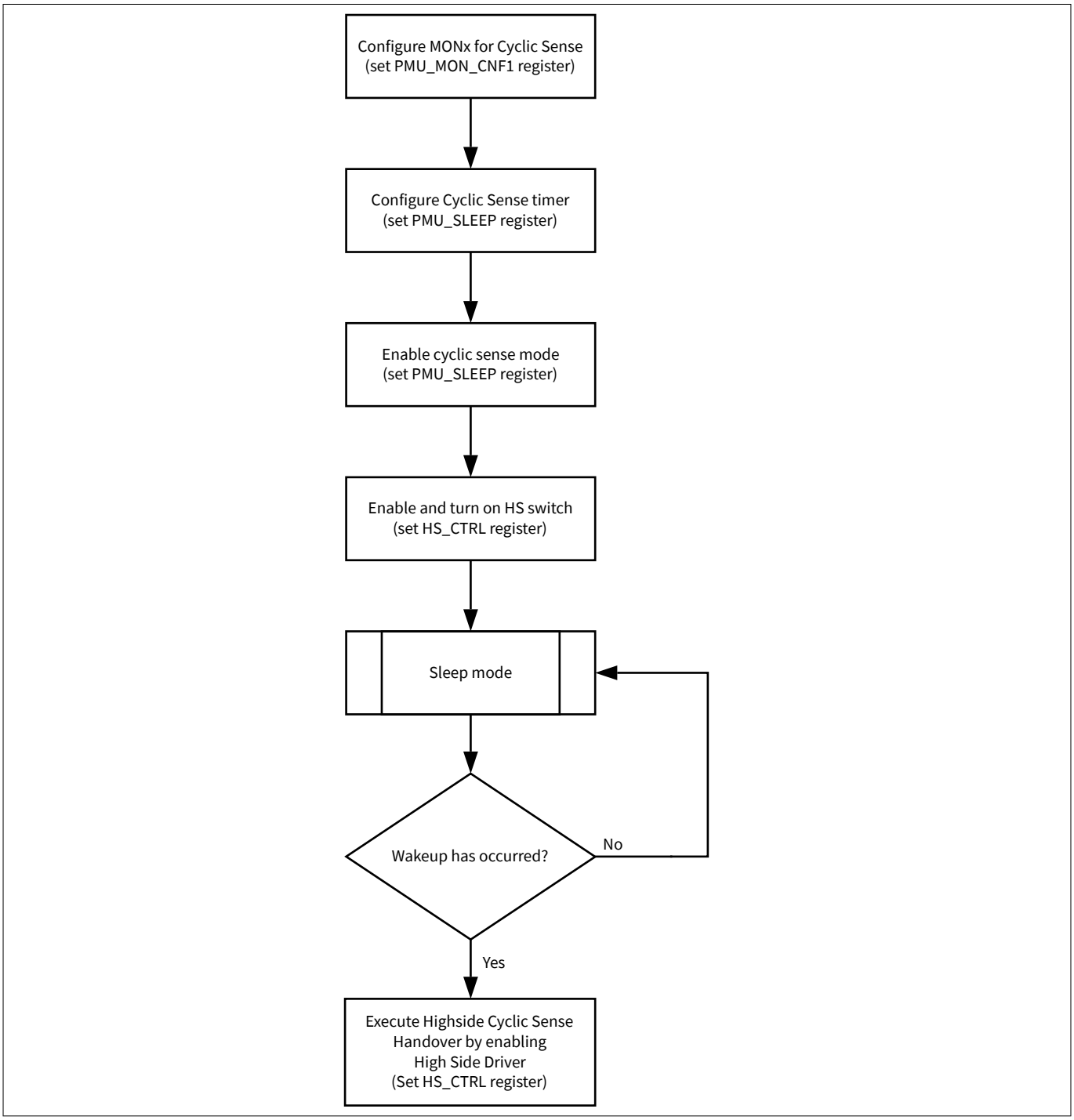


Figure 23 Configuration flow of cyclic sense mode

5.6.2 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 is elapsed the PMU enters the start-up mode and proceeds to active mode where the software takes over the system control. The cyclic wake interval is set in the PMU_SLEEP-XSFR.

5.6.3 Cyclic management unit (CMU) registers

The registers are addressed wordwise.

5.6.3.1 Register overview - Cyclic Management Unit registers (ascending offset address)

Table 16 Register overview - Cyclic Management Unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_SLEEP	PMU sleep behavior register	0020_H	95

5.6.3.2 Cyclic mode configuration registers (CYCMU)

Cyclic sense mode configuration

The off time (dead time) in cyclic sense mode is calculated by following formula:

$$4^{(E1E0)} \times (M3M2M1M0 + 1) \times 2\ \text{ms} \quad (1)$$

where E1E0 represent the exponent, which can be configured by the register bits PMU_SLEEP.CYC_SENSE_E01 <1:0>. M3M2M1M0 represent the mantissa configurable by the register bits PMU_SLEEP.CYC_SENSE_M03. With this setting a time range between

- minimum 2 ms and
- maximum 2048 ms

can be configured. In addition to the off time (dead time) a sample delay for the sensing period can be configured. The sample delay applies after the corresponding supply (HS/VDDEXT) used in the cyclic mode is turned on to the sensing window, where the wake inputs (MONx/GPIOx) are sensed. The delay time can be configured in the PMU_SLEEP.CYC_SENSE_S_DEL register. The sensing window is fixed to typ. 10 us.

Cyclic wake mode configuration

The off time (dead time) in cyclic wake mode is calculated by following formula:

$$4^{(E1E0)} \times (M3M2M1M0 + 1) \times 2\ \text{ms} \quad (2)$$

where E1E0 represent the exponent, which can be configured by the register bits PMU_SLEEP.CYC_WAKE_E01 <1:0>. M3M2M1M0 represent the mantissa configurable by the register bits PMU_SLEEP.CYC_WAKE_M03. With this setting a time range between

- minimum 2 ms and
- maximum 2048 ms

can be configured.

Note: All timings in the cyclic modes are derived from LP_CLK2. The values used in the register description are typical values. Their variation is depending on the variation of LP_CLK2.

5.7 Wake-up management unit (WMU)

The wake-up management unit (WMU) is mainly responsible for handling the wake-up events on *LIN*, *HV*-monitoring inputs (*MON1* to *MON4* or *MON5*, product variant dependent), hardware reset and all GPIOs belonging to port 1. Following wake scenarios are possible:

- Wake-up over GPIO port 1 pins: They can be configured for rising edge triggered and falling edge triggered wake-up events. This configuration can be used to wake-up the device from normal stop mode and stop mode with cyclic sense option. To bias the GPIOs, VDDEXT as current source can be used. 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. The wake-up feature from sleep mode is not possible.
- Wake-up over *MON1* to *MON4* or *MON5* (product variant dependent) pins: The MONx pins can be configured for rising edge triggered and/or falling edge triggered wake-up events. This setup can be used to wake-up the device from stop mode with or without cyclic sense, but also a wake-up from sleep mode with or without cyclic sense is possible
- Wake-up over *LIN* pin: Is a normal wake-up source and has no configuration possibilities
- Wake-up on VDDEXT fail from stop mode: Will be performed in case of VDDEXT failures described in Chapter 5.3.3.1.

Notes:

1. *GPIO port 2 pins cannot invoke any wake-up.*

2. *None of the GPIOs is supplied during sleep mode, therefore wake-up is not possible through them.*

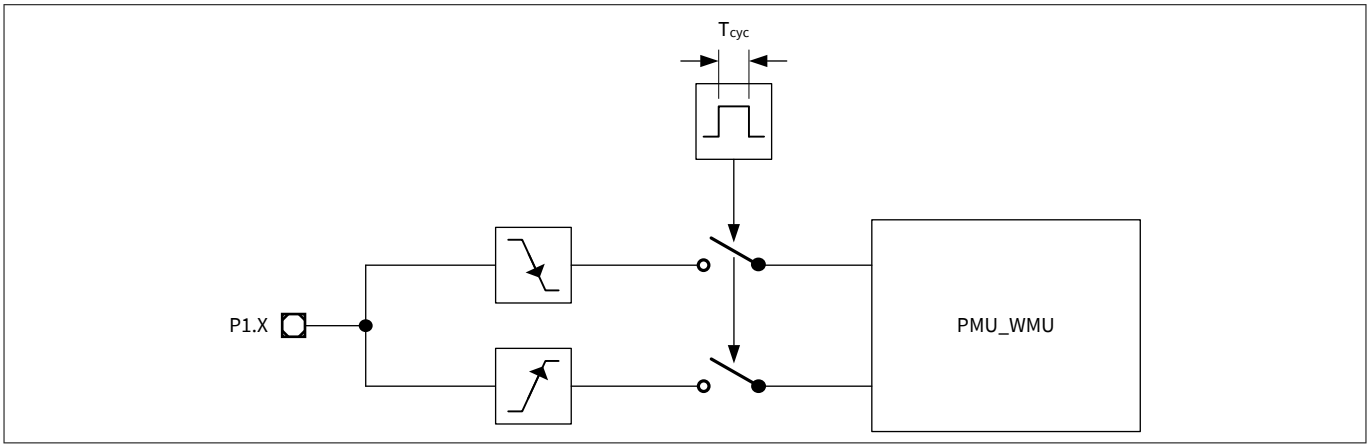


Figure 24 Block diagram of wake-up management unit in cyclic sense mode with VDDEXT

5.7.1 Wake-up management unit (WMU) registers

The registers listed below are for wake-up control of all wake-up capable general purpose inputs outputs.
The WMU is fully controllable by these SFR registers.
The PMU_WAKE_CNF_GPIO1 register is dedicated for the control of the PMU peripherals.
The registers are addressed wordwise.

5.7.1.1 Register overview - Wake-up Management Unit registers (ascending offset address)

Table 17 Register overview - Wake-up Management Unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_WAKE_STATUS	Main wake status register	0000_H	86
PMU_GPIO_WAKE_STATUS	GPIO port wake status register	0004_H	89
PMU_LIN_WAKE_EN	LIN wake enable register	0050_H	91
PMU_CNF_WAKE_FILTER	PMU wake-up timing register	$00AC_H$	92
PMU_WAKE_CNF_GPIO1	Wake configuration GPIO port 1 register	$00CC_H$	93

5.8 PMU data storage area

The PMU provides the possibility for the system to store data in registers which will retain their values, when the device is set to sleep mode. In sum there are 12 x 8 bit available.

5.8.1 PMU data storage registers

The registers are addressed wordwise.

5.8.1.1 Register overview - PMU data storage area registers (ascending offset address)

Table 18 Register overview - PMU data storage area registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_GPUDATA0to3	General purpose user DATA0to3 register	$00C0_H$	100
PMU_GPUDATA4to7	General purpose user DATA4to7 register	$00C4_H$	101
PMU_GPUDATA8to11	General purpose user DATA8to11 register	$00C8_H$	102

5.9 Power management unit (PMU) register definition

Note: HS2 and MON5 are device variant specific. In devices featuring only HS1 the HS2_XXX bitfields can be ignored. In devices featuring only MON1-4 the HS MON5_XXX bitfields can be ignored. Writing to these bitfields has no effect.

The registers are addressed wordwise.

5.9.1 Register address space - PMU

Table 19 Registers address space - PMU

Module	Base address	End address	Note
PMU	50004000_H	$50004FFF_H$	Power Management Unit registers

5.9.2 Register overview - PMU (ascending offset address)

Table 20 Register overview - PMU (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_WAKE_STATUS	Main wake status register	0000_H	86
PMU_GPIO_WAKE_STATUS	GPIO port wake status register	0004_H	89
PMU_SUPPLY_STS	Voltage reg status register	0008_H	79
PMU_VDDEXT_CTRL	VDDEXT control register	$000C_H$	81
PMU_RESET_STS	Reset status register	0010_H	97
PMU_SLEEP	PMU sleep behavior register	0020_H	95
PMU_MON_CNF1	Settings monitor 1-4 register	0034_H	103
PMU_MON_CNF2	Settings monitor 5 register	0038_H	107
PMU_LIN_WAKE_EN	LIN wake enable register	0050_H	91
PMU_HIGHSIDE_CTRL	High-side control register	$005C_H$	83
PMU_CNF_RST_TFB	Reset blind time register	$006C_H$	99
PMU_WFS	WFS system fail register	0070_H	84
PMU_CNF_WAKE_FILTER	PMU wake-up timing register	$00AC_H$	92
PMU_GPUDATA0to3	General purpose user DATA0to3 register	$00C0_H$	100
PMU_GPUDATA4to7	General purpose user DATA4to7 register	$00C4_H$	101
PMU_GPUDATA8to11	General purpose user DATA8to11 register	$00C8_H$	102
PMU_WAKE_CNF_GPIO1	Wake configuration GPIO port 1 register	$00CC_H$	93

5.9.3 Voltage reg status register

The PMU_SUPPLY_STS register shows the overvoltage and overload condition of VDDP and VDDC. To use this information as interrupt sources it must be selected explicitly in this register.

PMU_SUPPLY_STS Offset address: 0008_H

Voltage reg status register Reset values see: Table 21

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									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
PMU_1V5_OVERVOLT	0	r	**Overvoltage at VDDC regulator** *Note: This flag is automatically cleared, if error condition is removed.* 0_B **No_overvoltage**: No overvoltage 1_B **Overvoltage**: Overvoltage
PMU_1V5_OVERLOAD	1	r	**Overload at VDDC regulator** *Note: If this flag is set and an additional filter time of 290 us (typ.) is passed the system will be put to sleep mode. This flag is automatically cleared, if error condition is removed.* 0_B **No_overload**: No overload 1_B **Overload**: Overload
PMU_1V5_FAIL_EN	2	rw	**Enabling of VDDC status information as interrupt source** 0_B **No_interrupts**: No interrupts are generated 1_B **Interrupts**: Interrupts are generated
RES	3, 31:7	r	**Reserved** Always read as 0
PMU_5V_OVERVOLT	4	r	**Overvoltage at VDDP regulator** *Note: This flag is automatically cleared, if error condition is removed.* 0_B **No_overvoltage**: No overvoltage 1_B **Overvoltage**: Overvoltage
PMU_5V_OVERLOAD	5	r	**Overload at VDDP regulator**

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: If this flag is set and an additional filter time of 290 us (typ.) is passed the system will be put to sleep mode. This flag is automatically cleared, if error condition is removed. 0_B **No_overload**: No overload 1_B **Overload**: Overload
PMU_5V_FAIL_EN	6	rw	**Enabling of VDDP status information as interrupt source** 0_B **No_interrupts**: No interrupts are generated 1_B **Interrupts**: Interrupts are generated

Table 21 Reset values of **PMU_SUPPLY_STS**

Reset type	Reset value	Note
RESET_TYPE_3	$0000\ 0000_H$	Reset mask: 0x00000044
RESET_TYPE_0	$0000\ 0000_H$	Reset mask: 0x000000BB

5.9.4 VDDEXT control register

PMU_VDDEXT_CTRL Offset address: $000C_H$

VDDEXT control register Reset values see: Table 22

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		VDDEXT_OT_SC	VDDEXT_UV_ISC	VDDEXT_OT_ISC	RES			VDDEXT_STABLE	VDDEXT_OT	VDDEXT_OT_STS	VDDEXT_UV_IS	VDDEXT_OT_IS	VDDEXT_FAIL_EN	VDDEXT_CYC_EN	VDDEXT_ENABLE
r		w	w	w	r			r	r	r	r	r	rw	rw	rw

Field	Bits	Type	Description
VDDEXT_ENABLE	0	rw	**VDDEXT supply enable** 0_B **DISABLE**: VDDEXT supply disabled 1_B **ENABLE**: VDDEXT supply enabled
VDDEXT_CYC_EN	1	rw	**VDDEXT supply for cyclic sense enable** *Note: To use VDDEXT supply for cyclic sense the bits VDDEXT_CYC_EN AND VDDEXT_ENABLE must be set.* 0_B **DISABLE**: VDDEXT for cyclic sense disable 1_B **ENABLE**: VDDEXT for cyclic sense enable
VDDEXT_FAIL_EN	2	rw	**Enabling of VDDEXT supply status information as interrupt source** 0_B **DISABLE**: VDDEXT fail interrupts are disabled 1_B **ENABLE**: VDDEXT fail Interrupts are enabled
VDDEXT_OT_IS	3	r	**VDDEXT supply overtemperature interrupt status** 0_B **No_overtemperature**: VDDEXT no overtemperature condition 1_B **Overtemperature**: VDDEXT overtemperature condition
VDDEXT_UV_IS	4	r	**VDDEXT supply undervoltage interrupt status** 0_B **Not_in_undervoltage**: VDDEXT not in undervoltage condition 1_B **In_undervoltage**: VDDEXT in undervoltage condition
VDDEXT_OT_STS	5	r	**VDDEXT supply overtemperature status** 0_B **No_overtemperature**: VDDEXT not in overtemperature condition 1_B **Overtemperature**: VDDEXT in overtemperature condition
VDDEXT_OT	6	r	**VDDEXT supply overtemperature** 0_B **No_overtemperature**: VDDEXT not in overtemperature condition 1_B **Overtemperature**: VDDEXT in overtemperature condition
VDDEXT_STABLE	7	r	**VDDEXT supply stable** 0_B **Not_stable**: VDDEXT not in stable condition 1_B **Stable**: VDDEXT in stable condition

(table continues...)

(continued)

Field	Bits	Type	Description
RES	10:8, 31:14	r	**Reserved** Always read as 0
VDDEXT_OT_ISC	11	w	**VDDEXT supply overtemperature interrupt status clear** 0_B **Not_cleared**: VDDEXTovertemperature not cleared 1_B **Cleared**: VDDEXTovertemperature cleared
VDDEXT_UV_ISC	12	w	**VDDEXT supply undervoltage interrupt status clear** 0_B **Not_cleared**: VDDEXT undervoltage not cleared 1_B **Cleared**: VDDEXT undervoltage cleared
VDDEXT_OT_SC	13	w	**VDDEXT supply overtemperature status clear** 0_B **Status_not_cleared**: VDDEXT overtemperature status not cleared 1_B **Status_cleared**: VDDEXT overtemperature status cleared

Table 22 Reset values of **PMU_VDDEXT_CTRL**

Reset type	Reset value	Note
RESET_TYPE_3	0000 0000_H	ResetMask= “0xFFFFFFE7”
RESET_TYPE_0	0000 0000_H	ResetMask= “0x00000018”

5.9.5 High-side control register

PMU_HIGHSIDE_CTRL Offset address: $005C_H$

High-side control register RESET_TYPE_2 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					HS2_CYC_EN	RES							HS1_CYC_EN	RES	
r					rw	r							rw	r	

Field	Bits	Type	Description
RES	1:0, 9:3, 31:11	r	**Reserved** Always read as 0
HS1_CYC_EN	2	rw	**High-side 1 switch enable for cyclic sense** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
HS2_CYC_EN	10	rw	**High-side 2 switch enable for cyclic sense** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled

5.9.6 WFS system fail register

Note: The register PMU_WFS is cleared when the flag PMU_RESET_SYS.SYS_FAIL is reset.

PMU_WFS Offset address: 0070_H

WFS system fail register RESET_TYPE_0 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								LP_CLKWD	WDT1_SEQ_FAIL	SYS_OT	RES	PMU_5V_OVL	PMU_1V5_OVL	RES	SUPP_SHORT
r								r	rh	rh	r	rh	rh	r	rh

Field	Bits	Type	Description
SUPP_SHORT	0	rh	**Supply short** VDDC or VDDP internal diagnosis, fail safe undervoltage detection. 0_B **No_undervoltage**: VDDP and VDDC correct operation 1_B **Undervoltage**: VDDP or VDDC voltage fail
RES	1, 4, 31:8	r	**Reserved** Always read as 0
PMU_1V5_OVL	2	rh	**VDDC overload flag** Indicates overload condition at VDDC. 0_B **No_overload**: VDDC ok 1_B **Overload**: VDDC overload
PMU_5V_OVL	3	rh	**VDDP overload flag** Indicates overload condition at VDDP. 0_B **No_overload**: VDDP ok 1_B **Overload**: VDDP overload
SYS_OT	5	rh	**System overtemperature indication flag** The ADC2 raises an overtemperature flag when the system temperature exceeds an overtemperature threshold. 0_B **NORMAL**: Normal operation 1_B **OT**: Overtemperature
WDT1_SEQ_FAIL	6	rh	**External watchdog (WDT1) sequential fail** Indicates that watchdog is not serviced 5 times. 0_B **No_fail**: No fail, system working properly 1_B **Sequential_watchdog_fail**: 5 consecutive watchdog fails
LP_CLKWD	7	r	**LP_CLKWD**

(table continues...)

(continued)

Field	Bits	Type	Description
			Indicates an LP_CLK clock failure. 0_B **NORMAL**: Normal operation 1_B **FAIL**: LP_CLK clock failure

5.9.7 Main wake status register

PMU_WAKE_STATUS Offset address: 0000_H

Main wake status register RESET_TYPE_1 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES										RES		RES	VDD EXT_ UV	VDD EXT_ OT	RES
r										r		r	rh	rh	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			MON 5_W AKE_ STS	MON 4_W AKE_ STS	MON 3_W AKE_ STS	MON 2_W AKE_ STS	MON 1_W AKE_ STS	RES	RES	FAIL	CYC_ WAK E	GPIO 1	RES	MON	LIN_ WAK E
r			rh	rh	rh	rh	rh	r	r	rc	rh	rc	r	rc	rh

Field	Bits	Type	Description
LIN_WAKE	0	rh	**Wake-up via LIN- Message** *Note: This register is cleared automatically by read operation.* 0_B **No_wake_up**: No wake-up occurred 1_B **Wake_up**: Wake-up occurred
MON	1	rc	**Wake-up via MON which is a logical OR combination of all Wake_STS_MON bits** 0_B **No_wake_up**: No wake-up occurred 1_B **Wake_up**: Wake-up occurred
RES	2, 6, 7, 16:13, 19, 21:20, 31:22	r	**Reserved** Always read as 0
GPIO1	3	rc	**Wake-up via GPIO1 which is a logical OR combination of all Wake_STS_GPIO1 bits** 0_B **No_wake_up**: No wake-up occurred 1_B **Wake_up**: Wake-up occurred
CYC_WAKE	4	rh	**Wake-up caused by cyclic wake** *Note: This register is cleared automatically by read operation.* 0_B **No_wake_up**: No wake-up occurred 1_B **Wake_up**: Wake-up occurred
FAIL	5	rc	**Wake-up after VDDEXT fail**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **No_wake_up**: No wake-up occurred 1_B **Wake_up**: Wake-up occurred
MON1_WAKE_STS	8	rh	**Status of MON1** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up** : No wake-up detected 1_B **Wake_up**: Wake-up detected
MON2_WAKE_STS	9	rh	**Status of MON2** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
MON3_WAKE_STS	10	rh	**Status of MON3** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
MON4_WAKE_STS	11	rh	**Status of MON4** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
MON5_WAKE_STS	12	rh	**Status of MON5** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
VDDEXT_OT	17	rh	**Wake VDDEXT overtemperature** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up** : Wake-up detected
VDDEXT_UV	18	rh	**Wake VDDEXT undervoltage**

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active. 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up** : Wake-up detected

5.9.8 GPIO port wake status register

PMU_GPIO_WAKE_STATUS Offset address: 0004_H

GPIO port wake status register RESET_TYPE_1 value: $0000\ 0000_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			GPIO1_STS_4	RES	GPIO1_STS_2	GPIO1_STS_1	GPIO1_STS_0	RES							
r			rh	r	rh	rh	rh	r							

Field	Bits	Type	Description
RES	7:0, 11, 19:13, 31:20	r	**Reserved** Always read as 0
GPIO1_STS_0	8	rh	**Wake GPIO1_0** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
GPIO1_STS_1	9	rh	**Wake GPIO1_1** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
GPIO1_STS_2	10	rh	**Wake GPIO1_2** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected 1_B **Wake_up**: Wake-up detected
GPIO1_STS_4	12	rh	**Wake GPIO1_4** *Note: This register is cleared automatically by read operation. The user has to clear this flag before entering power saving modes otherwise the device will stay in active.* 0_B **No_wake_up**: No wake-up detected

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **Wake_up**: Wake-up detected

5.9.9 LIN wake enable register

PMU_LIN_WAKE_EN Offset address: 0050_H

LIN wake enable register RESET_TYPE_2 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								LIN_EN	RES						
r								rw	r						

Field	Bits	Type	Description
RES	6:0, 31:8	r	**Reserved** Always read as 0
LIN_EN	7	rw	**Lin wake enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled

5.9.10 PMU wake-up timing register

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

PMU_CNF_WAKE_FILTER Offset address: $00AC_H$

PMU wake-up timing register RESET_TYPE_2 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												**CNF_GPIO_FT**		**CNF_MON_FT**	**CNF_LIN_FT**
r												rw		rw	rw

Field	Bits	Type	Description
CNF_LIN_FT	0	rw	**Wake-up filter time for LIN WAKE** Selects the filter time for the wake-up. 0_B **30_us**: 30 µs filter time 1_B **50_us**: 50 µs filter time
CNF_MON_FT	1	rw	**Wake-up filter time for monitoring inputs** Selects the filter time for the wake-up. 0_B **20_us**: 20 µs filter time 1_B **40_us**: 40 µs filter time
CNF_GPIO_FT	3:2	rw	**Wake-up filter time for general purpose IO** Selects 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
RES	31:4	r	**Reserved** Always read as 0

5.9.11 Wake configuration GPIO port 1 register

PMU_WAKE_CNF_GPIO1 Offset address: $00CC_H$

Wake configuration GPIO port 1 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											CYC_4	RES	CYC_2	CYC_1	CYC_0
r											rw	r	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			FA_4	RES	FA_2	FA_1	FA_0	RES			RI_4	RES	RI_2	RI_1	RI_0
r			rw	r	rw	rw	rw	r			rw	r	rw	rw	rw

Field	Bits	Type	Description
RI_0	0	rw	**Port 1_0 wake-up on rising edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
RI_1	1	rw	**Port 1_1 wake-up on rising edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
RI_2	2	rw	**Port 1_2 wake-up on rising edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
RES	3, 7:5, 11, 15:13, 19, 31:21	r	**Reserved** Always read as 0
RI_4	4	rw	**Port 1_4 wake-up on rising edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
FA_0	8	rw	**Port 1_0 wake-up on falling edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
FA_1	9	rw	**Port 1_1 wake-up on falling edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
FA_2	10	rw	**Port 1_2 wake-up on falling edge enable** 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
FA_4	12	rw	**Port 1_4 wake-up on falling edge enable**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
CYC_0	16	rw	**GPIO1_0 input for cycle sense enable** 0_B **DISABLE**: Input for cycle sense disabled 1_B **ENABLE**: Input for cycle sense enabled
CYC_1	17	rw	**GPIO1_1 input for cycle sense enable** 0_B **DISABLE**: Input for cycle sense disabled 1_B **ENABLE**: Input for cycle sense enabled
CYC_2	18	rw	**GPIO1_2 input for cycle sense enable** 0_B **DISABLE**: Input for cycle sense disabled 1_B **ENABLE**: Input for cycle sense enabled
CYC_4	20	rw	**GPIO1_4 input for cycle sense enable** 0_B **DISABLE**: Input for cycle sense disabled 1_B **ENABLE**: Input for cycle sense enabled

5.9.12 PMU sleep behavior register

PMU_SLEEP Offset address: 0020_H

PMU sleep behavior register RESET_TYPE_2 value: $0037\ 0004_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES					CYC_SENSE_S_DEL			RES		CYC_WAKE_E01		CYC_WAKE_M03			
r					rw			r		rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	RES	CYC_SENSE_E01		CYC_SENSE_M03				RES	RES		RFU	CYC_SENSE_EN	CYC_WAKE_EN	EN_0V9_N	WAKE_W_RST
r	r	rw		rw				r	r		rw	rw	rw	rw	rw

Field	Bits	Type	Description
WAKE_W_RST	0	rw	**Wake-up with reset execution** Enables the stop-exit with reset execution. 0_B **Without_reset_execution** : Stop-exit without reset execution 1_B **With_reset_execution** : Stop-exit with reset execution
EN_0V9_N	1	rw	**Enables the reduction of the VDDC regulator output to reduced voltage during stop mode** 0_B **ENABLE**: Output voltage reduction enabled 1_B **DISABLE**: Output voltage reduction disabled
CYC_WAKE_EN	2	rw	**Enabling cyclic wake** This bit enables the cyclic wake feature for the power save modes. 0_B **DISABLE**: Cyclic wake disabled 1_B **ENABLE**: Cyclic wake enabled
CYC_SENSE_EN	3	rw	**Enabling cyclic sense** This bit enables the cyclic sense feature for the power save modes. 0_B **DISABLE**: Cyclic sense disabled 1_B **ENABLE**: Cyclic sense enabled
RFU	4	rw	**Reserved for future use** This bit is reserved for future use. 0_B **Writing_a_zero**: Writing a zero has no effect 1_B **Writing_a_one**: Writing a one has no effect
RES	6:5, 7, 14, 15, 23:22, 31:27	r	**Reserved** Always read as 0

(table continues...)

(continued)

Field	Bits	Type	Description
CYC_SENSE_M03	11:8	rw	**Mantissa** Mantissa value is calculated as CYC_SENSE_M03 + 1. 0_H **Mantissa_value_1**: Mantissa value is 1 F_H **Mantissa_value_16**: Mantissa value is 16
CYC_SENSE_E01	13:12	rw	**Exponent** 00_B **Exponent_value_0**: Exponent value is 0 01_B **Exponent_value_1**: Exponent value is 1 10_B **Exponent_value_2**: Exponent value is 2 11_B **Exponent_value_3**: Exponent value is 3
CYC_WAKE_M03	19:16	rw	**Mantissa** Mantissa value is calculated as CYC_WAKE_M03 + 1. 0_H **Mantissa_value_1**: Mantissa value is 1 F_H **Mantissa_value_16**: Mantissa value is 16
CYC_WAKE_E01	21:20	rw	**Exponent** 00_B **Exponent_value_0**: Exponent value is 0 01_B **Exponent_value_1**: Exponent value is 1 10_B **Exponent_value_2**: Exponent value is 2 11_B **Exponent_value_3**: Exponent value is 3
CYC_SENSE_S_DEL	26:24	rw	**Sample delay in cyclic sense mode** Delay time after HS/VDDEXT is turned to beginning of sensing window for MONx/GPIOx. The sensing window is fixed to 10 µs. 000_B **Delay_time_0**: Is 10 µs 001_B **Delay_time_1**: Is 20 µs 010_B **Delay_time_2**: Is 30 µs 011_B **Delay_time_3**: Is 40 µs 100_B **Delay_time_4**: Is 60 µs 101_B **Delay_time_5**: Is 80 µs 110_B **Delay_time_6**: Is 100 µs 111_B **Delay_time_7**: Is 150 µs

5.9.13 Reset status register

The PMU_RESET_STS register shows every executed reset request. The PMU writes the corresponding register bit of an executed reset. To clear the information of the PMU_RESET_STS register the user must overwrite the corresponding bit with a logic zero. The register is reset by RESET_TYPE_1.

The PMU_RESET_STS register shows every executed reset request. The PMU writes the corresponding register bit using settings of the asynchronously set input of the flip-flop. To clear the information of the PMU_RESET_STS register the user must overwrite the corresponding bit with a logic zero.

Note: The register PMU_RESET_STS is cleared when the flag PMU_RESET_STS.PMU_LPR is reset.

PMU_RESET_STS Offset address: 0010_H

Reset status register RESET_TYPE_1 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					LOCKUP	PMU_SOFT	RES	PMU_VS_POR	PMU_PIN	PMU_ExtWDT	PMU_ClkWDT	PMU_LPR	PMU_SleepEX	PMU_WAKE	SYS_FAIL
r					rwh	rwh	r	rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
SYS_FAIL	0	rwh	**Flag which indicates a reset caused by a system fail reported in the PMU_WFS register** 0_B **No_reset**: No reset caused by system fail executed 1_B **Reset**: Reset caused by system fail executed
PMU_WAKE	1	rwh	**Flag which indicates a reset caused by stop-exit** *Note: Stop-exit with reset must be configured explicitly in the PMU_SLEEP register.*[1)] 0_B **No_reset**: No reset caused by stop-exit executed 1_B **Reset**: Reset caused by stop-exit executed
PMU_SleepEX	2	rwh	**Flag which indicates a reset caused by sleep-exit** 0_B **No_reset**: No reset caused by sleep-exit executed 1_B **Reset**: Reset caused by sleep-exit executed
PMU_LPR	3	rwh	**Low priority resets** *Note: Low priority resets are PMU_SOFT & LOCKUP.* 0_B **Low_priority_reset**: Low priority-reset executed 1_B **Low_priority**: Low priority executed
PMU_ClkWDT	4	rwh	**Clock watchdog (CLKWDT) reset flag** 0_B **No_reset**: Noclock watchdog reset executed 1_B **Reset**: Clock watchdog reset executed
PMU_ExtWDT	5	rwh	**External watchdog (WDT1) reset flag**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **No_reset**: No external watchdog reset executed 1_B **Reset**: External watchdog reset executed
PMU_PIN	6	rwh	**PIN-reset flag** 0_B **No_reset**: No PIN-reset executed 1_B **Reset**: PIN-reset executed
PMU_VS_POR	7	rwh	**Power-on reset flag** 0_B **No_reset**: No power-on reset executed 1_B **Reset**: Power-on reset executed
RES	8, 31:11	r	**Reserved** Always read as 0
PMU_SOFT	9	rwh	**Soft-reset flag** 0_B **No_reset**: No soft-reset executed 1_B **Reset**: Soft-reset executed
LOCKUP	10	rwh	**Lockup-reset flag** 0_B **No_reset**: No lockup-reset executed 1_B **Reset**: Lockup-reset executed

1) Otherwise this flag is not set. The flag is always set in case of pin reset in Stop mode (in combination with the flag PMU_PIN).

5.9.14 Reset blind time register

PMU_CNF_RST_TFB Offset address: $006C_H$

Reset blind time register RESET_TYPE_1 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														**RST_TFB**	
r														rw	

Field	Bits	Type	Description
RST_TFB	1:0	rw	**Reset pin blind time selection bits** These bits select the blind time for the 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.
RES	31:2	r	**Reserved** Always read as 0

5.9.15 General purpose user DATA0to3 register

PMU_GPUDATA0to3 Offset address: $00C0_H$

General purpose user DATA0to3 register RESET_TYPE_1 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3								**DATA2**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1								**DATA0**							
rw								rw							

Field	Bits	Type	Description
DATA0	7:0	rw	**DATA0 storage byte** 1st byte of storage area
DATA1	15:8	rw	**DATA1 storage byte** 2nd byte of storage area
DATA2	23:16	rw	**DATA2 storage byte** 3rd byte of storage area
DATA3	31:24	rw	**DATA3 storage byte** 4th byte of storage area

5.9.16 General purpose user DATA4to7 register

PMU_GPUDATA4to7 Offset address: $00C4_H$

General purpose user DATA4to7 register RESET_TYPE_1 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7								**DATA6**							
rw								rw							

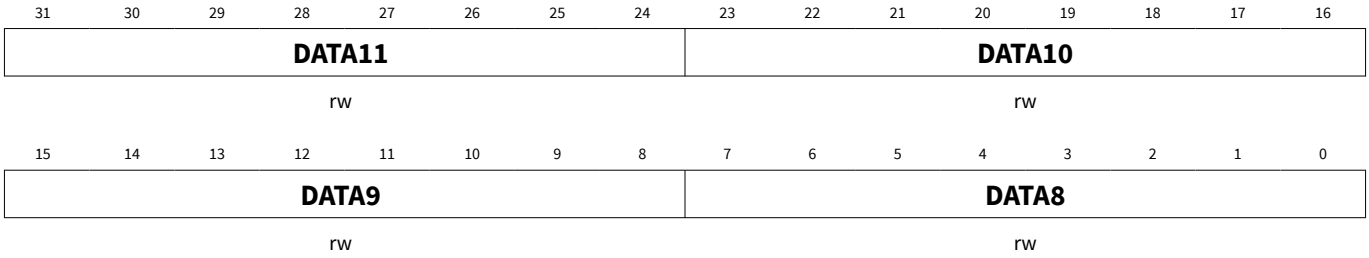
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5								**DATA4**							
rw								rw							

Field	Bits	Type	Description
DATA4	7:0	rw	**DATA4 storage byte** 5th byte of storage area
DATA5	15:8	rw	**DATA5 storage byte** 6th byte of storage area
DATA6	23:16	rw	**DATA6 storage byte** 7th byte of storage area
DATA7	31:24	rw	**DATA7 storage byte** 8th byte of storage area

5.9.17 General purpose user DATA8to11 register

PMU_GPUDATA8to11 Offset address: $00C8_H$

General purpose user DATA8to11 register RESET_TYPE_1 value: $0000\ 0000_H$



Field	Bits	Type	Description
DATA8	7:0	rw	**DATA8 storage byte** 9th byte of storage area
DATA9	15:8	rw	**DATA9 storage byte** 10th byte of storage area
DATA10	23:16	rw	**DATA10 storage byte** 11th byte of storage area
DATA11	31:24	rw	**DATA11 storage byte** 12th byte of storage area

5.9.18 Settings monitor 1-4 register

PMU_MON_CNF1 Offset address: 0034_H

Settings monitor 1-4 register RESET_TYPE_2 value: $4747\ 4747_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MON 4_STS	**RES1**	**MON 4_PU**	**MON 4_PD**	**MON 4_CYC**	**MON 4_RISE**	**MON 4_FALL**	**MON 4_EN**	**MON 3_STS**	**RES1**	**MON 3_PU**	**MON 3_PD**	**MON 3_CYC**	**MON 3_RISE**	**MON 3_FALL**	**MON 3_EN**
r	r	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON 2_STS	**RES1**	**MON 2_PU**	**MON 2_PD**	**MON 2_CYC**	**MON 2_RISE**	**MON 2_FALL**	**MON 2_EN**	**MON 1_STS**	**RES1**	**MON 1_PU**	**MON 1_PD**	**MON 1_CYC**	**MON 1_RISE**	**MON 1_FALL**	**MON 1_EN**
r	r	rw	rw	rw	rw	rw	rw	r	r	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
MON1_EN	0	rw	**MON1 enable** 0_B **DISABLE**: MON1 disabled 1_B **ENABLE**: MON1 enabled
MON1_FALL	1	rw	**MON1 wake-up on falling edge enable** *Note: Works only if MON1_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON1_RISE	2	rw	**MON1 wake-up on rising edge enable** *Note: Works only if MON1_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON1_CYC	3	rw	**MON1 for cycle sense enable** *Note: Works only if MON1_EN is enabled and if MON1_FALL and/or MON1_RISE is/are enabled* 0_B **DISABLE**: Cycle sense disabled 1_B **ENABLE**: Cycle sense enabled
MON1_PD	4	rw	**Pull-down current source for MON1 input enable** *Note: Works only if MON1_EN is enabled* 0_B **DISABLE**: Pull-down source disabled 1_B **ENABLE**: Pull-down source enabled
MON1_PU	5	rw	**Pull-up current source for MON1 input enable** *Note: Works only if MON1_EN is enabled* 0_B **DISABLE**: Pull-up source disabled

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLE**: Pull-up source enabled
RES1	6, 14, 22, 30	r	**Reserved** Always read as 1
MON1_STS	7	r	**MON1 status input** *Note: MONx_STS is not updated, when MONx_EN is switched off. MONx_STS is also not updated, when both wake-options (MONx_RISE and MONx_FALL) are off.* 0_B **Low_status**: MON input has low status 1_B **High_status**: MON input has high status
MON2_EN	8	rw	**MON2 enable** 0_B **DISABLE**: MON2 disabled 1_B **Enabled**: MON2 enabled
MON2_FALL	9	rw	**MON2 wake-up on falling edge enable** *Note: Works only if MON2_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON2_RISE	10	rw	**MON2 wake-up on rising edge enable** *Note: Works only if MON2_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON2_CYC	11	rw	**MON2 for cycle sense enable** *Note: Works only if MON2_EN is enabled and if MON2_FALL and/or MON2_RISE is/are enabled* 0_B **DISABLE**: Cycle sense disabled 1_B **ENABLE**: Cycle sense enabled
MON2_PD	12	rw	**Pull-down current source for MON2 Input enable** *Note: Works only if MON2_EN is enabled* 0_B **DISABLE**: Pull-down source disabled 1_B **ENABLE**: Pull-down source enabled
MON2_PU	13	rw	**Pull-up current source for MON2 input enable** *Note: Works only if MON2_EN is enabled* 0_B **DISABLE**: Pull-up source disabled 1_B **ENABLE**: Pull-up source enabled
MON2_STS	15	r	**MON2 status input**

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: MONx_STS is not updated, when MONx_EN is switched off. MONx_STS is also not updated, when both wake-options (MONx_RISE and MONx_FALL) are off. 0_B **Low_status**: MON input has low status 1_B **High_status**: MON input has high status
MON3_EN	16	rw	**MON3 enable** 0_B **DISABLE**: MON3 disabled 1_B **ENABLE**: MON3 enabled
MON3_FALL	17	rw	**MON3 wake-up on falling edge enable** *Note: Works only if MON3_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON3_RISE	18	rw	**MON3 wake-up on rising edge enable** *Note: Works only if MON3_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON3_CYC	19	rw	**MON3 for cycle sense enable** *Note: Works only if MON3_EN is enabled and if MON3_FALL and/or MON3_RISE is/are enabled* 0_B **DISABLE**: Cycle sense disabled 1_B **ENABLE**: Cycle sense enabled
MON3_PD	20	rw	**Pull-down current source for MON3 input enable** *Note: Works only if MON3_EN is enabled* 0_B **DISABLE**: Pull-down source disabled 1_B **ENABLE**: Pull-down source enabled
MON3_PU	21	rw	**Pull-up current source for MON3 Input enable** *Note: Works only if MON3_EN is enabled* 0_B **DISABLE**: Pull-up source disabled 1_B **ENABLE**: Pull-up source enabled
MON3_STS	23	r	**MON3 Status Input** *Note: MONx_STS is not updated, when MONx_EN is switched off. MONx_STS is also not updated, when both wake-options (MONx_RISE and MONx_FALL) are off.* 0_B **Low_status**: MON input has low status 1_B **High_status**: MON input has high status
MON4_EN	24	rw	**MON4 Enable** 0_B **DISABLE**: MON4 disabled

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLE**: MON4 enabled
MON4_FALL	25	rw	**MON4 wake-up on falling edge enable** *Note: Works only if MON4_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON4_RISE	26	rw	**MON4 wake-up on rising edge enable** *Note: Works only if MON4_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **Enabled**: Wake-up enabled
MON4_CYC	27	rw	**MON4 for cycle sense enable** *Note: Works only if MON4_EN is enabled and if MON4_FALL and/or MON4_RISE is/are enabled* 0_B **DISABLE**: Cycle sense disabled 1_B **ENABLE**: Cycle sense enabled
MON4_PD	28	rw	**Pull-down current source for MON4 input enable** *Note: Works only if MON4_EN is enabled* 0_B **DISABLE**: Pull-down source disabled 1_B **ENABLE**: Pull-down source enabled
MON4_PU	29	rw	**Pull-up current source for MON4 input enable** *Note: Works only if MON4_EN is enabled* 0_B **DISABLE**: Pull-up source disabled 1_B **ENABLE**: Pull-up source enabled
MON4_STS	31	r	**MON4 status input** *Note: MONx_STS is not updated, when MONx_EN is switched off. MONx_STS is also not updated, when both wake-options (MONx_RISE and MONx_FALL) are off.* 0_B **Low_status**: MON input has low status 1_B **High_status**: MON input has high status

5.9.19 Settings monitor 5 register

PMU_MON_CNF2 Offset address: 0038_H

Settings monitor 5 register RESET_TYPE_2 value: $0000\ 0047_H$

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								MON5_STS	RES1	MON5_PU	MON5_PD	MON5_CYC	MON5_RISE	MON5_FALL	MON5_EN
r								r	r	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
MON5_EN	0	rw	**MON5 enable** 0_B **DISABLE**: MON5 disabled 1_B **ENABLE**: MON5 enabled
MON5_FALL	1	rw	**MON5 wake-up on falling edge enable** *Note: Works only if MON5_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON5_RISE	2	rw	**MON5 wake-up on rising edge enable** *Note: Works only if MON5_EN is enabled* 0_B **DISABLE**: Wake-up disabled 1_B **ENABLE**: Wake-up enabled
MON5_CYC	3	rw	**MON5 for cycle sense enable** *Note: Works only if MON5_EN is enabled and if MON5_FALL and/or MON5_RISE is/are enabled* 0_B **DISABLE**: Cycle sense disabled 1_B **ENABLE**: Cycle sense enabled
MON5_PD	4	rw	**Pull-down current source for MON5 input enable** *Note: Works only if MON5_EN is enabled* 0_B **DISABLE**: Pull-down source disabled 1_B **ENABLE**: Pull-down source enabled
MON5_PU	5	rw	**Pull-up current source for MON5 input enable** *Note: Works only if MON5_EN is enabled* 0_B **DISABLE**: Pull-up source disabled 1_B **ENABLE**: Pull-up source enabled

(table continues...)

(continued)

Field	Bits	Type	Description
RES1	6	r	**Reserved** Always read as 1
MON5_STS	7	r	**MON5 status input** *Note: MONx_STS is not updated, when MONx_EN is switched off. MONx_STS is also not updated, when both wake-options (MONx_RISE and MONx_FALL) are off.* 0_B **Low_status**: MON input has low status 1_B **High_status**: MON input has high status
RES	31:8	r	**Reserved** Always read as 0

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, power down, sleep)
- Interrupt enabling 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 MOTIX™ TLE984xQX. The SCU is made up of the following submodules:

- Clock system and control (CGU) (see Chapter 6.3)
- Reset control (RCU) (see Chapter 6.4)
- Power management (PCU) (see Chapter 6.5)
- Interrupt management (ICU) (see Chapter 6.6)
- General port control (see Chapter 6.7)
- Flexible peripheral management (see Chapter 6.9)
- Module suspend control (see Chapter 6.10)
- Error detection and correction in data memory (see Chapter 6.13)
- Miscellaneous control (see Chapter 6.14)
- Register mapping (see SCU register overview)

6.2.1 Block diagram

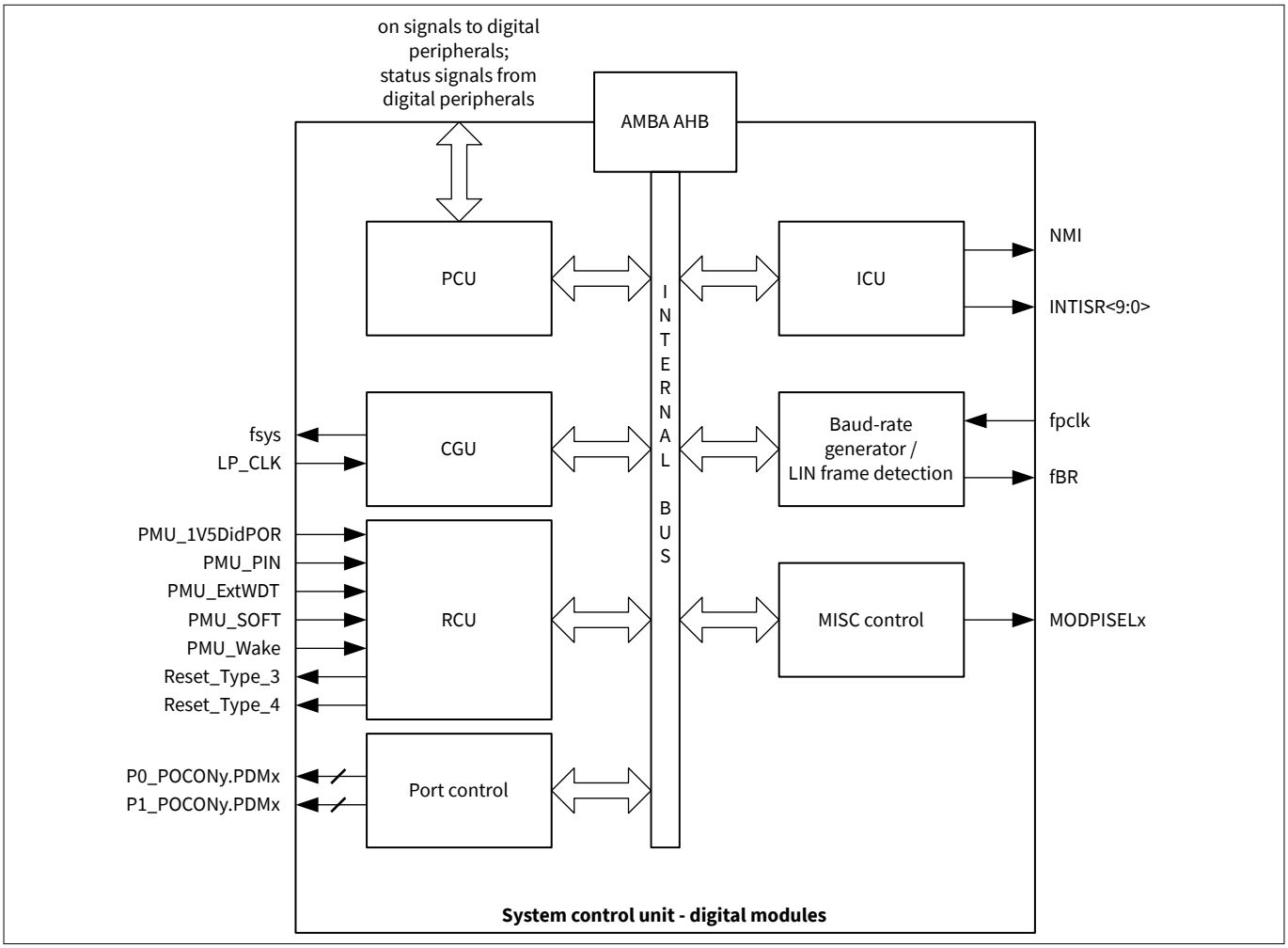


Figure 25 System control unit - digital modules block diagram

IO description of SCU_DM

- CGU:
  - f_{sys}: system clock
  - LP_CLK: low-power backup clock
- RCU:
  - 1V5DidPOR: undervoltage reset of power down supply
  - PMU_PIN: reset generated by reset pin
  - PMU_ExtWDT: WDT1 reset
  - PMU_SOFT: software reset
  - PMU_Wake: stop mode exit with reset
  - Reset_Type_3: peripheral reset (contains all resets)
  - Reset_Type_4: peripheral reset (without SOFT)
- Baud-rate generator:
  - f_{BR}: baud-rate clock for UART

- Port control:
  - P0_POCONy.PDMx: driver strength control
  - P1_POCONy.PDMx: driver strength control
- MISC:
  - MODPISELx: mode selection registers for UART (source selection) and Timer (trigger or count selection)

6.3 Clock generation unit

The clock generation unit (CGU) provides a flexible clock generation for MOTIX™ TLE984xQX. During user program execution the frequency can be programmed for an optimal ratio between performance and power consumption. Therefore the power consumption can be adapted to the actual application state.

The CGU in the MOTIX™ TLE984xQX consists of one oscillator circuit (OSC_HP), a phase-locked loop (PLL) module including 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 out of the following selectable clocks:

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

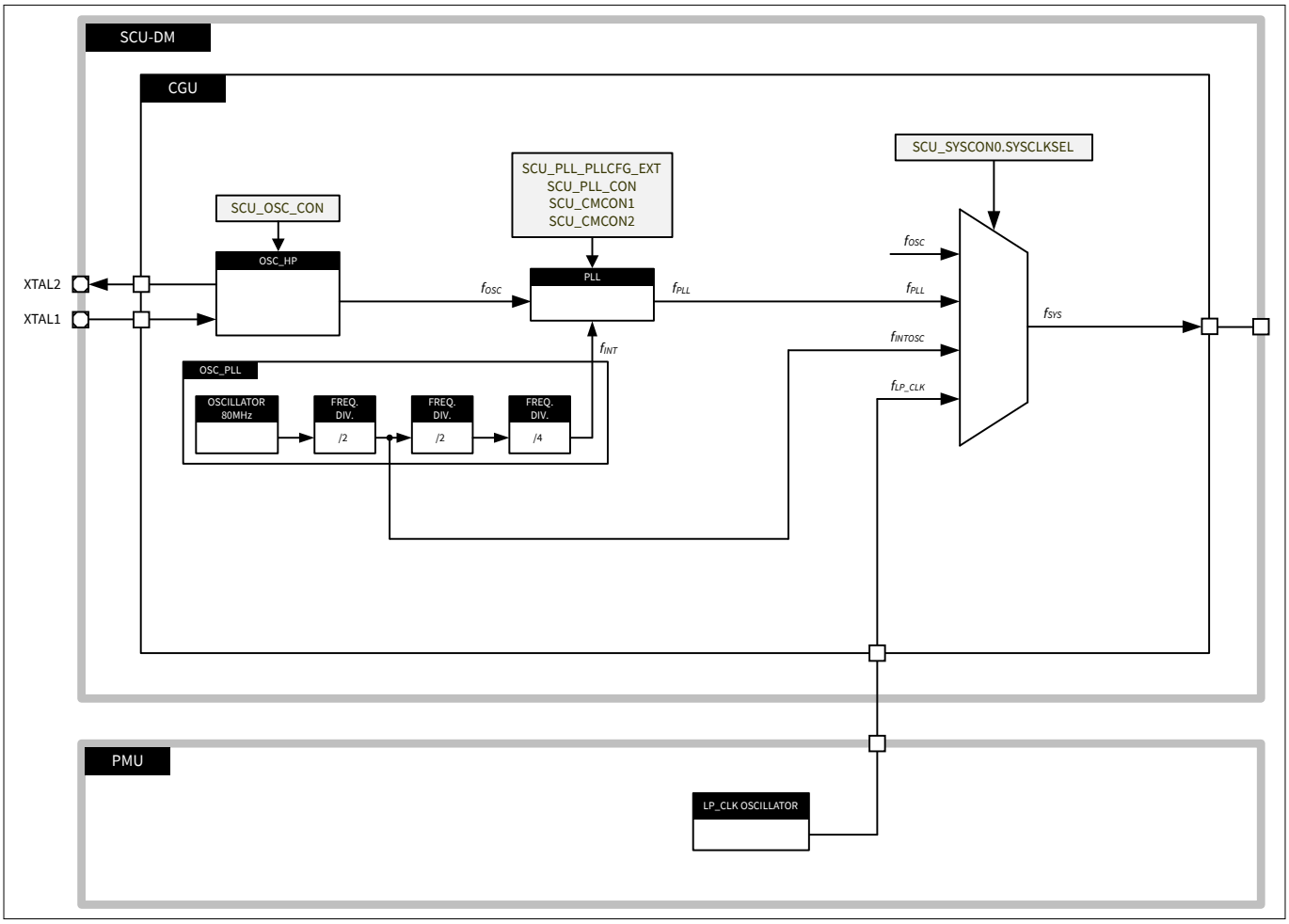


Figure 26 Clock generation unit block diagram

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, see f_{LP_CLK}) that is enabled by hardware as an independent clock source for the MOTIX™ TLE984xQX startup after reset and during the power-down wake-up sequence. There is no user configuration possible on f_{LP_CLK}.

6.3.2 High precision oscillator circuit (OSC_HP)

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

Figure 27 shows the recommended external circuitries for both operating modes, external crystal mode and external input clock mode.

6.3.2.1 External input clock mode

When supplying the clock signal directly, not using an external crystal and bypassing the oscillator, the input frequency needs to be within the range of 4 MHz to 6 MHz if the PLL VCO part is used.

When using an external clock signal 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 6 MHz. Set the bit field SCU_MODPISEL1.XTAL12EN to active the external resonator. An external oscillator load circuitry must be used, connected to both pins, XTAL1 and XTAL2. It consists normally of the two load capacitances C1 and C2, for some crystals a series damping resistor might be necessary. The exact values and related operating range are dependent on the crystal and have to be determined and optimized together with the crystal vendor using the negative resistance method. As starting point for the evaluation, the following load cap values may be used:

Table 23 External CAP capacitors

Fundamental mode crystal frequency (approx., MHz)	Load caps C_1, C_2 (pF)
4	33
5	22
6	18

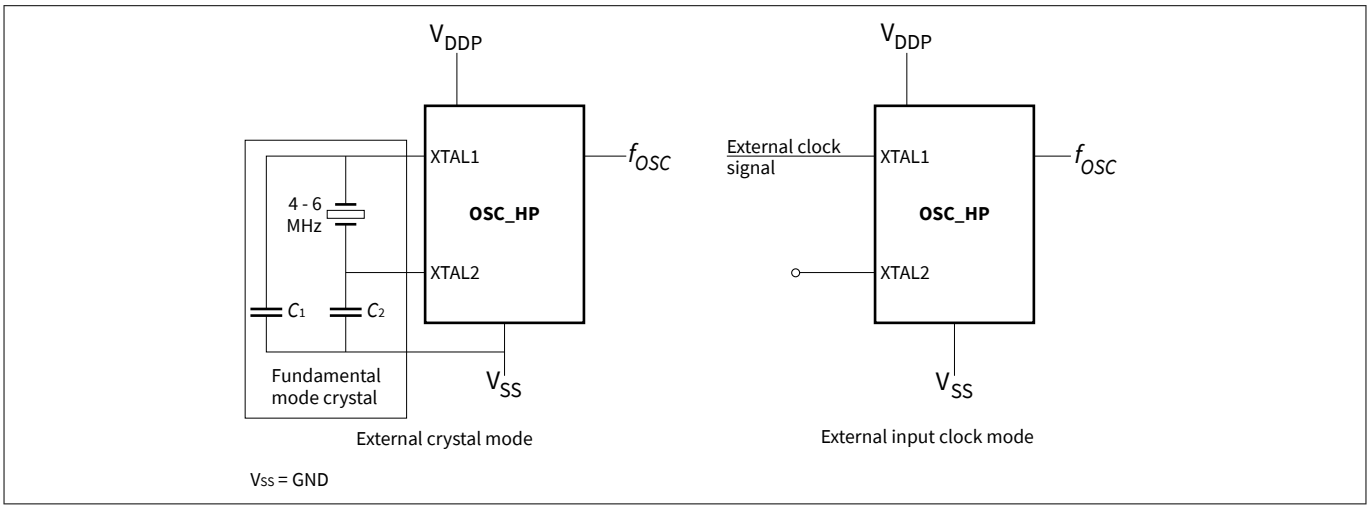


Figure 27 MOTIX™ TLE984xQX external circuitry for the OSC_HP

6.3.3 Phase-locked loop (PLL) module

This section describes the MOTIX™ TLE984xQX PLL module.

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

- Prescaler mode, also called VCO bypass mode
- Normal mode
- Free running mode

6.3.3.1 Features

Following is an overview of the PLL features and functions:

- Programmable clock generation PLL
- Loop filter
- Wide range of input frequencies (divided by configurable P-divider)
- Wide VCO frequency tuning range
- VCO lock detection
- Oscillator run detection
- VCO output frequency feedback N-divider
- VCO output frequency K1-divider and K2-divider
- Oscillator watchdog
- Prescaler mode
- Free running mode
- Normal mode
- Sleep mode automatically activated during device power-save 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

6.3.3.2 PLL functional description

The following figure shows the PLL block structure.

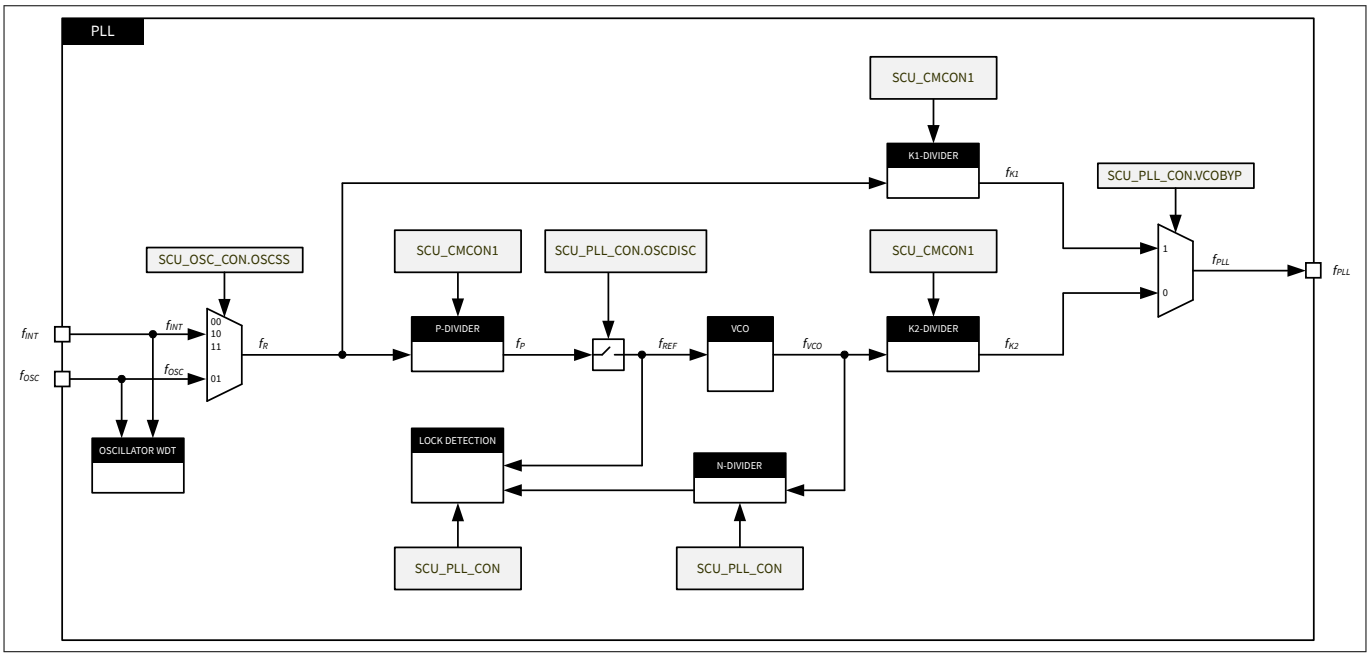


Figure 28 PLL block diagram

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}.

The PLL uses up to three dividers to set the system frequency f_{sys} in a flexible way. Each of the three dividers can be bypassed corresponding to the PLL operating mode (based on f_{PLL}):

- Bypassing P-, N- and K2-dividers; this defines the prescaler mode
- Bypassing K1-divider; this defines the normal mode
- Bypassing K1-divider and ignoring the P-divider; this defines the free running mode

The following table shows the selectable clock source options:

Table 24 Clock option selection

VCOBYP	OSCDISC	Mode selected
0	0	Normal mode
1	x	Prescaler mode
0	1	Free running mode

Normal mode

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

The output frequency is given by:

$$f_{\mathrm{PLL}} = \frac{N}{P \times K2} \times f_R \tag{3}$$

The normal mode is selected by the following settings:

- SCU_PLL_CON.VCOBYP = 0

The normal mode is active when:

- SCU_PLL_CON.VCOBYP = 0

- SCU_PLL_CON.OSCDISC = 0
- SCU_PLL_CON.LOCK = 1

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

Note: ***When configuring the PLL frequency f_{PLL} by the P- and N-dividers the user shall take care that the limits for f_{REF}, f_{VCO} and f_{sys} are not exceeded.***

Prescaler mode (VCO bypass mode)

In prescaler mode the reference frequency f_R is only divided down by a factor K1.
The output frequency is given by:

$$f_{\mathrm{PLL}} = \frac{f_R}{K1} \tag{4}$$

The prescaler mode is selected by the following settings:
- SCU_PLL_CON.VCOBYP = 1
- SCU_PLL_CON.OSCDISC = X

The prescaler mode is active when:
- SCU_PLL_CON.VCOBYP = 1
- SCU_PLL_CON.OSCDISC = X
- SCU_OSC_CON.OSC2L = 0 if f_{OSC} is provided as $f_{R\,(SCU\_OSC\_CON.OSCSS = 01B)}$

Free running mode

In free running mode the base frequency output of the voltage controlled oscillator (VCO) $f_{VCObase}$ is only divided down by a factor K2.
The output frequency is given by:

$$f_{\mathrm{PLL}} = \frac{f_{\mathrm{VCObase}}}{K2} \tag{5}$$

The free running mode is enabled by the following settings/conditions:
- SCU_PLL_CON.VCOBYP = 0 and SCU_PLL_CON.LOCK = 0

or
- SCU_PLL_CON.VCOBYP = 1 and SCU_OSC_CON.OSCSS = 1 and SCU_OSC_CON.OSC2L = 1

or
- SCU_PLL_CON.VCOBYP = 0 and SCU_PLL_CON.OSCDISC = 1 and SCU_PLL_CON.LOCK = 0

General configuration overview

The divider values and all necessary other values can be configured via the PLL configuration registers.
In MOTIX™ TLE984xQX the P factor can be programmed to the values 4, 5 or 6. The following table shows all possible values for the P factor and gives the valid input frequency range f_R for the P-divider dependent configuration and the resulting f_P frequency values which are directly linked to f_{REF}:

Table 25 P-divider factor

P	f_P for f_R =		
	4 MHz	**5 MHz**	**6 MHz**
4	1	1.25	not allowed
5	0.8	1	1.2
6	not allowed	0.833	1

Note: Of course the whole range in between two f_R columns in the above table is allowed if parameter f_{VCO} is kept within the specified limits. The minimum and maximum limits of f_P result out of the parameter specification of f_{REF} and its variation as f_P is directly linked to f_{REF}.

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:

Table 26 VCO range

Minimum VCO tuning range frequency	Maximum VCO tuning range frequency	VCO free running frequency	Unit
see f_{VCO_min}	see f_{VCO_max}	see $f_{VCOfree}$ [1)]	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 (N-divider factors) and gives the valid output frequency range for f_{REF} depending on N and the VCO frequency range. All not allowed combinations are related to the fact that using them the limits of parameter f_{REF} are violated:

Table 27 N-divider factor

N	f_{DIV} for f_{VCO} =				
	75	**96**	**112**	**136**	**160**
1-47	not accessible				
48	not allowed	not allowed	not allowed	not allowed	not allowed
50	not allowed	not allowed	not allowed	not allowed	not allowed
51	not allowed	not allowed	not allowed	not allowed	not allowed
52	not allowed	not allowed	not allowed	not allowed	not allowed
54	not allowed	not allowed	not allowed	not allowed	not allowed
60	1.25	not allowed	not allowed	not allowed	not allowed
67	1.12	not allowed	not allowed	not allowed	not allowed
72	1.04	not allowed	not allowed	not allowed	not allowed
75	1.00	not allowed	not allowed	not allowed	not allowed
78	0.96	1.23	not allowed	not allowed	not allowed
80	0.94	1.2	not allowed	not allowed	not allowed
88	0.85	$1.0\overline{9}$	not allowed	not allowed	not allowed
90	0.83	1.07	1.24	not allowed	not allowed
94	0.80	1.02	1.19	not allowed	not allowed

(table continues...)

Table 27 (continued) N-divider factor

N	f_{DIV} for f_{VCO} =				
	75	**96**	**112**	**136**	**160**
100	not allowed	0.96	1.12	not allowed	not allowed
160	not allowed	not allowed	not allowed	0.85	1.00
others	not accessible				

Note: The not allowed settings are related to the fact that the maximum system frequency f_{sys_max} is exceeded. The whole range in between two f_{VCO} columns in the above table is allowed if the specification for the parameter f_{REF} is maintained as f_{DIV} is compared to f_{REF}.

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

Note: Due to this operation, the VCO clock of the PLL has a frequency which is a multiple of f_{DIV}. The factor for this is controlled through the value applied to the N-divider in the feedback path. For this reason this factor is often called a multiplier, although it actually controls division.

The output frequency of the VCO, f_{VCO}, is divided by K2 to provide the final desired output frequency f_{PLL}. The following table shows the output frequency range depending on the K2-divider and the VCO frequency range:

Table 28 K2-divider table

K2	f_{PLL} for f_{VCO} =					Typical duty cycle [%]
	75	**96**	**112**	**136**	**160**	
2	37.5	not allowed	not allowed	not allowed	not allowed	50
3	25.0	32.0	37.3	not allowed	not allowed	40
4	18.8	24.0	28.0	34.0	40.0	50
5	15.0	19.2	22.4	27.2	32.0	44
others	not accessible					

Note: The whole range in between two f_{vco} columns in the above table is only allowed if the maximum specified system frequency f_{sys} is not exceeded.

For the K1-divider the same table is valid as for the K2-divider. The only difference is that not f_{VCO} is used as reference, f_R is used instead.

Table 29 K1-divider table

K1	f_{PLL} for f_R =			Duty cycle [%]
	4	**5**	**6**	
1	4.0	5.0	6.0	40 - 60
2	2.0	2.5	3.0	50
others	not accessible			

For different source oscillator, some examples for f_{PLL} are shown in the table below.

Table 30 System frequency

f_{PLL} selected	Oscillator	f_{OSC}	N	P	f_{REF}	K	Actual f_{sys}	Actual f_{VCO}
40 MHz	On-chip	5 MHz	80	5	1	2	40 MHz	80 MHz
	External	4 MHz	80	4	1	2	40 MHz	80 MHz
		5 MHz	80	5	1	2	40 MHz	80 MHz
		6 MHz	80	6	1	2	40 MHz	80 MHz
37.5 MHz	On-chip	5 MHz	90	4	1,25	3	37.5 MHz	112.5 MHz
	External	5 MHz	90	4	1,25	3	37.5 MHz	112.5 MHz
25 MHz	On-chip	5 MHz	100	5	1	4	25 MHz	100 MHz
	External	4 MHz	100	4	1	4	25 MHz	100 MHz
		5 MHz	100	5	1	4	25 MHz	100 MHz
		6 MHz	100	6	1	4	25 MHz	100 MHz
20 MHz	On-chip	5 MHz	80	5	1	4	20 MHz	80 MHz
	External	4 MHz	80	4	1	4	20 MHz	80 MHz
		5 MHz	80	5	1	4	20 MHz	80 MHz
		6 MHz	80	6	1	4	20 MHz	80 MHz
16 MHz	On-chip	5 MHz	80	5	1	5	16 MHz	80 MHz
	External	4 MHz	80	4	1	5	16 MHz	80 MHz
		5 MHz	80	5	1	5	16 MHz	80 MHz
		6 MHz	80	6	1	5	16 MHz	80 MHz

Note: For the MOTIX™ TLE984xQX, the value of P is configurable. In order to obtain the required f_{PLL}, the values of N and K can be chosen respectively by the bits NDIV and K2DIV for different oscillator input frequencies. When configuring the required f_{PLL} it has to be ensured that the limits of parameter f_{sys}, f_{REF} and f_{VCO} are kept.

6.3.3.3 Oscillator watchdog

The oscillator watchdog monitors the external incoming clock f_{OSC}. Only incoming frequencies that are too low to enable a stable operation of the VCO circuit are detected.

As reference clock the internal oscillator (OSC_PLL) frequency f_{INT} is used and therefore the internal oscillator must be put into operation.

By setting bit SCU_OSC_CON.OSCWDTRST the oscillator watchdog can be restarted without a reset of the complete PLL. The detection status output is only valid after some 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 level are not marked by a loss of lock because the VCO can handle such small changes without any problem for the system. The following table shows values below that the lock is not lost for different input values.

Table 31 Loss of VCO lock definition

Maximum allowed changing $\Delta f_{DIV}/\Delta t$ for f_{REF} =		
0.8 MHz	1 MHz	1.25 MHz
≤ 0.54 kHz/µs	≤ 0.96 kHz/µs	≤ 1.49 kHz/µs

6.3.3.5 Internal oscillator (OSC_PLL)

The PLL internal oscillator (OSC_PLL) is used for two different purposes:

Providing an input clock to the PLL

The PLL is supplied by a reference clock (f_{INT}) set to a nominal frequency of 5 MHz.
The OSC_PLL can be used as input clock for all PLL modes. This is controlled and configured via SCU_OSC_CON.OSCSS.

Operating the oscillator watchdog

The input frequency for the PLL direct from OSC_HP (XTAL), is supervised using the OSC_PLL as reference frequency. For more information see Chapter 6.3.3.3.

6.3.3.6 Switching PLL parameters

The following restriction applies when changing PLL parameters via the SCU_PLL_CON register:

- Disable PLL loss-of-lock NMI
- Set SCU_CMCON1.K1DIV = 1 (div 1) to set f_{PLL} to 5 MHz when Prescaler mode is enabled
- Switch to Prescaler mode by writing SCU_PLL_CON.VCOBYP = 1 [4)]
- Set SCU_CMCON1.PDIV to the desired value
- Set the SCU_PLL_CON.NDIV to the desired value
- Set SCU_CMCON1.K2DIV to the desired value
- Select the desired clock source by writing SCU_OSC_CON.OSCSS to the desired value
- Restart PLL lock detection by writing SCU_PLL_CON.RESLD = 1
- Wait until PLL locks (the PLL is locked when SCU_PLL_CON.LOCK = 1)
- Switch to Normal mode by writing SCU_PLL_CON.VCOBYP = 0
- Enable PLL loss-of-lock NMI if desired

6.3.3.7 Oscillator watchdog event or PLL loss of lock detection

In case of detection of too low frequency of the external clock source f_{OSC}, the OSC-too-low flag (SCU_OSC_CON.OSC2L) is set. If enabled by NMICON.NMIOWD, a trap request to the CPU is activated correspondingly only in these two cases:

1. When PLL is in prescaler mode and OSCSS = 01 selecting f_{OSC} as PLL input clock source and SCU_SYSCON0.SYSCLKSEL selects PLL clock output as the system frequency, or
2. When SCU_SYSCON0.SYSCLKSEL selects f_{OSC} as the system frequency.

4 This operation sets SCU_PLL_CON.OSCDISC to '0' by hardware.

With these 2 cases and the OSC2L condition, the OWD NMI flag FNMIOWD in NMISR is set.

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

An oscillator watchdog event normally leads to a following PLL loss-of-lock detection.

If PLL is not the system clock source (SCU_SYSCON0.SYSCLKSEL deselects PLL or PLL is in prescaler mode) when the loss-of-lock is detected, only the lock flag is reset (SCU_PLL_CON.LOCK = 0). No loss-of-lock NMI is generated and no further action is taken. Otherwise if PLL is selected as 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 that in the first place, the LOCK flag has to be set first before a loss-of-lock NMI request is generated. This avoids a potential PLL loss-of-lock NMI request after device power-on reset.

On an oscillator watchdog event when PLL is in prescaler mode and external clock (OSC_HP) is selected as PLL clock input; or on PLL loss-of-lock detection when PLL is in normal mode, the PLL will be switched to run in the free running mode on the VCO base frequency divided by K2, which is enforced by hardware until the prescaler mode is (re-)selected.

Due to the above, the PLL shall only run in prescaler mode when changing the PLL configuration or switching 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 reconfigured again by executing the following sequence as the OWD NMI routine:

1. Restart the oscillator watchdog detection by setting bit SCU_OSC_CON.OSCWDTRST.
2. Wait until SCU_OSC_CON.OSC2L is clear.
3. When bit SCU_OSC_CON.OSC2L is cleared, then
   - the prescaler mode has to be selected (SCU_PLL_CON.VCOBYP = 1)
   - setting the restart lock detection bit SCU_PLL_CON.RESLD = 1
   - waiting until the PLL VCO part becomes locked (SCU_PLL_CON.LOCK = 1)
   - when the LOCK is set again, the prescaler mode can be deselected (SCU_PLL_CON.VCOBYP = 0) and normal PLL operation is resumed.
4. Clear the OWD NMI flag FNMIOWD.

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

1. If input clock source is from XTAL (f_{OSC} from OSC_HP), ensure the input frequency is above threshold by checking SCU_OSC_CON.OSC2L.
2. The prescaler mode has to be selected (SCU_PLL_CON.VCOBYP = 1).
3. If desired, (re-)configure the PLL divider settings.
4. Setting the restart lock detection bit SCU_PLL_CON.RESLD = 1.
5. Waiting until the PLL VCO part becomes locked (SCU_PLL_CON.LOCK = 1).
6. When the LOCK is set again, the prescaler mode can be deselected (SCU_PLL_CON.VCOBYP = 0) and normal PLL operation is resumed.
7. Clear the PLL loss-of-lock NMI flag FNMIPLL.

6.3.4 Clock control unit

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

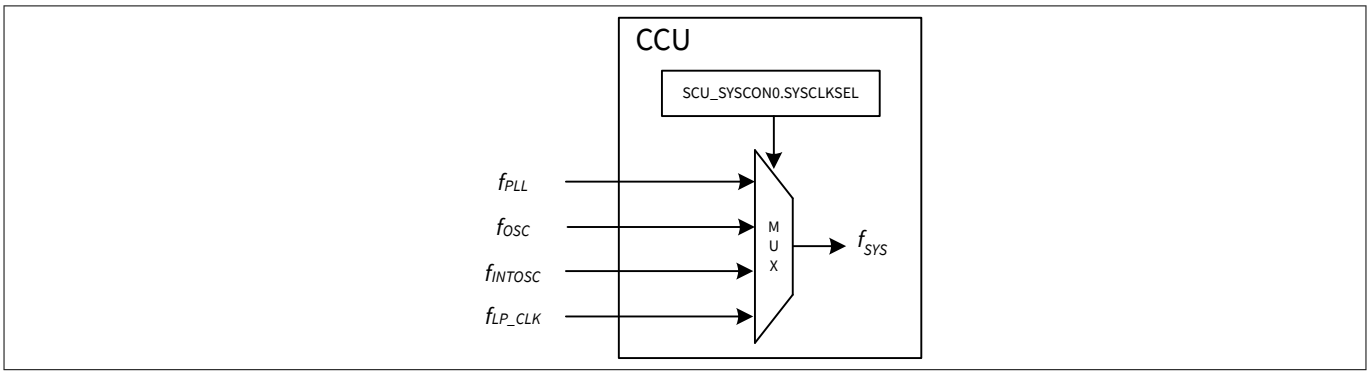


Figure 29 Clock inputs to clock control unit

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

- Clock slow down circuitry
- Centralized enable/disable circuit for clock control

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

- System frequency, f_{SYS} = up to 25 MHz or 40 MHz (product variant dependent) (measurement interface clock MI_CLK is derived from this clock)
- CPU clock (CCLK, SCLK) = up to 25 MHz or 40 MHz (product variant dependent) (divide-down of NVM access clock)
- NVM access clock (NVMACCCLK) = up to 25 MHz or 40 MHz (product variant dependent)
- Peripheral clock (PCLK, PCLK2, NVMCLK) = up to 25 MHz or 40 MHz (product variant dependent) (equals CPU clock; must be same or higher)
- TFILT_CLK: for digital filtering in analog peripherals, e.g. for comparators. Should be configured to be at 2 MHz (as close as possible).

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

For optimized NVM access (read/write) with reduced wait state(s) and with respect to system requirements on CPU operational frequency, bit field NVMCLKFAC is provided for setting the frequency factor between the NVM access clock NVMACCCLK and the CPU clock CCLK.

For the slow down mode, the operating frequency is reduced using the slow down circuitry with clock divider setting at the bit field CLKREL. Bit field CLKREL is only effective when slow down mode is enabled via SFR bit PMCON0.SD bit. Note that the slow down setting of bit field CLKREL correspondingly reduces the NVMACCCLK clock. Slow down setting does not influence the erase and write cycles for the NVM.

Peripherals UART1, UART2, T2 and T21 and are not influenced by CLKREL and either not by NVMCLKFAC, to allow functional LIN communication in slow down mode.

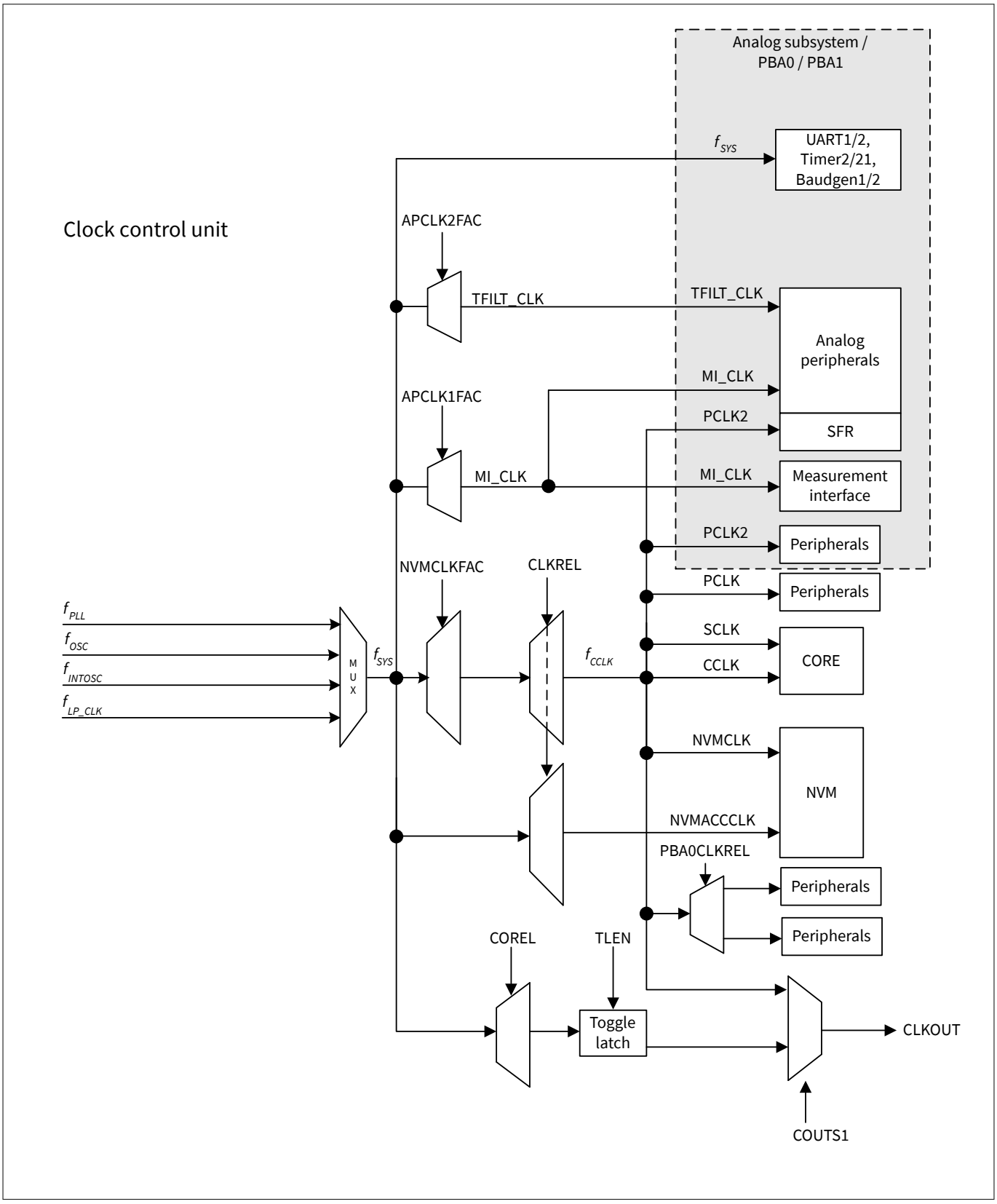


Figure 30 Clock generation from f_{sys}; CLKOUT generation

6.3.4.1 Clock tree

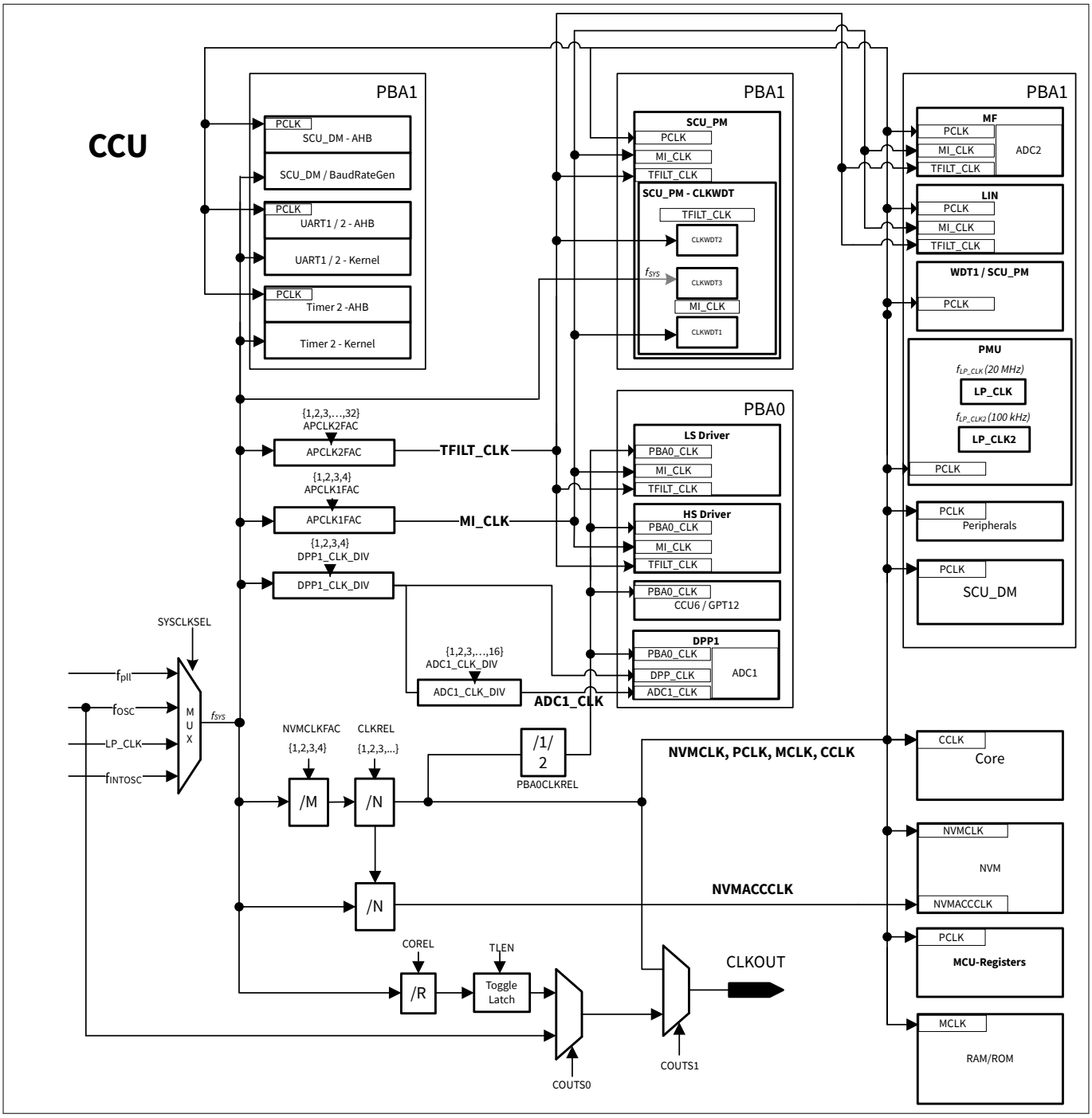


Figure 31 Clock tree

6.3.4.2 Startup control for system clock

Typically when the MOTIX™ TLE984xQX starts up after reset, the LP_CLK is selected by hardware to provide the system frequency f_{SYS}. CPU runs based on this system frequency during startup operation by boot firmware (unless otherwise specified and configured by firmware). Meanwhile, the system clock input is switched to the PLL output. With user boot configuration, the PLL is configured with internal oscillator (5 MHz) as input by default. User code can modify the default PLL configuration as required.

The exception to the above is with a reset that does not reset the clock system: soft reset. With this reset, the previous user configuration of PLL and clock system is retained across the reset.

Note: In the event the PLL fails to lock during startup operation, LP_CLK continues to provide the system clock input. The system clock input source is indicated by the register bit field SCU_SYSCON0.SYSCLKSEL.

6.3.5 External clock output

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

If COUTS1 = 0 (independent on COUTS0), the output clock is f_{CCLK}. Otherwise, if COUTS0 = 0, the output clock is from oscillator output frequency; if COUTS0 = 1, the clock output frequency is chosen by the bit field COREL which selects the n divider factor on f_{SYS}. Under this selection, 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 Clock generation unit (CGU) registers

The registers of the clock generation unit for PLL and oscillator control is not affected by the soft reset. Therefore the system clock configuration and frequency is maintained across these types of reset.

Unless otherwise stated, the reset value as stated for the following registers apply only with power-on reset, brown-out reset, hard reset, WDT1 reset or wake-up reset.

6.3.6.1 Register overview - Clock generation unit registers (ascending offset address)

Table 32 Register overview - Clock generation unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_PLL_CON	PLL control register	0044_H	153
SCU_CMCON1	Clock control 1 register	0048_H	155
SCU_CMCON2	Clock control 2 register	$004C_H$	157
SCU_APCLK_CTRL	Analog peripheral clock control register	0054_H	158
SCU_APCLK	Analog peripheral clock register	0058_H	159
SCU_APCLK_STS	Analog peripheral clock status register	$005C_H$	161
SCU_APCLK_SCLR	Analog peripheral clock status clear register	0064_H	163
SCU_ADC1_CLK	ADC1 peripheral clock register	$006C_H$	164
SCU_SYSCON0	System control 0 register	0070_H	165
SCU_OSC_CON	OSC control register	$00B0_H$	166
SCU_COCON	Clock output control register	$00B4_H$	168

6.3.6.2 PLL oscillator register

SCU_OSC_CON controls the setting and trimming of OSC_PLL, the power down of XTAL (OSC_HP) and the control and status monitor of oscillator watchdog.

6.3.6.3 PLL registers

SCU_PLL_CON, SCU_CMCON1 and SCU_CON2 control the PLL configuration or settings.

6.3.6.4 System clock control registers

The clock source for the system is selected via register SCU_SYSCON0.

6.3.6.5 Analog peripherals clock control registers

The clock frequency for the analog modules is selected via register SCU_APCLK. The APCLK is used as operating clock for all analog peripherals. For this reason it is important to choose always the required frequency range, if system clock is changed.

The clock source for the analog modules is selected via register SCU_APCLK1 and SCU_APCLK2.

Table 33 Possible clock configurations

Scenarios[1)]	f_{sys} [MHz]	pclk [MHz]	pba0_clk [MHz]	mi_clk [MHz]	tfilt_clk [MHz]
1: lowest possible system frequency	5	< 5	< 5	< 20	< 2
2: max. frequency scenario 1	25	< 25	< 25	< 20	< 2
3: max. frequency scenario 2	40	< 40	< 40	< 20	< 2

1) Besides of this scenarios which represent a kind of worst case all other scenarios shall not lead to an unrecoverable system state.

Table 34 Suggested value for APCLK

Clock frequency	APCLK1FAC	APCLK2FAC
24 MHz (Pll clk)	00_H	$0B_H$
40 MHz (Pll clk)	01_H	13_H

6.3.6.6 External clock control register

SCU_COCON controls the setting of external clock for CLKOUT.

6.4 Reset control

This section describes the types of reset and the effects of each reset on the MOTIX™ TLE984xQX.

6.4.1 Types of reset

The following reset types are recognized by the MOTIX™ TLE984xQX.

- Power-on reset
  - Requested asynchronously and released by supply voltage V_S reaching the upper threshold. Indication is a direct analysis of V_S undervoltage.
- Brown-out reset
  - Is not differentiated by system with power-on reset.
- Wake-up reset
  - Requested asynchronously by wake-up event during power save mode.
- Hardware reset
  - Requested asynchronously by event on external reset input (pin).
- WDT1 reset
  - Activated asynchronously by external WDT1 reset event.
- Soft reset
  - Requested synchronously by soft reset event.

6.4.2 Overview

When the MOTIX™ TLE984xQX is first powered up or with brown-out condition triggered by supply voltage input(s) going below the threshold, proper voltage thresholds must be reached before the MCU system starts operation with the release of the MCU, CPU and NVM resets. With all resets (except soft and SCU watchdog timer resets), the boot configuration is latched. The CPU starts to execute from the Boot ROM firmware with the release of MCU reset.

If the system is in power save mode, it is possible to wake-up with reset. Wake-up reset is basically equivalent to power-on reset except that it is a ‘warm’ reset and certain settings or configuration of the system are maintained across the reset. A wake-up via hard reset pin while in power save mode is effected as 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 on the system, if the timer is not refreshed before it overflows.

Soft reset can be triggered by application software where applicable.

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

6.4.3 Module reset behavior

The following table gives an overview on how the various modules or functions of the MOTIX™ TLE984xQX are affected with respect to the reset type. A "n" means that the module/function is reset to its default state. Refer to the following table for effective reset as priority.

Table 35 Effect of reset on modules/functions

Module/ function	Power-on/ brown-out reset	Wake-up reset[1)]	Hardware reset[1)]	WDT1 reset[1)]		Soft reset
CPU Core	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[1)]
Peripherals	n	n	n	n		n
Debug system	n	n	n	n		n
Port control	n	n	n	n		n
FW startup execution	Executes all INIT	Sleep: Executes all INIT	Executes most INIT	Executes most INIT		Skips not required INIT
On-chip static RAM	Initialized to 0	Sleep: Initialized to 0; Stop: Not affected[2)]	Not affected [2) 3)]	Not affected[2) 3)]		Not affected[2)]
Memory extension stack RAM	Affected	Affected	Affected	Affected		Affected
NVM	n	n	n	n		n incl. MapRAM
Clock system incl. PLL	n	n	n	n		Not affected[3)]

1) MCU sub-system: Hardware reset, WDT1 reset and wake-up reset (from stop mode or sleep mode) are generally HW-equivalent to power-on/brown-out reset, any exceptions are mainly due to power-on reset being a 'cold' start.
2) Not affected = Reset has no direct effect on RAM contents.
3) If the reset happens during a write to SRAM, the byte in the targeted write address may be corrupted.

6.4.4 Functional description of reset types

This section describes the definition and controls depending on the reset source.

6.4.4.1 Power-on/brown-out reset

Power-on reset is the highest level reset whereby the whole system is powered up and reset. 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

Wake-up reset occurs due to enabled event on defined functional input pins leading to reset of device while the device was in power-save mode. Wake-up reset from sleep and power-down (stop) mode is differentiated by respective indicator bits In case of wake-up from sleep mode, reset is always effected. Note that event on RESET input pin while device was in power-save mode is effectively a hardware reset (refer to PMU_RESET_STS register).

Wake-up reset has the next highest priority after power-on/brown-out reset.

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

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

In case of hardware reset is activated while the device is in power-save mode, this is effectively a wake-up reset.

In user mode, the system clock is switched.

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

6.4.4.4 WDT1 reset

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

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

6.4.4.5 Soft reset

Soft reset occurs due to software set of the soft reset request bit.

This has the lowest priority level. With this reset, the device continues running on the previous clock system configuration.

6.4.5 Reset control registers

6.4.5.1 Register overview - Reset control registers (ascending offset address)

Table 36 Register overview - Reset control registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_RSTCON	Reset control register	0068_H	170

6.4.6 Booting scheme

After any power-on reset, brown-out reset, hardware reset, WDT1 reset or wake-up reset, the pins TMS, P0.0, P0.2, together choose different modes. The following table shows the boot selection options available in the MOTIX™ TLE984xQX:

Table 37 MOTIX™ TLE984xQX boot options

TMS/SWD	P0.0	P0.2	MODE
0	x	x	User mode/BSL mode
1	1	0	Debug mode with serial wire (SW) port

6.5 Power management

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

6.5.1 Overview

The MOTIX™ TLE984xQX power management system allows 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 the following figure. Sleep mode is a special case which 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 provide flexible reduction of power consumption through a combination of techniques, including:

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

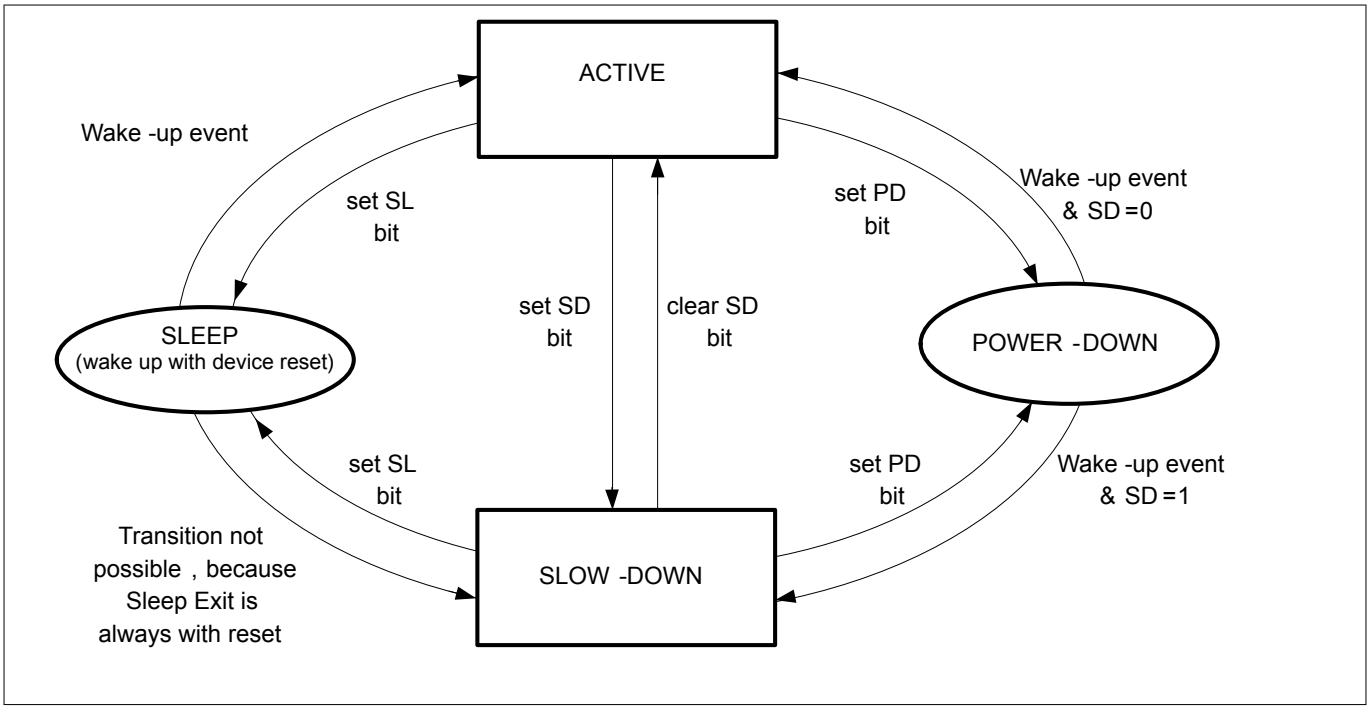


Figure 32 Transition between various modes of operation (without reset)

In slow down mode, the clock generation unit is instructed to reduce its clock frequency so that the clock to the system, that is core and peripheral, will be divided by a programmable factor.

In stop mode, the clock is turned off. Hence, it cannot be awakened by an interrupt or the watchdog timer. It will be awakened only when it receives an external wake-up signal or reset signal. The application must be prepared that the MOTIX™ TLE984xQX is served with one of these signals. A wake-up circuit is used to detect enabled wake-up signal(s) 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. On detection of wake-up event, a system reset is generated and the MCU is reset to default configuration then restart operation as initialized.

The priority for entry to the power-save modes starting from the highest is sleep mode, stop mode, then idle mode. Slow down mode can be enabled concurrently with idle mode.

6.5.2 Functional description

This section describes the power save modes, their operations, and entry and exit. It also describes the respective behavior of MOTIX™ TLE984xQX 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 bit SD in SFR PMCON0. The bit field SCU_CMCON1.CLKREL is used to select different slow down frequency. The CPU and peripherals are clocked at this lower frequency. The slow down mode is terminated by clearing bit SD.

6.5.2.2 Stop mode

In the stop mode, the NVM is put into NVM shut down mode (analog and digital except MapRAM shut down). The 5 V (VDDP) power supply to the analog modules ADC and PLL & internal oscillator is not removed. The MCU digital and NVM MapRAM is powered by the 1.5V (VDDC) regulator (reduced voltage). All functions of the microcontroller are stopped while the contents of the NVM, on-chip RAM, RAM, and the SFRs are maintained. As for the external ports, all digital pads 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 reset by asserting the hard reset pin.

Software requests stop mode by setting the bit PMCON0.PD to 1. In addition to this flag the WFI or WFE instruction has to be executed. When the controller will finish its currently executed interrupt task it will enter the stop mode. The following figure shows the required sequence to enter stop mode properly:

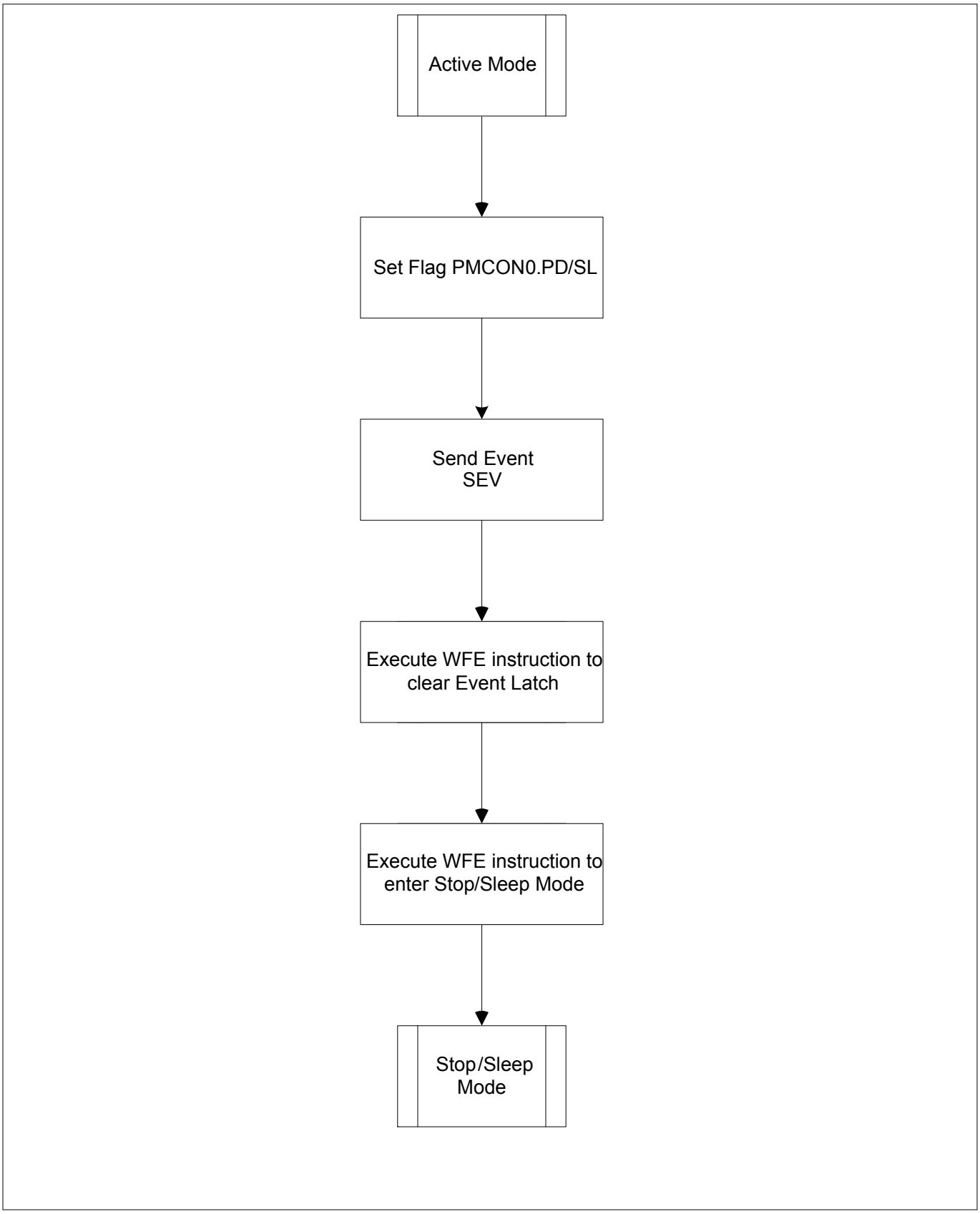


Figure 33 Stop mode entry programming sequence

Exiting stop mode

Stop mode can be exited by active edge on the enabled wake-up pin(s) or by asserting the hard reset pin.

The wake-up circuitry will perform a sequence of predefined actions such as restore all supply voltages, restore modules to operational mode including the oscillator and PLL. On stable clock per user configuration is restored, peripheral clock gating, CPU clock gating is removed and the CPU starts to run from the instruction following the one that sets the PD bit.

Note that if user has selected the PLL output as system clock (typical usage) but lock status of the PLL cannot be achieved, the device cannot wake up and shall hang in this state until a device reset.

6.5.2.2.1 Usage of Arm® core low power modes for stop and sleep mode

The Arm® core provides two low power modes, which are sleep and deep sleep. For stop mode of the system the deep sleep will be used. To enable the deep sleep mode the system control register at address $E000ED10_H$. When the user wants to enter sleep mode it can be done via two different instructions:

- WFI
- WFE

When the controller enters stop mode via WFI instruction, it executes the lowest prior pending interrupt and after that enters sleep mode. This feature is not recommended to be used for normal operation using 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 CPU will be woken up be this external event, it stays in thread mode and continue to execute the code 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 contents of non-volatile memory are retained. As for the external ports, only the wake-up pads are still powered (V_S). The supply to ADC pads is removed.

Sleep mode is always exit with a system reset, which is triggered by active edge on the enabled wake-up pin(s). It is not possible to exit sleep mode by asserting the hard reset pin as the digital 5 V pads will not be powered.

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

Exiting sleep mode

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

Note: Ready for first LIN message at > 400 µs (assume 64 kbyte MapRAM init): start-up boot, NVM pumps ramp up including SFR and MapRAM init.

6.5.3 Power management registers

6.5.3.1 Register overview - Power management registers (ascending offset address)

Table 38 Register overview - Power management registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_PMCON0	Power mode control 0 register	0040_H	171

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 total 24 maskable interrupt nodes. The following figure shows the block diagram of the interrupt management submodule:

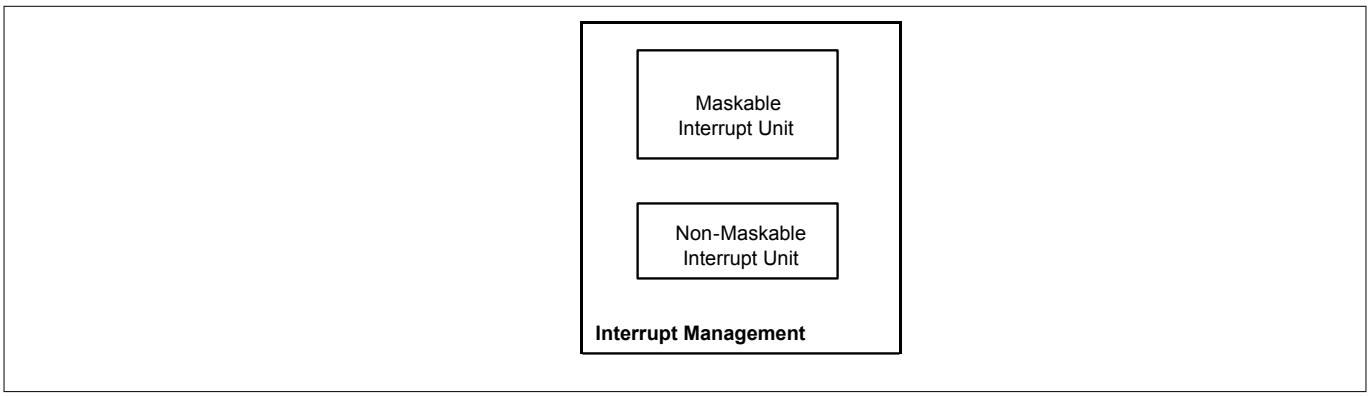


Figure 34 Interrupt management block diagram

The non-maskable interrupt unit controls the NMI requests. Incoming NMI request is not maskable and in this sense, differs from the regular interrupts. In addition, NMI request always has the highest priority to be serviced. In the MOTIX™ TLE984xQX, eight different sources can generate an NMI: 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/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 is via the NMI control register (NMICON).

There are generally 3 types of maskable inputs into the core: internal, external and extended interrupts. The maskable interrupt unit will generate the respective interrupt node request to the core and will maintain 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

The generation of interrupt request from an external source by edge detection in SCU is shown in the following figure. External interrupts can be positive, negative or double edge triggered. Register EXICON0 specifies the active edge for the external interrupt.

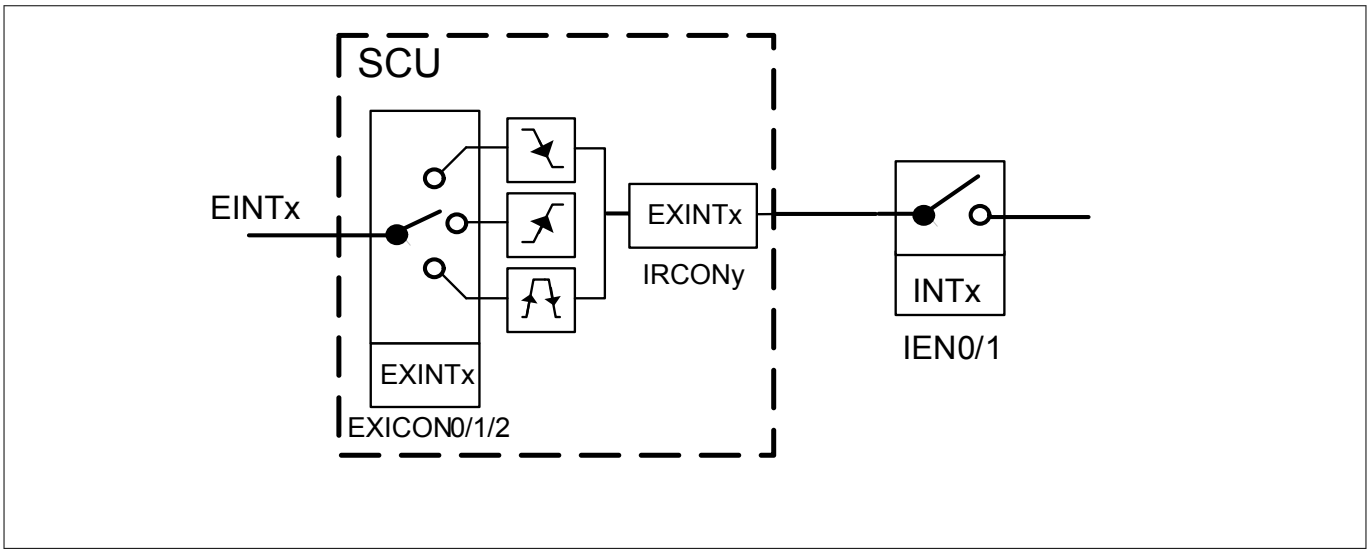


Figure 35 Interrupt request generation of external and peripheral interrupts

6.6.1.2 Extended interrupts

Extended interrupts are for non-core on-chip peripherals for core-external trigger of interrupt requests to the core. There are nine such interrupt request signals.

Interrupt signals from such on-chip peripherals are pulse triggered and active for two clock cycles. These 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) within the core may block or transfer these interrupt requests 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 (e.g. SC) have their own dedicated interrupt node.

6.6.2 Interrupt node assignment

The following table shows the interrupt node assignment for MOTIX™ TLE984xQX:

Table 39 NMI

Interrupt node	Vector address	Assignment for MOTIX™ TLE984xQX
NMI	0000_H	PLL, NVM operation complete, CLKWDT, oscillator watchdog, NVM map error, ECC error, pre-warn SUPP, pre-warn TEMP

Table 40 Interrupt vector table

Service request	Node ID	Description
GPT1	0	GPT1 interrupt (T2-T4)
GPT2	1	GPT2 interrupt (T5-T6, CR)
MU	2	MU interrupt / ADC2, VBG interrupt

(table continues...)

Table 40 (continued) Interrupt vector table

Service request	Node ID	Description
ADC1	3	ADC10 bit interrupt
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), T2, LINSYNC1, LIN
UART2	11	UART2 interrupt (receive, transmit), T21, external interrupt (EINT2)
EXINT0	12	External interrupt (EINT0)
EXINT1	13	External interrupt (EINT1)
WAKE-UP	14	Wake-up interrupt
LS1	17	Low-side 1 interrupt
LS2	18	Low-side 2 interrupt
HS1	19	High-side 1 interrupt
HS2	20	High-side 2 interrupt[1)]
DU	21	Differential unit - DPP1 (only TLE9845QX)
MON1-5	22	MON1-5[2)] interrupt - DPP1
Port 2.x	23	Port 2.x interrupt - DPP1

1) HS2 is device variant specific.
2) MON5 is device variant specific.

6.6.3 Interrupt management registers

Interrupt registers are used for interrupt node enable, external interrupt control, interrupt flags and interrupt priority setting.

The registers are addressed wordwise.

6.6.3.1 Register overview - Interrupt management registers (ascending offset address)

Table 41 Register overview - Interrupt management registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_NMISRCLR	NMI status clear register	0000_H	173
SCU_IRCON0	Interrupt request 0 register	0004_H	175
SCU_IRCON1	Interrupt request 1 register	0008_H	176
SCU_IRCON2	Interrupt request 2 register	$000C_H$	178
SCU_IRCON3	Interrupt request 3 register	0010_H	179
SCU_IRCON4	Interrupt request 4 register	0014_H	180
SCU_NMISR	NMI status register	0018_H	181
SCU_IEN0	Interrupt enable 0 register	$001C_H$	183
SCU_VTOR	Vector table reallocation register	0020_H	184
SCU_NMICON	NMI control register	0024_H	185
SCU_EXICON0	External interrupt control 0 register	0028_H	186
SCU_EXICON1	External interrupt control 1 register	$002C_H$	187
SCU_MODIEN1	Peripheral interrupt enable 1 register	0030_H	188
SCU_MODIEN2	Peripheral interrupt enable 2 register	0034_H	189
SCU_MODIEN3	Peripheral interrupt enable 3 register	0038_H	190
SCU_MODIEN4	Peripheral interrupt enable 4 register	$003C_H$	191
SCU_WAKECON	Wake-up interrupt control register	0078_H	192
SCU_IRCON5	Interrupt request 5 register	$007C_H$	193
SCU_GPT12IEN	General purpose timer 12 interrupt enable register	$015C_H$	194
SCU_GPT12IRC	Timer and counter control/status register	0160_H	195
SCU_IRCON0CLR	Interrupt request 0 clear register	0178_H	196
SCU_IRCON1CLR	Interrupt request 1 clear register	$017C_H$	197
SCU_GPT12ICLR	Timer and counter control/status clear register	0180_H	199
SCU_MONIEN	Monitoring input interrupt enable register	$018C_H$	200
SCU_IRCON2CLR	Interrupt request 2 clear register	0190_H	201
SCU_IRCON3CLR	Interrupt request 3 clear register	0194_H	202
SCU_IRCON4CLR	Interrupt request 4 clear register	0198_H	203
SCU_IRCON5CLR	Interrupt request 5 clear register	$019C_H$	204

6.6.3.2 Interrupt node enable registers

Register SCU_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 SCU_NMICON.

After reset, the enable bits in SCU_IEN0, SCU_IEN1 and SCU_NMICON are cleared to 0. This implies that all interrupt nodes are disabled by default.

6.6.3.3 External interrupt control registers

The external interrupts are driven into the MOTIX™ TLE984xQX from the ports. External interrupts can be positive, negative or double edge triggered. Register SCU_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 interrupt 2 share the interrupt node with other interrupt sources. Therefore in addition to the corresponding interrupt node enable, external interrupt 2 may be disabled individually, and is disabled by default after reset.

Note: Several external interrupts support alternative input pin, selected via SCU_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.

6.6.3.4 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. For a complete listing of the interrupt flags and their assignment to SFRs, refer to Table 68.

In case of software and hardware access to a flag bit at the same time, hardware will have higher priority.

6.6.3.5 Interrupt related registers

Several interrupt related registers are located in the SCU.

6.6.4 NMI event flags handling

Each NMI event and status flag is retained across soft reset. Specifically, this includes all the flags of SCU_NMISR register: FNMIPLL, FNMINVM, FNMIOCDS, FNMIOWD, FNMIMAP and indirectly, FNMIECC and FNMISUP. In the case of watchdog resets, the requestor can be identified via the reset indicator bit WDT1RST. 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 chapter. These NMI event and status flags are otherwise reset to default value with all other resets that is power-on, brown-out, hardware, WDT1 (except WDT1RST) and wake-up reset.

6.7 General port control

6.7.1 General port control registers

The SCU contains control registers for the selection of:

- Alternate input functions of UART, timers and external interrupts (Chapter 6.7.1.2)
- Port output driver strength and temperature compensation (Chapter 6.7.1.3)

For functional description of GPIO ports, refer to Chapter 14.

6.7.1.1 Register overview - General port control registers (ascending offset address)

Table 42 Register overview - General port control registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_MODPISEL	Peripheral input select register	$00B8_H$	205
SCU_MODPISEL1	Peripheral input select 1 register	$00BC_H$	207
SCU_MODPISEL2	Peripheral input select 2 register	$00C0_H$	208
SCU_MODPISEL3	Peripheral input select 3 register	$00C4_H$	209
SCU_GPT12PISEL	GPT12 peripheral input select register	$00D0_H$	210
SCU_P0_POCON0	Port output control register	$00E8_H$	211
SCU_TCCR	Temperature compensation control register	$00F4_H$	213
SCU_P1_POCON0	Port output control register	$00F8_H$	214

6.7.1.2 Input pin function selection registers

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

6.7.1.3 Port output control registers

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

6.7.1.4 GPT12 T3IN/T4IN input pin function selection registers

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

6.8 Differential unit trigger enable (only TLE9845QX)

The differential unit inside DPP1 module requires enable signals for telling the processing when to accept and calculate a new result based on an incoming trigger signal. To realize a certain blank timer for the DU Unit to perform the measurements aligned to the dedicated PWM Signal the Timer 13 of CCU6 is used.

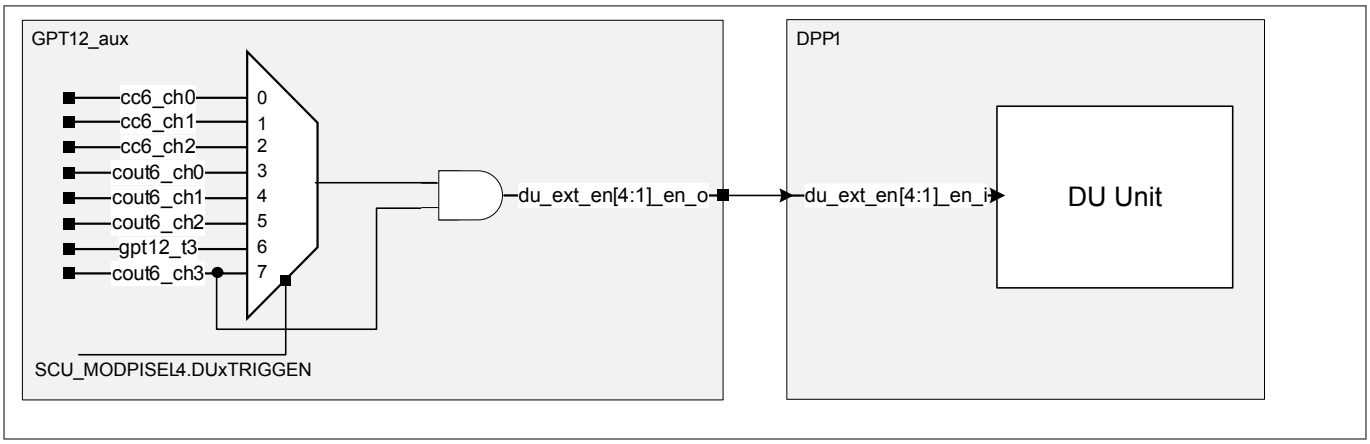


Figure 36 Differential unit

6.8.1 Differential unit trigger register

6.8.1.1 Register overview - Differential unit trigger registers for TLE9845QX only (ascending offset address)

Table 43 Register overview - Differential unit trigger registers for TLE9845QX only (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_MODPISEL4	Peripheral input select 4 register	$00FC_H$	216

6.9 Flexible peripheral management

The flexible peripheral management submodule provides the system designer greater control on the operational status of each individual digital peripheral. Peripherals which are not required for a particular functionality can be disabled by programming the assigned register bits which would gate off the clock inputs. This would further reduce 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 will then synchronize the gating off of the clock to the peripheral.

6.9.1 Peripheral management registers

6.9.1.1 Register overview - Flexible peripheral management registers (ascending offset address)

Table 44 Register overview - Flexible peripheral management registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_PMCON	Peripheral management control register	0060_H	218

6.10 Module suspend control

When the on-chip debug support (debug mode) is in monitor mode (halted_o from Arm® debug), timers in certain modules in MOTIX™ TLE984xQX can be suspended based on the settings of their corresponding module suspend bits in register MODSUSP. When 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.

6.10.1 Module suspend control registers

6.10.1.1 Register overview - Module suspend control registers (ascending offset address)

Table 45 Register overview - Module suspend control registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_MODSUSP	Module suspend control register	$00C8_H$	219

6.11 Baud-rate generator

The baud-rate generator in SCU is used to generate the baud-rate for the UART module. See Chapter 18.6 for the functional description. The SCU contains two of this registers. One is dedicated for UART1 and the other for UART2.

6.11.1 Baud-rate generator registers

6.11.1.1 Register overview - Baud-rate generator registers (ascending offset address)

Table 46 Register overview - Baud-rate generator registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_BCON1	Baud-rate control 1 register	0088_H	221
SCU_BGL1	Baud-rate timer/reload, low byte 1 register	$008C_H$	222
SCU_BG1	Baud-rate timer/reload 1 register	0090_H	223
SCU_BCON2	Baud-rate control 2 register	0098_H	224
SCU_BGL2	Baud-rate timer/reload, low byte 2 register	$009C_H$	225
SCU_BG2	Baud-rate timer/reload 2 register	$00A0_H$	226

6.11.1.2 Baud-rate generator control and status registers

6.11.1.3 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. Also this register should be present twice. One is for UART1 and the other for UART2.

6.12 LIN break and sync byte detection

Hardware logic is implemented in the SCU to support LIN break and sync byte detection. See Chapter 18.7 for the functional description.

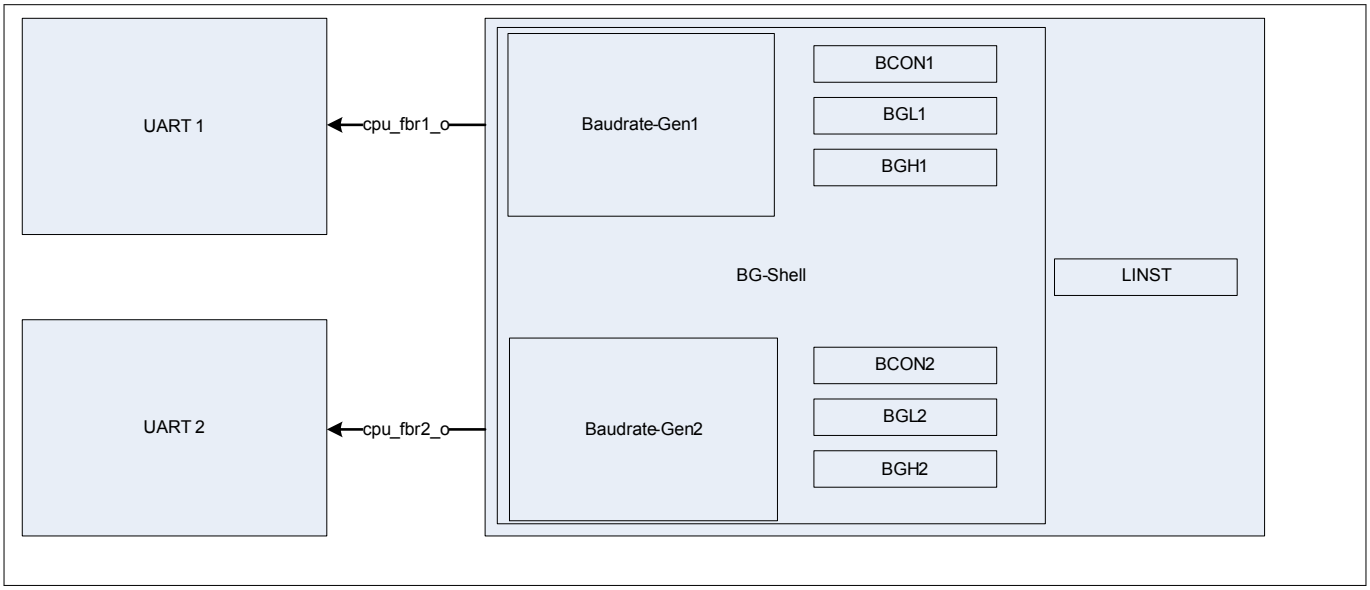


Figure 37 Structure of baud-rate generator

6.12.1 LIN break and sync byte detection control registers

6.12.1.1 Register overview - LIN break and sync byte registers (ascending offset address)

Table 47 Register overview - LIN break and sync byte registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_LINST	LIN status register	0094_H	227
SCU_LINSCLR	LIN status clear register	$00A4_H$	228

6.13 Error detection and correction control for memories

This section defines the registers used for error detection and correction control of memories – namely RAM and NVM, which support this function.

6.13.1 Error detection and correction control for memories registers

6.13.1.1 Register overview - Error detection and correction control for memories registers (ascending offset address)

Table 48 Register overview - Error detection and correction control for memories registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_EDCCON	Error detection and correction control register	$00D4_H$	229
SCU_EDCSTAT	Error detection and correction status register	$00D8_H$	230
SCU_EDCSCLR	Error detection and correction status clear register	$010C_H$	231

6.13.1.2 Error detection and correction control register

The SCU_EDCCON register determines the generation of an NMI due to double bit ECC error when read these memories.

6.13.1.3 Error detection and correction status register

The SCU_EDCSTAT register contains the status flags of ECC errors when read these memories. The corresponding flags for the IRAM are not more necessary, because IRAM was removed.

6.14 Miscellaneous control

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

6.14.1 Miscellaneous control registers

6.14.1.1 Register overview - Miscellaneous control registers (ascending offset address)

Table 49 Register overview - Miscellaneous control registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_SYS_STRTUP_STS	System startup status register	0074_H	232
SCU_ID	Identity register	$00A8_H$	234
SCU_PASSWD	Password register	$00AC_H$	235
SCU_EMOP	Emergency and program operation status register	$00CC_H$	236
SCU_MEMSTAT	Memory status register	$00DC_H$	237
SCU_NVM_PROT_STS	NVM protection status register	$00E0_H$	238
SCU_MEM_ACC_STS	Memory access status register	$00E4_H$	240

6.14.1.2 Bit protection register

The bit protection scheme does not allow the writing of the protected bits, as listed in the table below.

There are 2 ways of disabling the bit protection scheme:

- The first way is to disable it temporarily:
  - Writing 10011_B to the PASS bit field, opens access to writing (so disable the bit protection scheme) **for maximum 32 clock cycles**
  - Writing 10101_B to the PASS bit field, closes access to writing previously open
- The second way is to disable it permanently:
  - Writing 11000_B to the PASS bit field allows to write the PW_MODE bit field
  - Writing the PW_MODE bit field with 00_B, allows to disable **permanently** the bit protection scheme

Note that access is opened for maximum 32 CCLKs if the “close access” password is not written. If “open access” password is written again before the end of 32 CCLK cycles, there will be a recount of 32 CCLK cycles.

Note: It is recommended to disable interrupts before writing to this register, otherwise interrupts might delay the write access the protected bits in a way that the 32-cycles-access-window is closed already.

The SCU_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 below.

Table 50 List of protected bits

Register	Bit field
SCU_SYSCON0	SYSCLKSEL
SCU_RSTCON	LOCKUP_EN

(table continues...)

Table 50 (continued) List of protected bits

Register	Bit field
	LOCKUP
SCU_OSC_CON	OSCSS
	XPD
SCU_PLL_CON	NDIV
SCU_CMCON1	K1DIV
	K2DIV
	PDIV
SCU_CMMON2	PBA0CLKREL
SCU_APCLK_CTRL	CLKWDT_IE
SCU_APCLK	BGCLK_DIV
	BGCLK_SEL
SCU_PMCON0	SD
	PD
	SL
SCU_MODPISEL1	XTAL12EN

6.14.1.3 System control and status registers

The system startup status register provide information to the user about the system initialization with the user programmable 100 TP Page at startup. These register is written by firmware.

This register SYS__STS is reset by reset_type_4.

6.15 System control unit - digital modules (SCU) register definition

This chapter contains an overview of all SCU registers.

Note: HS2 and MON5 are device variant specific. In devices featuring only HS1 the HS2_XXX bitfields can be ignored. In devices featuring only MON1-4 the HS MON5_XXX bitfields can be ignored. Writing to these bitfields has no effect.

The registers are addressed wordwise.

6.15.1 Register address space - SCUDM

Table 51 Registers address space - SCUDM

Module	Base address	End address	Note
SCU	50005000_H	$50005FFF_H$	System Control Unit - Digital Modules (SCU-DM) registers

6.15.2 Register overview - SCUDM (ascending offset address)

Table 52 Register overview - SCUDM (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_NMISRCLR	NMI status clear register	0000_H	173
SCU_IRCON0	Interrupt request 0 register	0004_H	175
SCU_IRCON1	Interrupt request 1 register	0008_H	176
SCU_IRCON2	Interrupt request 2 register	$000C_H$	178
SCU_IRCON3	Interrupt request 3 register	0010_H	179
SCU_IRCON4	Interrupt request 4 register	0014_H	180
SCU_NMISR	NMI status register	0018_H	181
SCU_IEN0	Interrupt enable 0 register	$001C_H$	183
SCU_VTOR	Vector table reallocation register	0020_H	184
SCU_NMICON	NMI control register	0024_H	185
SCU_EXICON0	External interrupt control 0 register	0028_H	186
SCU_EXICON1	External interrupt control 1 register	$002C_H$	187
SCU_MODIEN1	Peripheral interrupt enable 1 register	0030_H	188
SCU_MODIEN2	Peripheral interrupt enable 2 register	0034_H	189
SCU_MODIEN3	Peripheral interrupt enable 3 register	0038_H	190
SCU_MODIEN4	Peripheral interrupt enable 4 register	$003C_H$	191
SCU_PMCON0	Power mode control 0 register	0040_H	171
SCU_PLL_CON	PLL control register	0044_H	153
SCU_CMCON1	Clock control 1 register	0048_H	155
SCU_CMCON2	Clock control 2 register	$004C_H$	157
SCU_APCLK_CTRL	Analog peripheral clock control register	0054_H	158
SCU_APCLK	Analog peripheral clock register	0058_H	159

(table continues...)

Table 52 (continued) Register overview - SCUDM (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_APCLK_STS	Analog peripheral clock status register	$005C_H$	161
SCU_PMCON	Peripheral management control register	0060_H	218
SCU_APCLK_SCLR	Analog peripheral clock status clear register	0064_H	163
SCU_RSTCON	Reset control register	0068_H	170
SCU_ADC1_CLK	ADC1 peripheral clock register	$006C_H$	164
SCU_SYSCON0	System control 0 register	0070_H	165
SCU_SYS_STRTUP_STS	System startup status register	0074_H	232
SCU_WAKECON	Wake-up interrupt control register	0078_H	192
SCU_IRCON5	Interrupt request 5 register	$007C_H$	193
SCU_BCON1	Baud-rate control 1 register	0088_H	221
SCU_BGL1	Baud-rate timer/reload, low byte 1 register	$008C_H$	222
SCU_BG1	Baud-rate timer/reload 1 register	0090_H	223
SCU_LINST	LIN status register	0094_H	227
SCU_BCON2	Baud-rate control 2 register	0098_H	224
SCU_BGL2	Baud-rate timer/reload, low byte 2 register	$009C_H$	225
SCU_BG2	Baud-rate timer/reload 2 register	$00A0_H$	226
SCU_LINSCLR	LIN status clear register	$00A4_H$	228
SCU_ID	Identity register	$00A8_H$	234
SCU_PASSWD	Password register	$00AC_H$	235
SCU_OSC_CON	OSC control register	$00B0_H$	166
SCU_COCON	Clock output control register	$00B4_H$	168
SCU_MODPISEL	Peripheral input select register	$00B8_H$	205
SCU_MODPISEL1	Peripheral input select 1 register	$00BC_H$	207
SCU_MODPISEL2	Peripheral input select 2 register	$00C0_H$	208
SCU_MODPISEL3	Peripheral input select 3 register	$00C4_H$	209
SCU_MODSUSP	Module suspend control register	$00C8_H$	219
SCU_EMOP	Emergency and program operation status register	$00CC_H$	236
SCU_GPT12PISEL	GPT12 peripheral input select register	$00D0_H$	210
SCU_EDCCON	Error detection and correction control register	$00D4_H$	229
SCU_EDCSTAT	Error detection and correction status register	$00D8_H$	230
SCU_MEMSTAT	Memory status register	$00DC_H$	237
SCU_NVM_PROT_STS	NVM protection status register	$00E0_H$	238
SCU_MEM_ACC_STS	Memory access status register	$00E4_H$	240
SCU_P0_POCON0	Port output control register	$00E8_H$	211
SCU_TCCR	Temperature compensation control register	$00F4_H$	213

(table continues...)

Table 52 (continued) Register overview - SCUDM (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_P1_POCON0	Port output control register	$00F8_H$	214
SCU_MODPISEL4	Peripheral input select 4 register	$00FC_H$	216
SCU_EDCSCLR	Error detection and correction status clear register	$010C_H$	231
SCU_GPT12IEN	General purpose timer 12 interrupt enable register	$015C_H$	194
SCU_GPT12IRC	Timer and counter control/status register	0160_H	195
SCU_IRCON0CLR	Interrupt request 0 clear register	0178_H	196
SCU_IRCON1CLR	Interrupt request 1 clear register	$017C_H$	197
SCU_GPT12ICLR	Timer and counter control/status clear register	0180_H	199
SCU_MONIEN	Monitoring input interrupt enable register	$018C_H$	200
SCU_IRCON2CLR	Interrupt request 2 clear register	0190_H	201
SCU_IRCON3CLR	Interrupt request 3 clear register	0194_H	202
SCU_IRCON4CLR	Interrupt request 4 clear register	0198_H	203
SCU_IRCON5CLR	Interrupt request 5 clear register	$019C_H$	204

6.15.3 Register overview - UART1/2 control/status registers

Table 53 Register overview - UART1/2 control/status registers

Short name	Long name	Offset address	Page number
SCU_SCON1	UART1 control/status register	XXX_H	241
SCU_SCON2	UART2 control/status register	XXX_H	242

6.15.4 PLL control register

SCU_PLL_CON Offset address: 0044_H

PLL control register RESET_TYPE_4 value: $0000\ 00A4_H$

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				UNPROT_VCOBYP	UNPROT_OSCDISC	RES		NDIV				VCOBYP	OSCDISC	RESLD	LOCK
r				w	w	r		rwpw				rw	rw	rw	r

Field	Bits	Type	Description
LOCK	0	r	**PLL lock status flag** **Notes** **1.** In case of a loss of VCO lock the f_{VCO} goes to the upper boundary of the selected VCO band if the reference clock input is greater as expected. **2.** In case of a loss of VCO lock the f_{VCO} goes to the lower boundary of the selected VCO band if the reference clock input is lower as expected. **3.** On loss-of-lock detection (LOCK: 1 →0) and when VCOBYP = 0, PLL switches to free running mode. **4.** Loss-of-lock NMI request is activated only on loss-of-lock detection when VCOBYP = 0 and SYSCON0.SYSCLKSEL selects PLL clock as system frequency. 0_B **UNLOCKED**: The frequency difference of fREF and fDIV is greater than allowed. The VCO part of the PLL can not lock on a target frequency 1_B **LOCKED**: The frequency difference of fREF and fDIV is small enough to enable a stable VCO operation
RESLD	1	rw	**Restart lock detection** Setting this bit will reset the PLL lock status flag and restart the lock detection. This bit will be automatically reset to 0 and thus always be read back as 0. 0_B **NO**: No effect 1_B **RESET**: Reset lock flag and restart lock detection
OSCDISC	2	rw	**Oscillator disconnect** By default after power-on reset, PLL is running in free running mode (osc is disconnected). 0_B **CONNECTED**: Oscillator is connected to the PLL 1_B **DISCONNECTED**: Oscillator is disconnected to the PLL
VCOBYP	3	rw	**PLL VCO bypass mode select**

(table continues...)

(continued)

Field	Bits	Type	Description
			This bit is cleared by hardware when PLL switches to free running mode. When the bit value changes from 0 to 1, bit OSCDISC = 0. 0_B **NORMAL**: Normal (or free running) operation (default) 1_B **PRESCALER**: Prescaler mode; VCO is bypassed (PLL output clock is derived from input clock divided by K1-divider)
NDIV	7:4	rwpw	**PLL N-divider** This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly. 0_H **48**: N = 48 1_H **50**: N = 50 2_H **51**: N = 51 3_H **52**: N = 52 4_H **54**: N = 54 5_H **60**: N = 60 6_H **67**: N = 67 7_H **72**: N = 72 8_H **75**: N = 75 9_H **78**: N = 78 A_H **80**: N = 80 B_H **88**: N = 88 C_H **90**: N = 90 D_H **94**: N =94 E_H **100**: N = 100 F_H **160**: N = 160
RES	9:8, 31:12	r	**Reserved** Returns 0 if read; should be written with 0.
UNPROT_OSCDISC	10	w	**Unprotect write access of OSC_DISC** Writing this bit within an write access of OSCDISC will overtake the provided value to OSCDISC without protection Note: Read is always ‘0’
UNPROT_VCOBYP	11	w	**Unprotect write access of VCO_BYP** Writing this bit within an write access of VCO_BYP will overtake the provided value to VCO_BYP without protection Note: Read is always ‘0’

6.15.5 Clock control 1 register

SCU_CMCON1 Offset address: 0048_H

Clock control 1 register RESET_TYPE_4 value: $0000\ 0100_H$

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						PDIV		RES	K1DIV	K2DIV		CLKREL			
r						rwpw		r	rwpw	rwpw		rw			

Field	Bits	Type	Description
CLKREL	3:0	rw	**Slow down clock divider for fCCLK generation** This setting is effective only when the device is enabled in slow down mode. *Note: f_{sys} is further divided by the NVMCLKFAC factor to generate f_{CCLK}.* 0_H **1**: fSYS 1_H **2**: fSYS/2 2_H **3**: fSYS/3 3_H **4**: fSYS/4 4_H **8**: fSYS/8 5_H **16**: fSYS/16 6_H **24**: fSYS/24 7_H **32**: fSYS/32 8_H **48**: fSYS/48 9_H **64**: fSYS/64 A_H **96**: fSYS/96 B_H **128**: fSYS/128 C_H **192**: fSYS/192 D_H **256**: fSYS/256 E_H **384**: fSYS/384 F_H **512**: fSYS/512
K2DIV	5:4	rwpw	**PLL K2-divider** Depending on f_{VCO}, the user has to set the K2-divider factor large enough to ensure the PLL output frequency in free running mode is never higher than that specified for the device. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 00_B **2**: K2 = 2

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **3**: K2 = 3 10_B **4**: K2 = 4 11_B **5**: K2 = 5
K1DIV	6	rwpw	**PLL K1-divider** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **2**: K1 = 2 1_B **1**: K1 = 1
RES	7, 31:10	r	**Reserved** Returns 0 if read; should be written with 0.
PDIV	9:8	rwpw	**PLL PDIV-divider** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 00_B **4**: 4 01_B **5**: 5 (default) 10_B **6_1**: 6 11_B **6_2**: 6

6.15.6 Clock control 2 register

SCU_CMCON2 Offset address: $004C_H$

Clock control 2 register RESET_TYPE_4 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															**PBA0 CLK REL**
r															rwpw

Field	Bits	Type	Description
PBA0CLKREL	0	rwpw	**PBA0 clock divider** This flag configures the PBA0 clock divider. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **1**: Divide by 1 1_B **2**: Divide by 2
RES	31:1	r	**Reserved** This bit field is always read as 0.

6.15.7 Analog peripheral clock control register

SCU_APCLK_CTRL Offset address: 0054_H

Analog peripheral clock control register RESET_TYPE_4 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							CLKWDT_IE	RES							APCLK_SET
r							rwpw	r							rwh1

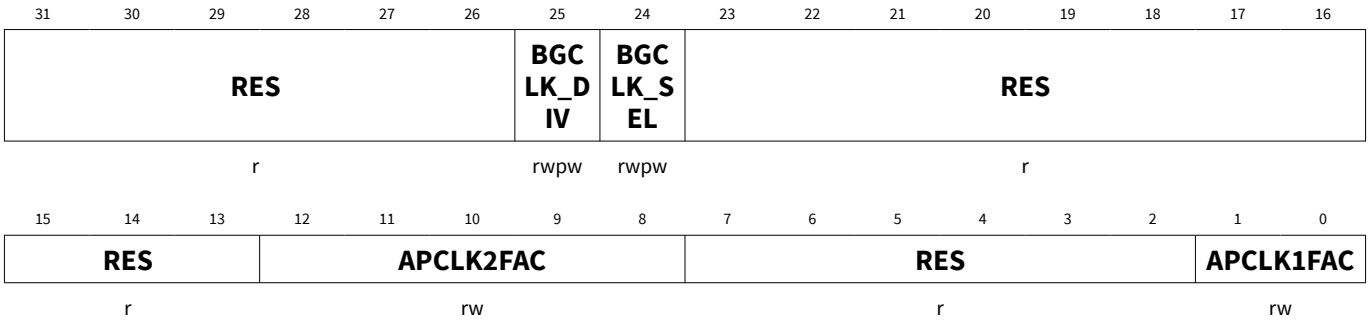
Field	Bits	Type	Description
APCLK_SET	0	rwh1	**Set and overtake flag for clock settings** This flag makes the APCLK1, APCLK2 settings valid. *Note: If APCLK_SET is cleared by hardware once the clock setting are overtaken.* 0_B **IGNORED**: Clock settings are ignored (previous values are held) 1_B **OVERTAKEN**: Clock settings are overtaken
RES	7:1, 31:9	r	**Reserved** Returns 0 if read; should be written with 0.
CLKWDT_IE	8	rwpw	**Clock watchdog interrupt enable** Returns 0 if read; should be written with 0. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled

6.15.8 Analog peripheral clock register

The clock source for the analog modules is selected via register APCLK.

SCU_APCLK Offset address: 0058_H

Analog peripheral clock register RESET_TYPE_4 value: $0000\ 0000_H$



Field	Bits	Type	Description
APCLK1FAC	1:0	rw	**Analog module clock factor** This bit field defines the factor by which the system clock is divided down, with respect to the synchronous MI_CLK clock. The APCLKFAC bit is not protected. This setting is only effective when APCLK_SET = 1. *Note: If SYSCLKSEL[1] = '1' (LP_CLK) APCLK1FAC should be set to "00" (divide by 1).* *If SYCLKSEL[1:0] = "11" APCLK1FAC should be set to "01".* 00_B **1**: Divide by 1 01_B **2**: Divide by 2 10_B **3**: Divide by 3 11_B **4**: Divide by 4
RES	7:2, 23:13, 31:26	r	**Reserved** Always read as 0.
APCLK2FAC	12:8	rw	**Slow down clock divider for TFILT_CLK generation** This setting is effective only when the APCLK_SET = 1. Other bit combinations equivalent. *Note: If SYSCLKSEL[1:0] = '10' (f_{LP_CLK}) APCLK2FAC should be set to '8'.* *If SYSCLKSEL[1:0] = '11' (f_{INTOSC}) APCLK2FAC should be set to '19'.* *f_{SYS} is further divided by the APCLK2FAC factor to generate TFILT_CLK. The clock should be always at 2 MHz.* 00_H **1**: fSYS 01_H **2**: fSYS/2 02_H **3**: fSYS/3

(table continues...)

(continued)

Field	Bits	Type	Description
			03_H **4**: fSYS/4 04_H **5**: fSYS/5 05_H **6**: fSYS/6 06_H **7**: fSYS/7 07_H **8**: fSYS/8 08_H **9**: fSYS/9 09_H **10**: fSYS/10 $0A_H$ **11**: fSYS/11 $0B_H$ **12**: fSYS/12 $1E_H$ **24**: fSYS/24 $1F_H$ **32**: fSYS/32
BGCLK_SEL	24	rwpw	**Bandgap clock selection** This flag selects the bandgap clock. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* *Note: If SYSCLKSEL[1] = '1' the default BGCLK_SEL = "0" (LP_CLK) is taken.* 0_B **LP_CLK**: LP_CLK is selected 1_B **fSYS**: fSYS is selected
BGCLK_DIV	25	rwpw	**Bandgap clock divider** This flag configures the bandgap clock divider. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **2**: Divide by 2 1_B **1**: Divide by 1

6.15.9 Analog peripheral clock status register

SCU_APCLK_STS Offset address: $005C_H$

Analog peripheral clock status register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES							PLL_LOCK	RES							APCLK3STS
r							r	r							r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						APCLK2STS		RES						APCLK1STS	
r						r		r						r	

Field	Bits	Type	Description
APCLK1STS	1:0	r	**Analog peripherals clock status** This bit field reflects the analog peripheral clock source status that is used as system clock for the analog module operation. The implemented clock watchdog (see Chapter SCU_PM) is monitoring the frequency of the analog subsystem. *Note: The functionality of the analog modules can only be guaranteed, when their clock is in the required range.* 00_B **RANGE**: The MI_CLK clock is in the required range 01_B **HIGHER**: The MI_CLK clock exceeds the higher limit 10_B **LOWER**: The MI_CLK clock exceeds the lower limit 11_B **OUTSIDE**: The MI_CLK clock is not inside the specified limit
RES	7:2, 15:10, 23:17, 31:25	r	**Reserved** Returns 0 if read; should be written with 0.
APCLK2STS	9:8	r	**Analog peripherals clock status** This bit field reflects the analog peripheral clock source status that is used as system clock for the analog module operation. The implemented clock watchdog (see Chapter SCU_PM) is monitoring the frequency of the analog subsystem. *Note: The functionality of the analog modules can only be guaranteed, when their clock is in the required range.* 00_B **RANGE**: The TFILT_CLK clock is in the required range 01_B **HIGHER**: The TFILT_CLK clock exceeds the higher limit 10_B **LOWER**: The TFILT_CLK clock exceeds the lower limit 11_B **OUTSIDE**: The TFILT_CLK clock is not inside the specified limit
APCLK3STS	16	r	**Loss of clock status** This bit field indicate the loss of clock status.

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **NO_LOSS**: No loss of clock 1_B **LOSS**: Loss of lock occurred
PLL_LOCK	24	r	**PLL LOCK status** This bit field indicates the PLL lock status. 0_B **NOT_LOCKED**: PLL has not locked 1_B **LOCKED**: PLL has locked

6.15.10 Analog peripheral clock status clear register

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

SCU_APCLK_SCLR Offset address: 0064_H

Analog peripheral clock status clear register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES							PLL_LOCK_SCLR	RES							APCLK3SCLR
r							w	r							w

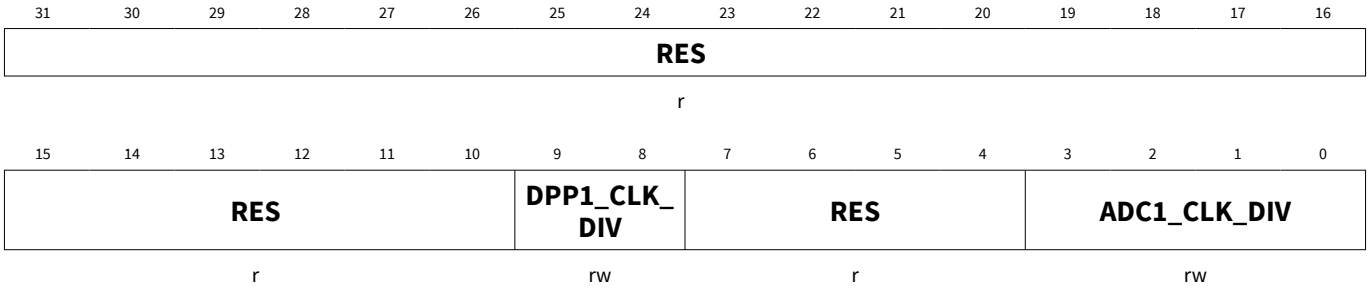
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES							APCLK2SCLR	RES							APCLK1SCLR
r							w	r							w

Field	Bits	Type	Description
APCLK1SCLR	0	w	**Analog peripherals clock status clear** This bit field is used for APCLK1 status clear.
RES	7:1, 15:9, 23:17, 31:25	r	**Reserved** Returns 0 if read; should be written with 0.
APCLK2SCLR	8	w	**Analog peripherals clock status clear** This bit field is used for APCLK2 status clear.
APCLK3SCLR	16	w	**Analog peripherals clock 3 status clear** This bit field is used for APCLK3 status clear.
PLL_LOCK_SCLR	24	w	**PLL lock status clear** This bit field is used for PLL_LOCK status clear.

6.15.11 ADC1 peripheral clock register

SCU_ADC1_CLK Offset address: $006C_H$

ADC1 peripheral clock register RESET_TYPE_4 value: $0000\ 0000_H$



Field	Bits	Type	Description
ADC1_CLK_DIV	3:0	rw	**ADC1 clock divider** This bit field defines the factor by which the divided system clock from DPP1_CLK_DIV is divided additionally for ADC1 core clock. 0_H **1**: Divide by 1 … 4_H **5**: Divide by 5 6_H **16**: Divide by 16
RES	7:4, 31:10	r	**Reserved** Returns 0 if read; should be written with 0.
DPP1_CLK_DIV	9:8	rw	**ADC1 post processing clock divider** This bit field defines the factor by which the system clock is divided for the post processing of ADC1. 00_B **1**: Divide by 1 01_B **2**: Divide by 2 10_B **3**: Divide by 3 11_B **4**: Divide by 4

6.15.12 System control 0 register

SCU_SYSCON0 Offset address: 0070_H

System control 0 register RESET_TYPE_4 value: $0000\ 0080_H$

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								SYSCLKSEL		NVMCLKFAC		RES			
r								rw		rw		r			

Field	Bits	Type	Description
RES	3:0, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
NVMCLKFAC	5:4	rw	**NVM access clock factor** This bit field defines the factor by which the system clock is divided down, with respect to the synchronous NVMACCCLK clock. *Note: Can only be changed via dedicated BROM routine.* 00_B **1**: Divide by 1 01_B **2**: Divide by 2 10_B **3**: Divide by 3 11_B **4**: Divide by 4
SYSCLKSEL	7:6	rw	**System clock select** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* This bit field defines the clock source that is used as system clock for the system operation. *Note: In normal application, it is expected that the system is running on the PLL clock output.* 00_B **PLL**: The PLL clock output signal fPLL is used 01_B **OSC**: The direct clock input from fOSC is used 10_B **LP_CLK**: The direct low-precision clock input from fLP_CLK is used 11_B **INTOSC**: The direct input from internal oscillator fINTOSC is used

6.15.13 OSC control register

SCU_OSC_CON Offset address: $00B0_H$

OSC control register RESET_TYPE_4 value: $0000\ 0010_H$

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		XPD	OSC 2L	OSC WDT RST	OSCSS	
r									r		rwpw	r	rwh1	rwpw	

Field	Bits	Type	Description
OSCSS	1:0	rwpw	**Oscillator source select** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* **Notes** **1.** Synchronous switching of clock source to internal oscillator is not possible when XPD = 1 or no external clock is available (check bit OSC2L). **2.** Use the 1X option only when the external clock is not available. 00_B **PLL_SYNC**: PLL internal oscillator OSC_PLL (fINT) is selected synchronously as fR 01_B **XTAL**: XTAL (fOSC from OSC_HP) is selected synchronously as fR 10_B **PLL_ASYNC**: PLL internal oscillator OSC_PLL (fINT) is selected asynchronously as fR 11_B **PLL_ASYNC**: PLL internal oscillator OSC_PLL (fINT) is selected asynchronously as fR
OSCWDTRST	2	rwh1	**Oscillator watchdog reset** Setting this bit will reset the OSC2L status flag to 1 and restart the oscillator detection. This bit will be automatically reset to 0 and thus always be read back as 0. 0_B **NO_EFFECT**: No effect 1_B **RESET**: Reset OSC2L flag and restart the oscillator watchdog of the PLL
OSC2L	3	r	**OSC-too-low condition flag** The oscillator watchdog monitors the f_{OSC}. On OSC-too-low detection (OSC2L: 0 →1) and VCOBYP = 1 and OSCSS = 01, PLL switches to free running mode. On above condition, and when f_{OSC} is selected as the system clock source, hardware switches the system clock source to PLL (SCU_SYSCON0.SYSCLKSEL is also updated).

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: OWD NMI request is activated on OSC-too-low condition only in two cases: 1) when VCOBYP = 1 and OSCSS = 01 and SYSCLKSEL selects PLL clock as system clock source; 2) when SYSCLKSEL selects f_{OSC} as system clock source. 0_B **ABOVE**: fOSC is above threshold 1_B **BELOW**: fOSC is below threshold
XPD	4	rwpw	**XTAL (OSC_HP) power down control** This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly. *Note: When XPD is set, switch of clock source to internal oscillator has to be done asynchronous.* 0_B **NOT_POWERED**: XTAL (OSC_HP) is not powered down 1_B **POWERED**: XTAL (OSC_HP) is powered down
RES	6:5, 31:7	r	**Reserved** This bit field is always read as 0.

6.15.14 Clock output control register

SCU_COCON Offset address: $00B4_H$

Clock output control register RESET_TYPE_4 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								EN	COUTS1	TLEN	COUTS0	COREL			
r								rw	rw	rw	rw	rw			

Field	Bits	Type	Description
COREL	3:0	rw	**Clock output divider** 0_H **1**: fSYS 1_H **2**: fSYS/2 2_H **3**: fSYS/3 3_H **4**: fSYS/4 4_H **6**: fSYS/6 5_H **8**: fSYS/8 6_H **10**: fSYS/10 7_H **12**: fSYS/12 8_H **14**: fSYS/14 9_H **16**: fSYS/16 A_H **18**: fSYS/18 B_H **20**: fSYS/20 C_H **24**: fSYS/24 D_H **32**: fSYS/32 E_H **36**: fSYS/36 F_H **40**: fSYS/40
COUTS0	4	rw	**Clock out source select bit 0** This bit is effective only if COUTS1 is set to 1. 0_B **OSCILLATOR**: Oscillator output frequency is selected 1_B **COREL**: Clock output frequency is chosen by the bit field COREL
TLEN	5	rw	**Toggle latch enable** Enable this bit if 50% duty cycle is desired on CLKOUT. This bit is only applicable when both COUTS1 and COUTS0 are set to 1. 0_B **DISABLED**: Toggle latch is disabled. Clock output frequency is chosen by the bit field COREL 1_B **ENABLED**: Toggle latch is enabled. Clock output frequency is half of the frequency that is chosen by the bit field COREL. The resulting output frequency has 50% duty cycle
COUTS1	6	rw	**Clock out source select bit 1** 0_B **fCCLK**: fCCLK is selected

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **COUTS0**: Based on setting of COUTS0
EN	7	rw	**CLKOUT enable** 0_B **NO_EXTERNAL**: No external clock signal is provided 1_B **EXTERNAL**: The configured external clock signal is provided
RES	31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.15 Reset control register

SCU_RSTCON Offset address: 0068_H

Reset control register RESET_TYPE_3 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								LOCKUP_EN	RES						LOCKUP
r								rwpw	r						rwpw

Field	Bits	Type	Description
LOCKUP	0	rwpw	**Lockup flag** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **NOT_ACTIVE**: Lockup status not active 1_B **ACTIVE**: Lockup status active
RES	6:1, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
LOCKUP_EN	7	rwpw	**Lockup reset enable flag** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **DISABLED**: Lockup is disabled 1_B **ENABLED**: Lockup is enabled

6.15.16 Power mode control 0 register

SCU_PMCON0 Offset address: 0040_H

Power mode control 0 register RESET_TYPE_3 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												SD	PD	SL	XTAL_ON
r												rwpw	rwpw	rwpw	rw

Field	Bits	Type	Description
XTAL_ON	0	rw	**OSC_HP operation in STOP mode** This provides user the option for reduced power consumption in the STOP mode. It must be noted that the startup time of OSC_HP can be in the range of some milliseconds. Alternatively for fast wake-up from STOP mode while avoiding this power consumption, the user can selectively enable internal oscillator as clock source and disable OSC_HP before enable STOP mode. 0_B **XTAL_OFF**: OSC_HP (XTAL) will be suspended by hardware in STOP mode 1_B **XTAL_ON**: OSC_HP (XTAL) continues to operate in STOP mode, if enabled by SCU_OSC_CON.XPD
SL	1	rwpw	**Sleep mode** Setting this bit will cause the chip to go into sleep mode. Reset by wake-up circuit. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **INACTIVE**: No change 1_B **ACTIVE**: Device goes into Sleep mode
PD	2	rwpw	**Power-down mode** Setting this bit will cause the chip to go into a power down mode. Reset by wake-up circuit. *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* 0_B **INACTIVE**: No change 1_B **ACTIVE**: Device goes into Power-down mode
SD	3	rwpw	**Slow-down mode** Setting this bit will cause the chip to go into slow down mode. Reset by user.

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly. 0_B **INACTIVE**: No change 1_B **ACTIVE**: Device goes into Slow-down mode
RES	31:4	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.17 NMI status clear register

SCU_NMISRCLR Offset address: 0000_H

NMI status clear register RESET_TYPE_3 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								FNMISUPC	FNMIECCC	FNMIMAPC	FNMIOWDC	FNMIOTC	FNMINVMC	FNMIPLLC	RES
r								w	w	w	w	w	w	w	r

Field	Bits	Type	Description
RES	0, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
FNMIPLLC	1	w	**PLL NMI flag** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
FNMINVMC	2	w	**NVM operation complete NMI flag** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
FNMIOTC	3	w	**Overtemperature 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 **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
FNMIOWDC	4	w	**Oscillator watchdog NMI flag** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
FNMIMAPC	5	w	**NVM map error NMI flag** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
FNMIECCC	6	w	**ECC error NMI flag** This flag is cleared automatically by hardware when the corresponding enabled event flags are cleared. 0_B **NOT_CLEARED**: Interrupt event is not cleared

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **CLEARED**: Interrupt event is cleared
FNMISUPC	7	w	**Supply prewarning NMI flag** This flag is cleared automatically by hardware when the corresponding event flags are cleared. *Note: This flag has no effect as it is an logical OR of all supply flags and is automatically cleared when the sources are cleared.* 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared

6.15.18 Interrupt request 0 register

SCU_IRCON0 Offset address: 0004_H

Interrupt request 0 register RESET_TYPE_3 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										**EXINT2F**	**EXINT2R**	**EXINT1F**	**EXINT1R**	**EXINT0F**	**EXINT0R**
r										r	r	r	r	r	r

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 **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: 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 **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: 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 **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: 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 **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: 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 **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: 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 **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: Interrupt on falling edge event has occurred
RES	31:6	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.19 Interrupt request 1 register

SCU_IRCON1 Offset address: 0008_H

Interrupt request 1 register RESET_TYPE_3 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						MON 5F	MON 5R	MON 4F	MON 4R	MON 3F	MON 3R	MON 2F	MON 2R	MON 1F	MON 1R
r						r	r	r	r	r	r	r	r	r	r

Field	Bits	Type	Description
MON1R	0	r	**Interrupt flag for MON1x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: Interrupt on rising edge event has occurred
MON1F	1	r	**Interrupt flag for MON1x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: Interrupt on falling edge event has occurred
MON2R	2	r	**Interrupt flag for MON2x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: Interrupt on rising edge event has occurred
MON2F	3	r	**Interrupt flag for MON2x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: Interrupt on falling edge event has occurred
MON3R	4	r	**Interrupt flag for MON3x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: Interrupt on rising edge event has occurred
MON3F	5	r	**Interrupt flag for MON3x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: Interrupt on falling edge event has occurred
MON4R	6	r	**Interrupt flag for MON4x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: Interrupt on rising edge event has occurred

(table continues...)

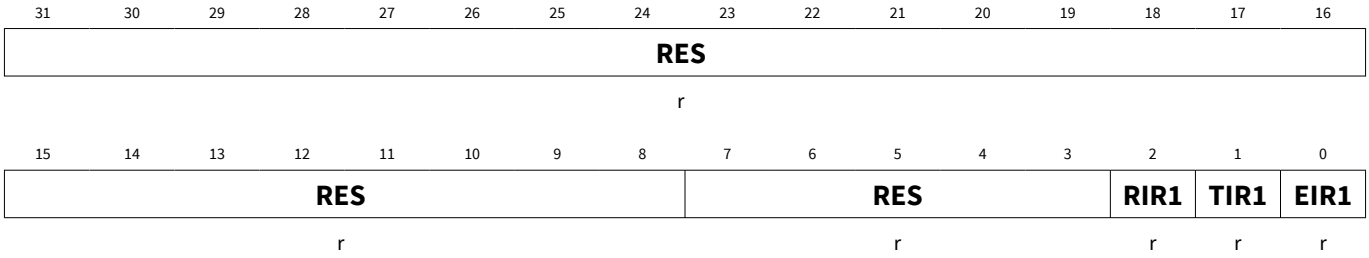
(continued)

Field	Bits	Type	Description
MON4F	7	r	**Interrupt flag for MON4x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: Interrupt on falling edge event has occurred
MON5R	8	r	**Interrupt flag for MON5x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on rising edge event has not occurred 1_B **OCCURED**: Interrupt on rising edge event has occurred
MON5F	9	r	**Interrupt flag for MON5x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt on falling edge event has not occurred 1_B **OCCURED**: Interrupt on falling edge event has occurred
RES	31:10	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.20 Interrupt request 2 register

SCU_IRCON2 Offset address: $000C_H$

Interrupt request 2 register RESET_TYPE_3 value: $0000\ 0000_H$

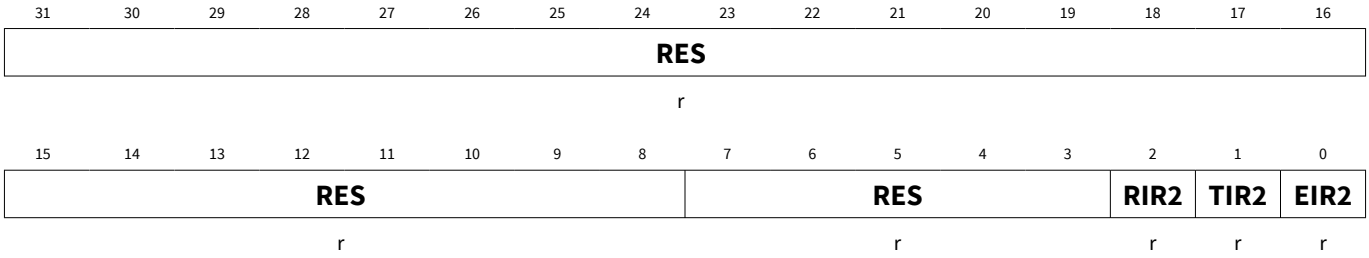


Field	Bits	Type	Description
EIR1	0	r	**Error interrupt flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
TIR1	1	r	**Transmit interrupt flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
RIR1	2	r	**Receive interrupt flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
RES	7:3, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.21 Interrupt request 3 register

SCU_IRCON3 Offset address: 0010_H

Interrupt request 3 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
EIR2	0	r	**Error interrupt flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
TIR2	1	r	**Transmit interrupt flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
RIR2	2	r	**Receive interrupt flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
RES	7:3, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.22 Interrupt request 4 register

SCU_IRCON4 Offset address: 0014_H

Interrupt request 4 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											CCU6 SR3	RES			CCU6 SR2
r											r	r			r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											CCU6 SR1	RES			CCU6 SR0
r											r	r			r

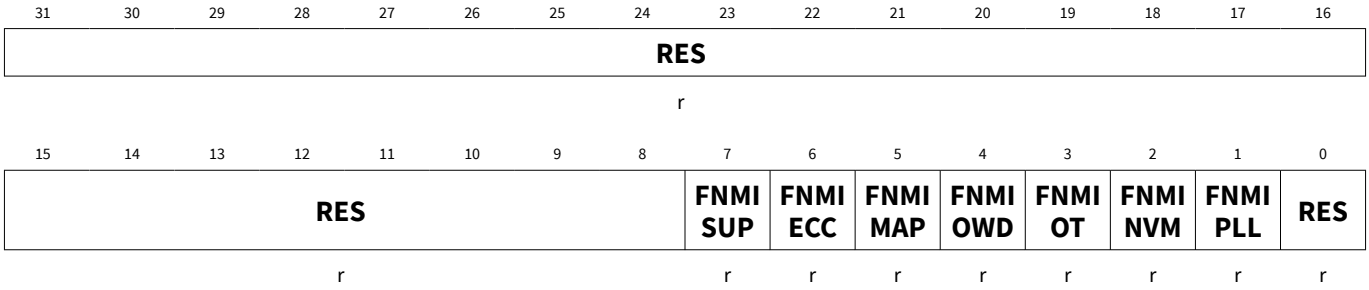
Field	Bits	Type	Description
CCU6SR0	0	r	**Interrupt flag 0 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
RES	3:1, 15:5, 19:17, 31:21	r	**Reserved** Returns 0 if read; should 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 **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
CCU6SR2	16	r	**Interrupt flag 2 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
CCU6SR3	20	r	**Interrupt flag 3 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred

6.15.23 NMI status register

Each NMI event and status flag is retained across soft reset. This includes all the flags of NMISR register: FNMIPLL, FNMINVM, FNMIOCDS, FNMIOWD, FNMIMAP, and indirectly, FNMIECC and FNMISUP. In the case of NMIs with shared source that is watchdog, ECC or supply prewarning NMI, the respective indicator or event flags not located in NMISR are also retained.

SCU_NMISR Offset address: 0018_H

NMI status register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
RES	0, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
FNMIPLL	1	r	**PLL NMI flag** This bit is set by hardware and can only be cleared by software. 0_B **NO_PLL**: No PLL NMI has occurred 1_B **PLL**: PLL loss-of-lock to the external crystal 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**: No NVM NMI has occurred 1_B **NVM**: NVM operation complete event has occurred
FNMIOT	3	r	**Overtemperature 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**: No OT NMI has occurred 1_B **OT**: OT NMI event has occurred
FNMIOWD	4	r	**Oscillator watchdog NMI flag** This bit is set by hardware and can only be cleared by software. 0_B **NO_OSCILLATOR**: No oscillator watchdog NMI has occurred 1_B **OSCILLATOR**: Oscillator watchdog event has occurred
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**: No NVM map error NMI has occurred 1_B **NVM**: NVM map error has occurred
FNMIECC	6	r	**ECC error NMI flag**

(table continues...)

(continued)

Field	Bits	Type	Description
			This flag is cleared automatically by hardware when the corresponding enabled event flags are cleared. 0_B **NO_ECC**: No uncorrectable ECC error has occurred on NVM, XRAM 1_B **ECC**: Uncorrectable ECC error has occurred on NVM, RAM
FNMISUP	7	r	**Supply prewarning NMI flag** This flag is cleared automatically by hardware when the corresponding event flags are cleared. 0_B **NO_PREWARN**: No supply prewarning NMI has occurred 1_B **PREWARN**: Supply prewarning has occurred

6.15.24 Interrupt enable 0 register

SCU_IEN0 Offset address: $001C_H$

Interrupt enable 0 register RESET_TYPE_4 value: $0000\ 0000_H$

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

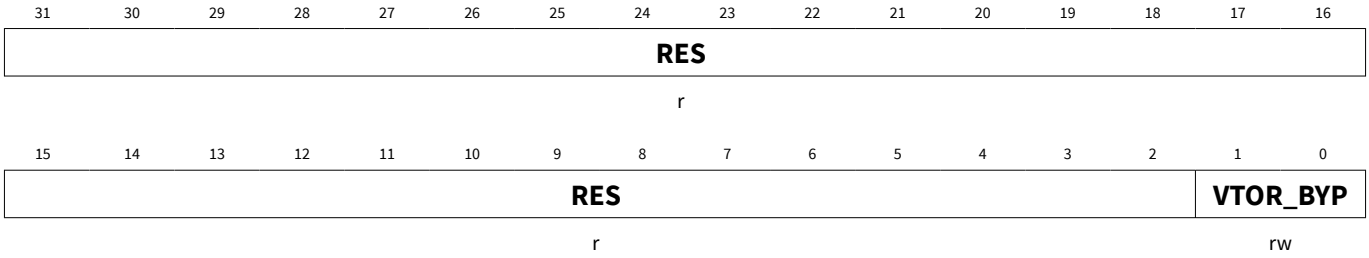
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES															
r															

Field	Bits	Type	Description
RES	23:0, 30:24	r	**Reserved** Returns 0 if read; should be written with 0.
EA	31	rw	**Global interrupt mask** 0_B **BLOCKED**: All pending interrupt requests (except NMI) are blocked from the core 1_B **NOT_BLOCKED**: Pending interrupt requests are not blocked from the core

6.15.25 Vector table reallocation register

SCU_VTOR Offset address: 0020_H

Vector table reallocation register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
VTOR_BYP	1:0	rw	**Vector table bypass mode** 00_B **NOT_REMAPPED**: VTOR is not remapped (ROM) (start address: 0x0000000000) 01_B **RAM**: VTOR is remapped to RAM (start address: 0x1800000000) 10_B **BSL**: VTOR is remapped to NVM (start address: 0x1100000000, begin of customer BSL region) 11_B **NVM**: VTOR is remapped to NVM (start address: begin of NVM linear region after customer BSL region)
RES	31:2	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.26 NMI control register

SCU_NMICON Offset address: 0024_H

NMI control register RESET_TYPE_3 value: $0000\ 0000_H$

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

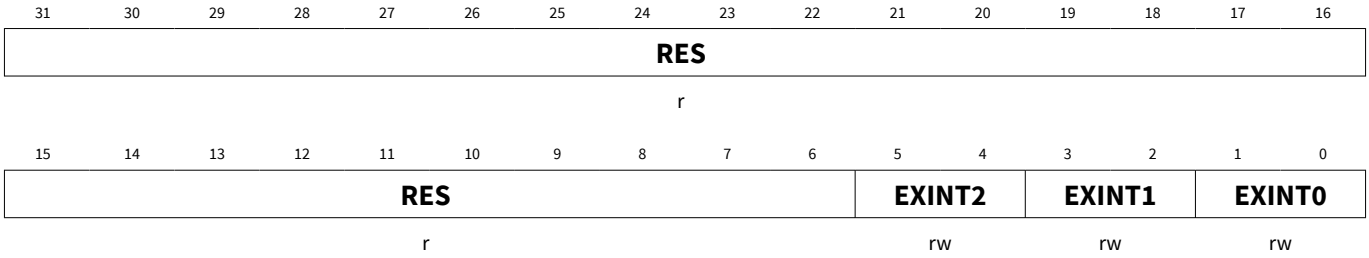
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								NMISUP	NMIECC	NMIMAP	NMIOWD	NMIOT	NMINVM	NMIPLL	RES
r								rw	rw	rw	rw	rw	rw	rw	r

Field	Bits	Type	Description
RES	0, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
NMIPLL	1	rw	**PLL loss of lock NMI enable** 0_B **DISABLED**: PLL loss of lock NMI is disabled 1_B **ENABLED**: PLL loss of lock NMI is enabled
NMINVM	2	rw	**NVM operation complete NMI enable** 0_B **DISABLED**: NVM operation complete NMI is disabled 1_B **ENABLED**: NVM operation complete NMI is enabled
NMIOT	3	rw	**NMI OT enable** 0_B **DISABLED**: NMI OT is disabled 1_B **ENABLED**: NMI OT is enabled
NMIOWD	4	rw	**Oscillator watchdog NMI enable** 0_B **DISABLED**: Oscillator watchdog NMI is disabled 1_B **ENABLED**: Oscillator watchdog NMI is enabled
NMIMAP	5	rw	**NVM map error NMI enable** 0_B **DISABLED**: NVM map error NMI is disabled 1_B **ENABLED**: NVM map error NMI is enabled
NMIECC	6	rw	**ECC error NMI enable** 0_B **DISABLED**: ECC Error NMI is disabled 1_B **ENABLED**: ECC Error NMI is enabled
NMISUP	7	rw	**Supply prewarning NMI enable** 0_B **DISABLED**: Supply NMI is disabled 1_B **ENABLED**: Supply NMI is enabled

6.15.27 External interrupt control 0 register

SCU_EXICON0 Offset address: 0028_H

External interrupt control 0 register RESET_TYPE_3 value: $0000\ 0030_H$

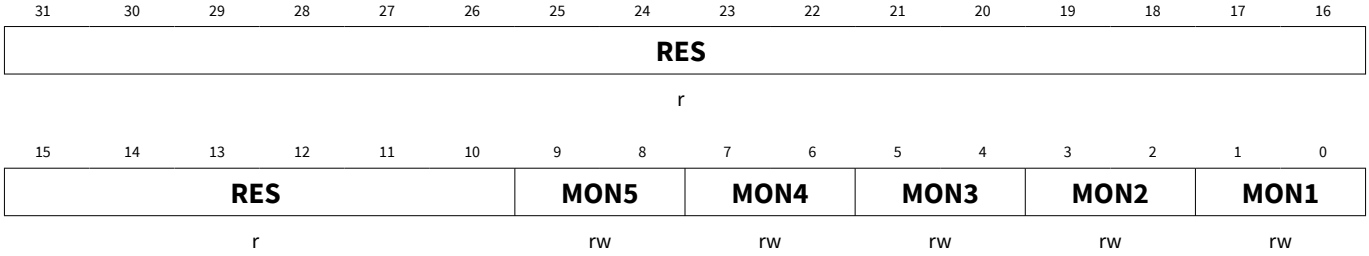


Field	Bits	Type	Description
EXINT0	1:0	rw	**External interrupt 0 trigger select** 00_B **DISABLED**: Interrupt disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
EXINT1	3:2	rw	**External interrupt 1 trigger select** 00_B **DISABLED**: Interrupt disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
EXINT2	5:4	rw	**External interrupt 2 trigger select** 00_B **DISABLED**: Interrupt disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
RES	31:6	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.28 External interrupt control 1 register

SCU_EXICON1 Offset address: $002C_H$

External interrupt control 1 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
MON1	1:0	rw	**MON1 input trigger select** 00_B **DISABLED**: External interrupt MON is disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
MON2	3:2	rw	**MON2 input trigger select** 00_B **DISABLED**: External interrupt MON is disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
MON3	5:4	rw	**MON3 input trigger select** 00_B **DISABLED**: External interrupt MON is disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
MON4	7:6	rw	**MON4 input trigger select** 00_B **DISABLED**: External interrupt MON is disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
MON5	9:8	rw	**MON5 input trigger select** 00_B **DISABLED**: External interrupt MON is disabled 01_B **RISING**: Interrupt on rising edge 10_B **FALLING**: Interrupt on falling edge 11_B **BOTH**: Interrupt on both rising and falling edge
RES	31:10	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.29 Peripheral interrupt enable 1 register

SCU_MODIEN1 Offset address: 0030_H

Peripheral interrupt enable 1 register RESET_TYPE_3 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					RIREN2	TIREN2	EIREN2	RES					RIREN1	TIREN1	EIREN1
r					rw	rw	rw	r					rw	rw	rw

Field	Bits	Type	Description
EIREN1	0	rw	**SSC 1 error interrupt enable** 0_B **DISABLED**: Error interrupt is disabled 1_B **ENABLED**: Error interrupt is enabled
TIREN1	1	rw	**SSC 1 transmit interrupt enable** 0_B **DISABLED**: Transmit interrupt is disabled 1_B **ENABLED**: Transmit interrupt is enabled
RIREN1	2	rw	**SSC 1 receive interrupt enable** 0_B **DISABLED**: Receive interrupt is disabled 1_B **ENABLED**: Receive interrupt is enabled
RES	7:3, 31:11	r	**Reserved** Returns 0 if read; should be written with 0.
EIREN2	8	rw	**SSC 2 error interrupt enable** 0_B **DISABLED**: Error interrupt is disabled 1_B **ENABLED**: Error interrupt is enabled
TIREN2	9	rw	**SSC 2 transmit interrupt enable** 0_B **DISABLED**: Transmit interrupt is disabled 1_B **ENABLED**: Transmit interrupt is enabled
RIREN2	10	rw	**SSC 2 receive interrupt enable** 0_B **DISABLED**: Receive interrupt is disabled 1_B **ENABLED**: Receive interrupt is enabled

6.15.30 Peripheral interrupt enable 2 register

SCU_MODIEN2 Offset address: 0034_H

Peripheral interrupt enable 2 register RESET_TYPE_3 value: $0000\ 0000_H$

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

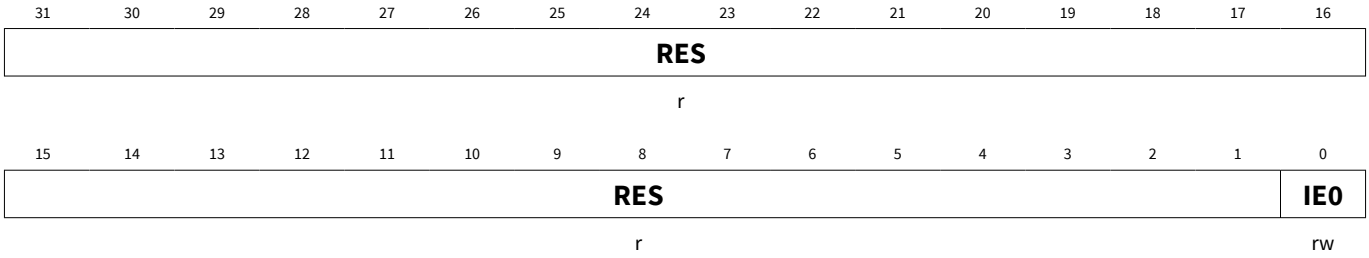
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								TIEN2	RIEN2	EXINT2_EN	RES			TIEN1	RIEN1
r								rw	rw	rw	r			rw	rw

Field	Bits	Type	Description
RIEN1	0	rw	**UART 1 receive interrupt enable** 0_B **DISABLED**: Receive interrupt is disabled 1_B **ENABLED**: Receive interrupt is enabled
TIEN1	1	rw	**UART 1 transmit interrupt enable** 0_B **DISABLED**: Transmit interrupt is disabled 1_B **ENABLED**: Transmit interrupt is enabled
RES	4:2, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
EXINT2_EN	5	rw	**External interrupt 2 enable** 0_B **DISABLED**: External interrupt is disabled 1_B **ENABLED**: External interrupt is enabled
RIEN2	6	rw	**UART 2 receive interrupt enable** 0_B **DISABLED**: Receive interrupt is disabled 1_B **ENABLED**: Receive interrupt is enabled
TIEN2	7	rw	**UART 2 transmit interrupt enable** 0_B **DISABLED**: Transmit interrupt is disabled 1_B **ENABLED**: Transmit interrupt is enabled

6.15.31 Peripheral interrupt enable 3 register

SCU_MODIEN3 Offset address: 0038_H

Peripheral interrupt enable 3 register RESET_TYPE_3 value: $0000\ 0000_H$

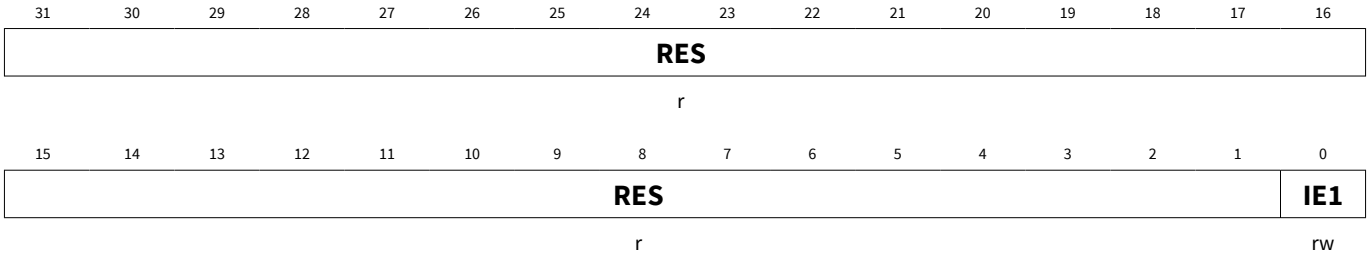


Field	Bits	Type	Description
IE0	0	rw	**External interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
RES	31:1	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.32 Peripheral interrupt enable 4 register

SCU_MODIEN4 Offset address: $003C_H$

Peripheral interrupt enable 4 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
IE1	0	rw	**External interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
RES	31:1	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.33 Wake-up interrupt control register

SCU_WAKECON Offset address: 0078_H

Wake-up interrupt control register RESET_TYPE_3 value: $0000\ 0000_H$

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

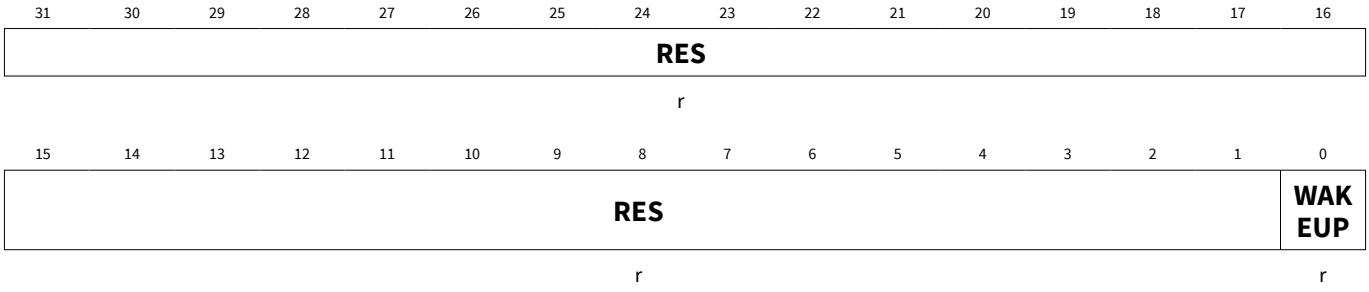
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES															**WAKEUPEN**
r															rw

Field	Bits	Type	Description
WAKEUPEN	0	rw	**Wake-up interrupt enable** 0_B **DISABLED**: Wake-up interrupt is disabled 1_B **ENABLED**: Wake-up interrupt is enabled
RES	31:1	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.34 Interrupt request 5 register

SCU_IRCON5 Offset address: $007C_H$

Interrupt request 5 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
WAKEUP	0	r	**Interrupt flag for wake-up** This bit is set by hardware and can only be cleared by software. 0_B **NOT_OCCURED**: Interrupt event has not occurred 1_B **OCCURED**: Interrupt event has occurred
RES	31:1	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.35 General purpose timer 12 interrupt enable register

SCU_GPT12IEN Offset address: $015C_H$

General purpose timer 12 interrupt enable register RESET_TYPE_3 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										**CRIE**	**T6IE**	**T5IE**	**T4IE**	**T3IE**	**T2IE**
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
T2IE	0	rw	**GPT12 T2 interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
T3IE	1	rw	**GPT12 T3 interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
T4IE	2	rw	**GPT12 T4 interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
T5IE	3	rw	**GPT12 T5 interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
T6IE	4	rw	**GPT12 T6 interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
CRIE	5	rw	**GPT12 capture and reload interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
RES	31:6	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.36 Timer and counter control/status register

SCU_GPT12IRC Offset address: 0160_H

Timer and counter control/status register RESET_TYPE_3 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		GPT12CR	GPT2T6	GPT2T5	GPT1T4	GPT1T3	GPT1T2
r								r		r	r	r	r	r	r

Field	Bits	Type	Description
GPT1T2	0	r	**GPT module 1 Timer2 interrupt status** Timer2 of GPT1 module interrupt status. 0_B **NOT_OCCURRED**: No Timer2 interrupt has occurred 1_B **OCCURRED**: Timer2 interrupt has occurred
GPT1T3	1	r	**GPT module 1 Timer3 interrupt status** Timer3 of GPT1 module interrupt status. 0_B **NOT_OCCURRED**: No Timer3 interrupt has occurred 1_B **OCCURRED**: Timer3 interrupt has occurred
GPT1T4	2	r	**GPT module 1 Timer4 interrupt status** Timer4 of GPT1 module interrupt status. 0_B **NOT_OCCURRED**: No Timer4 interrupt has occurred 1_B **OCCURRED**: Timer4 interrupt has occurred
GPT2T5	3	r	**GPT module 2 Timer5 interrupt status** Timer5 of GPT2 module interrupt status. 0_B **NOT_OCCURRED**: No Timer5 interrupt has occurred 1_B **OCCURRED**: Timer5 interrupt has occurred
GPT2T6	4	r	**GPT module 2 Timer6 interrupt status** Timer6 of GPT module interrupt status. 0_B **NOT_OCCURRED**: No Timer6 interrupt has occurred 1_B **OCCURRED**: Timer6 interrupt has occurred
GPT12CR	5	r	**GPT12 capture reload interrupt status** Capture reload event of GPT1 module interrupt status. 0_B **NOT_OCCURRED**: No capture reload interrupt has occurred 1_B **OCCURRED**: Capture reload interrupt has occurred
RES	7:6, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.37 Interrupt request 0 clear register

SCU_IRCON0CLR Offset address: 0178_H

Interrupt request 0 clear register RESET_TYPE_3 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										EXINT2FC	EXINT2RC	EXINT1FC	EXINT1RC	EXINT0FC	EXINT0RC
r										w	w	w	w	w	w

Field	Bits	Type	Description
EXINT0RC	0	w	**Interrupt flag for external interrupt 0x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
EXINT0FC	1	w	**Interrupt flag for external interrupt 0x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
EXINT1RC	2	w	**Interrupt flag for external interrupt 1x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
EXINT1FC	3	w	**Interrupt flag for external interrupt 1x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
EXINT2RC	4	w	**Interrupt flag for external interrupt 2x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
EXINT2FC	5	w	**Interrupt flag for external interrupt 2x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	31:6	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.38 Interrupt request 1 clear register

SCU_IRCON1CLR Offset address: $017C_H$

Interrupt request 1 clear register RESET_TYPE_3 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						**MON 5FC**	**MON 5RC**	**MON 4FC**	**MON 4RC**	**MON 3FC**	**MON 3RC**	**MON 2FC**	**MON 2RC**	**MON 1FC**	**MON 1RC**
r						w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
MON1RC	0	w	**Interrupt flag for MON1x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON1FC	1	w	**Interrupt flag for MON1x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON2RC	2	w	**Interrupt flag for MON2x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON2FC	3	w	**Interrupt flag for MON2x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON3RC	4	w	**Interrupt flag for MON3x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON3FC	5	w	**Interrupt flag for MON3x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON4RC	6	w	**Interrupt flag for MON4x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared

(table continues...)

(continued)

Field	Bits	Type	Description
MON4FC	7	w	**Interrupt flag for MON4x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON5RC	8	w	**Interrupt flag for MON5x on rising edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
MON5FC	9	w	**Interrupt flag for MON5x on falling edge** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	31:10	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.39 Timer and counter control/status clear register

SCU_GPT12ICLR Offset address: 0180_H

Timer and counter control/status clear register RESET_TYPE_3 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		GPT12CRC	GPT2T6C	GPT2T5C	GPT1T4C	GPT1T3C	GPT1T2C
r								r		w	w	w	w	w	w

Field	Bits	Type	Description
GPT1T2C	0	w	**GPT module 1 Timer2 interrupt status** Timer2 of GPT1 module interrupt status. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
GPT1T3C	1	w	**GPT module 1 Timer3 interrupt status** Timer3 of GPT1 module interrupt status. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
GPT1T4C	2	w	**GPT module 1 Timer4 interrupt status** Timer4 of GPT1 module interrupt status. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
GPT2T5C	3	w	**GPT module 2 Timer5 interrupt status** Timer5 of GPT2 module interrupt status. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
GPT2T6C	4	w	**GPT module 2 Timer6 interrupt status** Timer6 of GPT module interrupt status. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
GPT12CRC	5	w	**GPT12 capture reload interrupt status** Capture reload event of GPT1 module interrupt status. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	7:6, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.40 Monitoring input interrupt enable register

SCU_MONIEN Offset address: $018C_H$

Monitoring input interrupt enable register RESET_TYPE_3 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											MON 5IE	MON 4IE	MON 3IE	MON 2IE	MON 1IE
r											rw	rw	rw	rw	rw

Field	Bits	Type	Description
MON1IE	0	rw	**MON1 interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
MON2IE	1	rw	**MON2 interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
MON3IE	2	rw	**MON3 interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
MON4IE	3	rw	**MON4 interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
MON5IE	4	rw	**MON5 interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
RES	31:5	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.41 Interrupt request 2 clear register

SCU_IRCON2CLR Offset address: 0190_H

Interrupt request 2 clear register RESET_TYPE_3 value: $0000\ 0000_H$

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

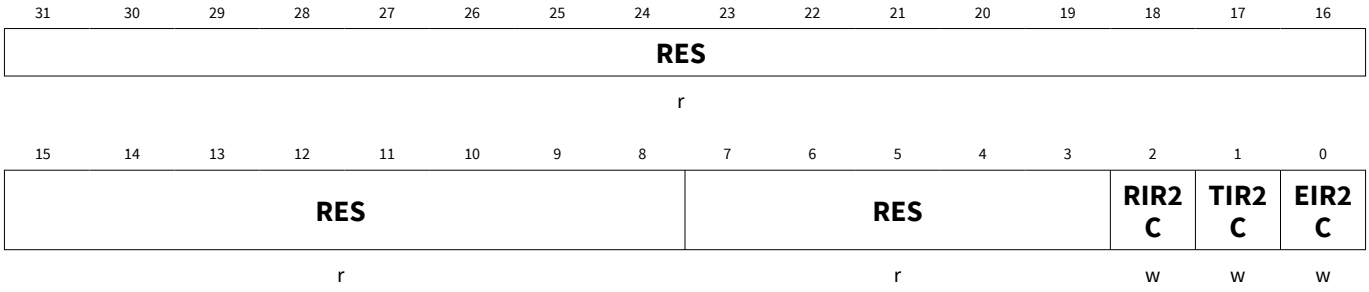
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								**RES**					**RIR1C**	**TIR1C**	**EIR1C**
r								r					w	w	w

Field	Bits	Type	Description
EIR1C	0	w	**Error interrupt flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
TIR1C	1	w	**Transmit interrupt flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RIR1C	2	w	**Receive interrupt flag for SSC1** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	7:3, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.42 Interrupt request 3 clear register

SCU_IRCON3CLR Offset address: 0194_H

Interrupt request 3 clear register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
EIR2C	0	w	**Error interrupt flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
TIR2C	1	w	**Transmit interrupt flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RIR2C	2	w	**Receive interrupt flag for SSC2** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	7:3, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.43 Interrupt request 4 clear register

SCU_IRCON4CLR Offset address: 0198_H

Interrupt request 4 clear register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											CCU6 SR3C	RES			CCU6 SR2C
r											w	r			w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											CCU6 SR1C	RES			CCU6 SR0C
r											w	r			w

Field	Bits	Type	Description
CCU6SR0C	0	w	**Interrupt flag 0 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	3:1, 15:5, 19:17, 31:21	r	**Reserved** Returns 0 if read; should be written with 0.
CCU6SR1C	4	w	**Interrupt flag 1 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
CCU6SR2C	16	w	**Interrupt flag 2 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
CCU6SR3C	20	w	**Interrupt flag 3 for CCU6** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared

6.15.44 Interrupt request 5 clear register

SCU_IRCON5CLR Offset address: $019C_H$

Interrupt request 5 clear register RESET_TYPE_3 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															**WAKEUPC**
r															w

Field	Bits	Type	Description
WAKEUPC	0	w	**Clear flag for wake-up interrupt** This bit is set by hardware and can only be cleared by software. 0_B **NOT_CLEARED**: Interrupt event is not cleared 1_B **CLEARED**: Interrupt event is cleared
RES	31:1	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.45 Peripheral input select register

SCU_MODPISEL Offset address: $00B8_H$

Peripheral input select register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													SSC12_S_MRST_OUTSEL	SSC12_M_MTSR_OUTSEL	SSC12_M_SCK_OUTSEL
r													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								U_TX_CONDIS	URIOS1	EXINT2IS		EXINT1IS		EXINT0IS	
r								rw	rw	rw		rw		rw	

Field	Bits	Type	Description
EXINT0IS	1:0	rw	**External interrupt 0 input select** 00_B **EXINT0_0**: External interrupt input EXINT0_0 is selected 01_B **EXINT0_1**: External interrupt input EXINT0_1 is selected 10_B **EXINT0_2**: External interrupt input EXINT0_2 is selected 11_B **EXINT0_3**: External interrupt input EXINT0_3 is selected
EXINT1IS	3:2	rw	**External interrupt 1 input select** 00_B **EXINT1_0**: External interrupt input EXINT1_0 is selected 01_B **EXINT1_1**: External interrupt input EXINT1_1 is selected 10_B **EXINT1_2**: External interrupt input EXINT1_2 is selected 11_B **EXINT1_3**: External interrupt input EXINT1_3 is selected
EXINT2IS	5:4	rw	**External interrupt 2 input select** 00_B **EXINT2_0**: External interrupt input EXINT2_0 is selected 01_B **EXINT2_1**: External interrupt input EXINT2_1 is selected 10_B **EXINT2_2**: External interrupt input EXINT2_2 is selected 11_B **EXINT2_3**: External interrupt input EXINT2_3 is selected
URIOS1	6	rw	**UART1 input select** *Note: For more details, please refer to the figure "Interconnect TRX, UART1, TIMER2, GPIO, CCU6, SCU, PMU". The port ALTSELx registers need to be configured additionally.* 0_B **TRX**: UART1 receiver input UART1_RXD (connected to transceiver) 1_B **GPIO**: UART1 receiver input UART1_RXD (connected to GPIO)
U_TX_CONDIS	7	rw	**TRX input select** *Note: For more details, please refer to the figure "Interconnect TRX, UART1, TIMER2, GPIO, CCU6, SCU, PMU".*

(table continues...)

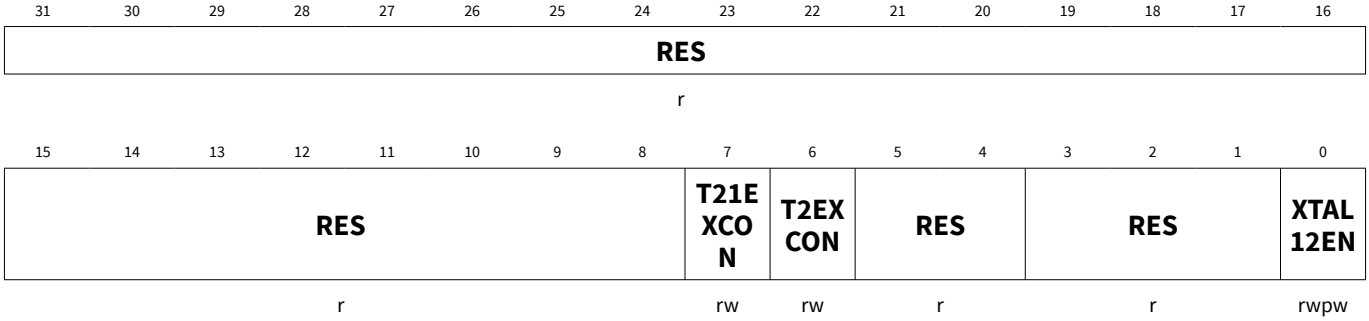
(continued)

Field	Bits	Type	Description
			0_B **UART**: Transceiver TXD input connected to UART1_TXD output 1_B **GPIO**: Transceiver TXD input connected to GPIO
RES	15:8, 31:19	r	**Reserved** Returns 0 if read; should be written with 0.
SSC12_M_SCK_OUTSEL	16	rw	**Output selection for SSC12_M_SCK** See GPIO ports and peripheral I/O, port implementation details. 0_B **SSC1**: SSC1_M_SCK 1_B **SSC2**: SSC2_M_SCK
SSC12_M_MTSR_OUTSEL	17	rw	**Output selection for SSC12_M_MTSR** See GPIO ports and peripheral I/O, port implementation details. 0_B **SSC1**: SSC1_M_MTSR 1_B **SSC2**: SSC2_M_MTSR
SSC12_S_MRST_OUTSEL	18	rw	**Output selection for SSC12_S_MRST** See GPIO ports and peripheral I/O, port implementation details. 0_B **SSC1**: SSC1_S_MRST 1_B **SSC2**: SSC2_S_MRST

6.15.46 Peripheral input select 1 register

SCU_MODPISEL1 Offset address: $00BC_H$

Peripheral input select 1 register Reset values see: Table 54



Field	Bits	Type	Description
XTAL12EN	0	rwpw	**Pins XTAL1/2 enable bit** *Note: This is a PASSWD protected bit. When the protection scheme is activated (default), this bit cannot be written directly.* *Note: This bit is RESET_TYPE_4.* 0_B **NOT_AVAILABLE**: Pins XTAL1/2 is not available. This setting overrides the OSC_CON.XPD setting 1_B **AVAILABLE**: Pins XTAL1/2 is available
RES	3:1, 5:4, 31:8	r	**Reserved** Returns 0 if read; should be written with 0.
T2EXCON	6	rw	**Timer 2 external input control** 0_B **SELECT**: Timer2 input T2EX is selected by bit field SCU_MODPISEL2.T2EXIS 1_B **CONNECT**: Timer2 input T2EX is connected to signal from CCU6 (Output>cc6_cout60)
T21EXCON	7	rw	**Timer 21 external input control** 0_B **SELECT**: Timer21 input T21EX is selected by bit field SCU_MODPISEL2.T21EXIS 1_B **CONNECT**: Timer21 input T21EX is connected to signal from CCU6 (Output >cc6_ch0)

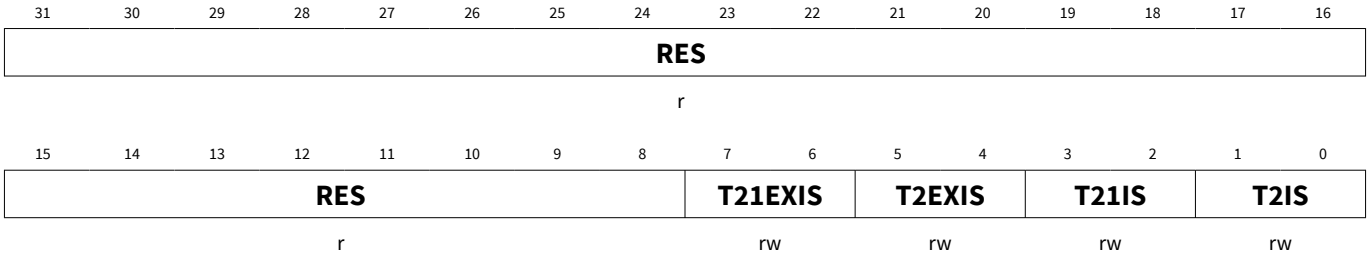
Table 54 Reset values of **SCU_MODPISEL1**

Reset type	Reset value	Note
RESET_TYPE_3	$0000\ 0000_H$	ResetMask= “0b00000000000000000000000000000001”
RESET_TYPE_4	$0000\ 0000_H$	ResetMask= “0b11111111111111111111111111111110”

6.15.47 Peripheral input select 2 register

SCU_MODPISEL2 Offset address: $00C0_H$

Peripheral input select 2 register RESET_TYPE_3 value: $0000\ 0000_H$

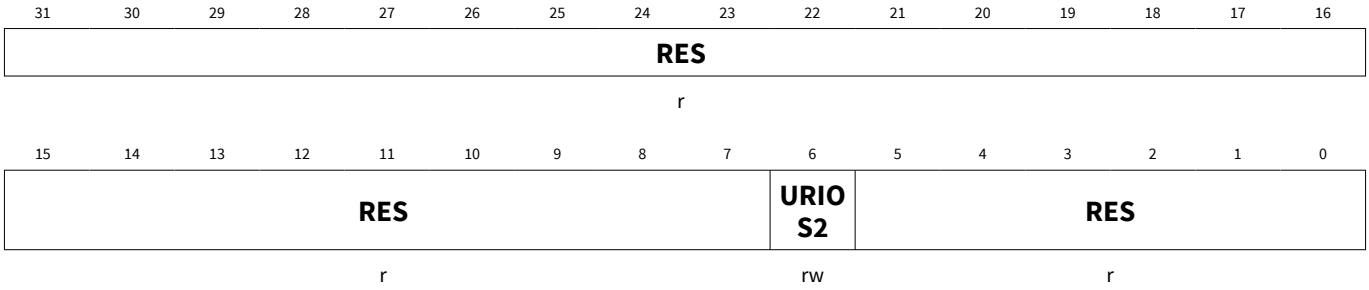


Field	Bits	Type	Description
T2IS	1:0	rw	**Timer 2 input select** 00_B **T2_0**: Timer2 input T2_0 is selected 01_B **T2_1**: Timer2 input T2_1 is selected 10_B **T2_2**: Timer2 input T2_2 is selected 11_B **RES**: Reserved
T21IS	3:2	rw	**Timer 21 input select** 00_B **T21_0**: Timer21 input T21_0 is selected 01_B **T21_1**: Timer21 input T21_1 is selected 10_B **T21_2**: Timer21 input T21_2 is selected 11_B **RES**: Reserved
T2EXIS	5:4	rw	**Timer 2 external input select** *Note: This selection takes effect only when SCU_MODPISEL1.T2EXCON = 0.* 00_B **T2EX_0**: Timer2 input T2EX_0 is selected 01_B **T2EX_1**: Timer2 input T2EX_1 is selected 10_B **T2EX_2**: Timer2 input T2EX_2 i selected 11_B **T2EX_3**: Timer2 input T2EX_3 is selected
T21EXIS	7:6	rw	**Timer 21 external input select** *Note: This selection takes effect only when SCU_MODPISEL1.T21EXCON = 0.* 00_B **T21EX_0**: Timer21 input T21EX_0 is selected 01_B **T21EX_1**: Timer21 input T21EX_1 is selected 10_B **T21EX_2**: Timer21 input T21EX_2 is selected 11_B **T21EX_3**: Timer21 input T21EX_3 is selected
RES	31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.48 Peripheral input select 3 register

SCU_MODPISEL3 Offset address: $00C4_H$

Peripheral input select 3 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
RES	5:0, 31:7	r	**Reserved** Returns 0 if read; should be written with 0.
URIOS2	6	rw	**UART2 input select** *Note: Port ALTSELx registers need to be configured additionally.* 0_B **GPIO**: UART2 receiver input UART2_RXD (connected to GPIO) 1_B **GPIO**: UART2 receiver input UART2_RXD (connected to GPIO)

6.15.49 GPT12 peripheral input select register

SCU_GPT12PISEL Offset address: $00D0_H$

GPT12 peripheral input select register RESET_TYPE_3 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										GPT12_SEL	TRIG_CONF	GPT12			
r										rw	rw	rw			

Field	Bits	Type	Description
GPT12	3:0	rw	**GPT12 T3INB/T4IND input select** 0_H **CC60**: CC60 1_H **CC61**: CC61 2_H **CC62**: CC62 3_H **T12_ZM**: T12 ZM 4_H **T12_PM**: T12 PM 5_H **T12_CM0**: T12 CM0 6_H **T12_CM1**: T12 CM1 7_H **T12_CM2**: T12 CM2 8_H **T13_PM**: T13 PM 9_H **T13_ZM**: T13 ZM A_H **T13_CM**: T13 CM B_H **ANY**: Any pos or neg edge on CC60/61/62 C_H **RES**: Reserved … F_H **RES**: Reserved
TRIG_CONF	4	rw	**CCU6 trigger Configuration** 0_B **ONE**: Trigger is just for one measurement (default) 1_B **NEXT**: Trigger is present until next input edge (selected by GPT12) – continuous measurement
GPT12_SEL	5	rw	**CCU6 trigger configuration** 0_B **T21**: CCU6_INT is triggered by Timer21 1_B **GPT12**: CCU6_INT is triggered by GPT12PISEL.GPT12
RES	31:6	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.50 Port output control register

SCU_P0_POCON0 Offset address: $00E8_H$

Port output control register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES									P0_PDM5			RES	P0_PDM4		
r									rw			r	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	P0_PDM3			RES	P0_PDM2			RES	P0_PDM1			RES	P0_PDM0		
r	rw			r	rw			r	rw			r	rw		

Field	Bits	Type	Description
P0_PDM0	2:0	rw	**P0.0 port driver mode** Code driver strength[1)] and edge shape[2)] 000_B **MEDIUM**: Medium driver 001_B **NU**: Not used 010_B **NU**: Not used 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
RES	3, 7, 11, 15, 19, 31:23	r	**Reserved** Returns 0 if read; should be written with 0.
P0_PDM1	6:4	rw	**P0.1 port driver mode** Code driver strength[1)] and edge shape[2)] 000_B **MEDIUM**: Medium driver 001_B **NU**: Not used 010_B **NU**: Not used 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver

(table continues...)

(continued)

Field	Bits	Type	Description
			111_B **WEAK**: Weak driver
P0_PDM2	10:8	rw	**P0.2 port driver mode** Code driver strength[1] and edge shape[2] 000_B **STRONG_SHARP**: Strong driver and sharp edge mode 001_B **STRONG_MEDIUM**: Strong driver and medium edge mode 010_B **STRONG_SOFT**: Strong driver and soft edge mode 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
P0_PDM3	14:12	rw	**P0.3 port driver mode** Code driver strength[1] and edge shape[2] 000_B **STRONG_SHARP**: Strong driver and sharp edge mode 001_B **STRONG_MEDIUM**: Strong driver and medium edge mode 010_B **STRONG_SOFT**: Strong driver and soft edge mode 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
P0_PDM4	18:16	rw	**P0.4 port driver mode** Code driver strength[1] and edge shape[2] 000_B **STRONG_SHARP**: Strong driver and sharp edge mode 001_B **STRONG_MEDIUM**: Strong driver and medium edge mode 010_B **STRONG_SOFT**: Strong driver and soft edge mode 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
P0_PDM5	22:20	rw	**P0.5 port driver mode** Code driver strength[1] and edge shape[2] 000_B **STRONG_SHARP**: Strong driver and sharp edge mode 001_B **STRONG_MEDIUM**: Strong driver and medium edge mode 010_B **STRONG_SOFT**: Strong driver and soft edge mode 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver

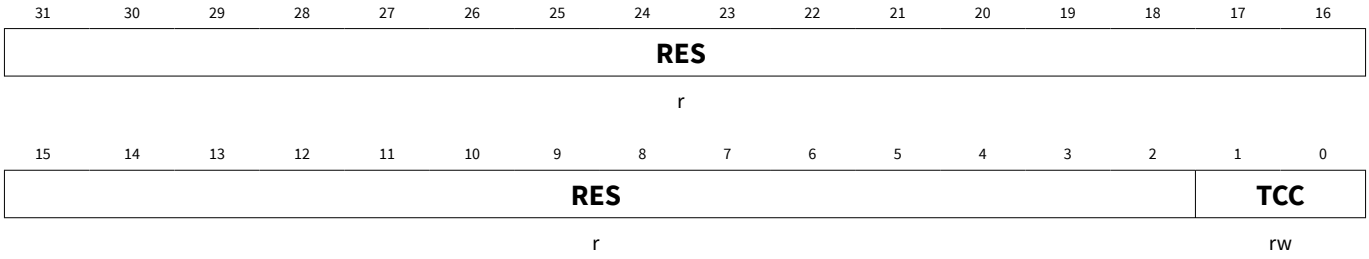
1) Defines the current the respective driver can deliver to the external circuitry.
2) Defines the switching characteristics to the respective new output driver. This also influences the peak currents through the driver when producing an edge, that is when changing the output level.

6.15.51 Temperature compensation control register

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

SCU_TCCR Offset address: $00F4_H$

Temperature compensation control register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
TCC	1:0	rw	**Temperature compensation control** The slew rate of the output driver is kept stable over the selected temperature range: 00_B **_40_0**: Tj: -40°C to 0°C 01_B **0_40**: Tj: 0°C to 40°C 10_B **40_80**: Tj: 40°C to 80°C 11_B **80_150**: Tj: 80°C to 150°C
RES	31:2	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.52 Port output control register

SCU_P1_POCON0 Offset address: $00F8_H$

Port output control register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													P1_PDM4		
r													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					P1_PDM2			RES	P1_PDM1			RES	P1_PDM0		
r					rw			r	rw			r	rw		

Field	Bits	Type	Description
P1_PDM0	2:0	rw	**P1.0 port driver mode** Code driver strength[1)] and edge shape[2)] 000_B **MEDIUM**: Medium driver 001_B **NU**: Not used 010_B **NU**: Not used 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
RES	3, 7, 15:11, 31:19	r	**Reserved** Returns 0 if read; should be written with 0.
P1_PDM1	6:4	rw	**P1.1 port driver mode** Code driver strength[1)] and edge shape[2)] 000_B **MEDIUM**: Medium driver 001_B **NU**: Not used 010_B **NU**: Not used 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
P1_PDM2	10:8	rw	**P1.2 port driver mode**

(table continues...)

(continued)

Field	Bits	Type	Description
			Code driver strength[1)] and edge shape[2)] 000_B **MEDIUM**: Medium driver 001_B **NU**: Not used 010_B **NU**: Not used 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver
P1_PDM4	18:16	rw	**P1.4 port driver mode** Code driver strength[1)] and edge shape[2)] 000_B **STRONG_SHARP**: Strong driver and sharp edge mode 001_B **STRONG_MEDIUM**: Strong driver and medium edge mode 010_B **STRONG_SOFT**: Strong driver and soft edge mode 011_B **WEAK**: Weak driver 100_B **MEDIUM**: Medium driver … 110_B **MEDIUM**: Medium driver 111_B **WEAK**: Weak driver

6.15.53 Peripheral input select 4 register

SCU_MODPISEL4 Offset address: $00FC_H$

Peripheral input select 4 register RESET_TYPE_3 value: $0403\ 0100_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES					**DU4TRIGGEN**			**RES**					**DU3TRIGGEN**		
r					rw			r					rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					**DU2TRIGGEN**			**RES**					**DU1TRIGGEN**		
r					rw			r					rw		

Field	Bits	Type	Description
DU1TRIGGEN	2:0	rw	**Differential unit trigger enable** *Note: These bits configure the enable input of the differential unit.* 000_B **CC60**: CC60 is selected 001_B **CC61**: CC61 is selected 010_B **CC62**: CC62 is selected 011_B **COUT60**: COUT60 is selected 100_B **COUT61**: COUT61 is selected 101_B **COUT62**: COUT62 is selected 110_B **T3OUT**: T3OUT is selected 111_B **COUT6**: COUT63 is selected
RES	7:3, 15:11, 23:19, 31:27	r	**Reserved** Returns 0 if read; should be written with 0.
DU2TRIGGEN	10:8	rw	**Differential unit trigger enable** *Note: These bits configure the enable input of the differential unit.* 000_B **CC60**: CC60 is selected 001_B **CC61**: CC61 is selected 010_B **CC62**: CC62 is selected 011_B **COUT60**: COUT60 is selected 100_B **COUT61**: COUT61 is selected 101_B **COUT62**: COUT62 is selected 110_B **T3OUT**: T3OUT is selected 111_B **COUT63**: COUT63 is selected
DU3TRIGGEN	18:16	rw	**Differential unit trigger enable** *Note: These bits configure the enable input of the differential unit.* 000_B **CC60**: CC60 is selected 001_B **CC61**: CC61 is selected 010_B **CC62**: CC62 is selected

(table continues...)

(continued)

Field	Bits	Type	Description
			011_B **COUT60**: COUT60 is selected 100_B **COUT61**: COUT61 is selected 101_B **COUT62**: COUT62 is selected 110_B **T3OUT**: T3OUT is selected 111_B **COUT63**: COUT63 is selected
DU4TRIGGEN	26:24	rw	**Differential unit trigger enable** *Note: These bits configure the enable input of the differential unit.* 000_B **CC60**: CC60 is selected 001_B **CC61**: CC61 is selected 010_B **CC62**: CC62 is selected 011_B **COUT60**: COUT60 is selected 100_B **COUT61**: COUT61 is selected 101_B **COUT62**: COUT62 is selected 110_B **T3OUT**: T3OUT is selected 111_B **COUT63**: COUT63 is selected

6.15.54 Peripheral management control register

SCU_PMCON Offset address: 0060_H

Peripheral management control register RESET_TYPE_3 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					T21_DIS	RES	SSC2_DIS	RES			GPT12_DIS	T2_DIS	CCU6_DIS	SSC1_DIS	ADC1_DIS
r					rw	r	rw	r			rw	rw	rw	rw	rw

Field	Bits	Type	Description
ADC1_DIS	0	rw	**ADC1 disable request, active high** 0_B **NORMAL**: ADC1 is in normal operation (default) 1_B **DISABLE**: Request to disable the ADC
SSC1_DIS	1	rw	**SSC1 disable request, active high** 0_B **NORMAL**: SSC is in normal operation (default) 1_B **DISABLE**: Request to disable the SSC
CCU6_DIS	2	rw	**CCU6 disable request, active high** 0_B **NORMAL**: CCU6 is in normal operation (default) 1_B **DISABLE**: Request to disable the CCU6
T2_DIS	3	rw	**T2 disable request, active high** 0_B **NORMAL**: T2 is in normal operation (default) 1_B **DISABLE**: Request to disable the T2
GPT12_DIS	4	rw	**General purpose timer 12 disable request, active high** 0_B **NORMAL**: GPT12 is in normal operation (default) 1_B **DISABLE**: Request to disable the GPT12
RES	7:5, 9, 31:11	r	**Reserved** Returns 0 if read; should be written with 0.
SSC2_DIS	8	rw	**SSC2 disable request, active high** 0_B **NORMAL**: SSC is in normal operation (default) 1_B **DISABLE**: Request to disable the SSC
T21_DIS	10	rw	**T21 disable request, active high** 0_B **NORMAL**: T21 is in normal operation (default) 1_B **DISABLE**: Request to disable the T21

6.15.55 Module suspend control register

SCU_MODSUSP Offset address: $00C8_H$

Module suspend control register RESET_TYPE_3 value: $0000\ 0081_H$

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_SUSP	MU_SUSP	RES	WDT1SUSP	T21_SUSP	RES	GPT12_SUSP	T2_SUSP	T13SUSP	T12SUSP	RES1
r					rw	rw	r	rw	rw	r	rw	rw	rw	rw	r

Field	Bits	Type	Description
RES1	0	r	**Reserved** Returns 1 if read.
T12SUSP	1	rw	**Timer 12 debug suspend bit** When suspended, additionally the T12 PWM outputs are set to inactive level and capture inputs are disabled. 0_B **NOT_SUSPENDED**: Timer12 in capture/compare unit will not be suspended 1_B **SUSPENDED**: Timer12 in capture/compare unit will be suspended
T13SUSP	2	rw	**Timer 13 debug suspend bit** When suspended, additionally the T13 PWM output is set to inactive level. 0_B **NOT_SUSPENDED**: Timer13 in capture/compare unit will not be suspended 1_B **SUSPENDED**: Timer13 in capture/compare unit will be suspended
T2_SUSP	3	rw	**Timer 2 debug suspend bit** 0_B **NOT_SUSPENDED**: Timer2 will not be suspended 1_B **SUSPENDED**: Timer2 will be suspended
GPT12_SUSP	4	rw	**GPT12 debug suspend bit** 0_B **NOT_SUSPENDED**: GPT12 will not be suspended 1_B **SUSPENDED**: GPT12 will be suspended
RES	5, 8, 31:11	r	**Reserved** Returns 0 if read; should be written with 0.
T21_SUSP	6	rw	**Timer 21 debug suspend bit** 0_B **NOT_SUSPENDED**: Timer21 will not be suspended 1_B **SUSPENDED**: Timer21 will be suspended
WDT1SUSP	7	rw	**Watchdog timer 1 debug suspend bit** 0_B **NOT_SUSPENDED**: WDT1 will not be suspended

(table continues...)

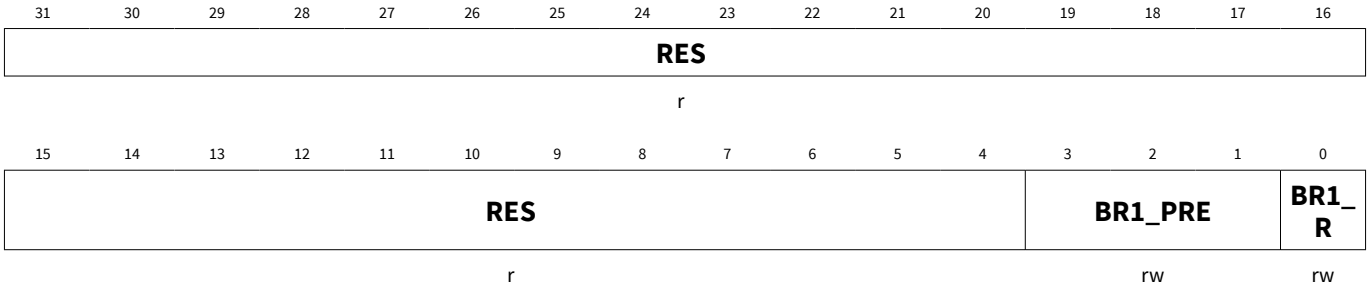
(continued)

Field	Bits	Type	Description
			1_B **SUSPENDED**: WDT1 will be suspended
MU_SUSP	9	rw	**Measurement unit debug suspend bit** 0_B **NOT_SUSPENDED**: MU will not be suspended 1_B **SUSPENDED**: MU will be suspended
ADC1_SUSP	10	rw	**ADC1 unit debug suspend bit** 0_B **NOT_SUSPENDED**: ADC1 will not be suspended 1_B **SUSPENDED**: ADC1 will be suspended

6.15.56 Baud-rate control 1 register

SCU_BCON1 Offset address: 0088_H

Baud-rate control 1 register RESET_TYPE_3 value: $0000\ 0000_H$

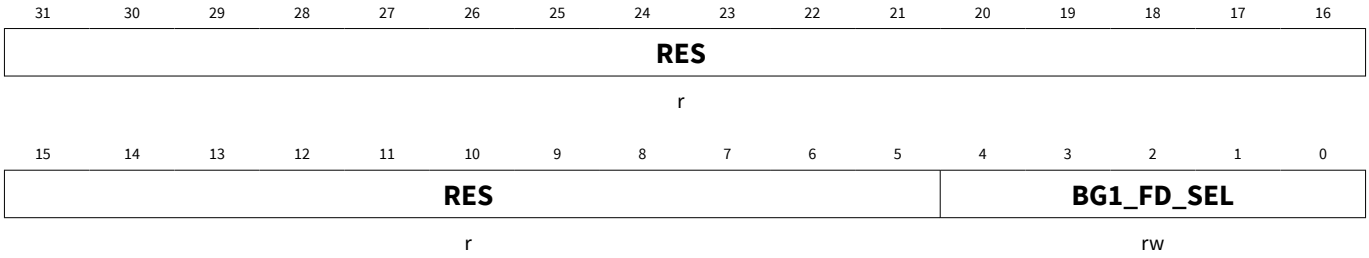


Field	Bits	Type	Description
BR1_R	0	rw	**Baud-rate generator run control bit** *Note: BR_VALUE should only be written if R = 0.* 0_B **DISABLED**: Baud-rate generator disabled 1_B **ENABLED**: Baud-rate generator enabled
BR1_PRE	3:1	rw	**Prescaler bit** Selects the input clock for which is derived from the peripheral clock. Others: reserved 000_B **1**: fDIV = fPCLK 001_B **2**: fDIV = fPCLK/2 010_B **4**: fDIV = fPCLK/4 011_B **8**: fDIV = fPCLK/8 100_B **16**: fDIV = fPCLK/16 101_B **32**: fDIV = fPCLK/32
RES	31:4	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.57 Baud-rate timer/reload, low byte 1 register

SCU_BGL1 Offset address: $008C_H$

Baud-rate timer/reload, low byte 1 register RESET_TYPE_3 value: $0000\ 0000_H$

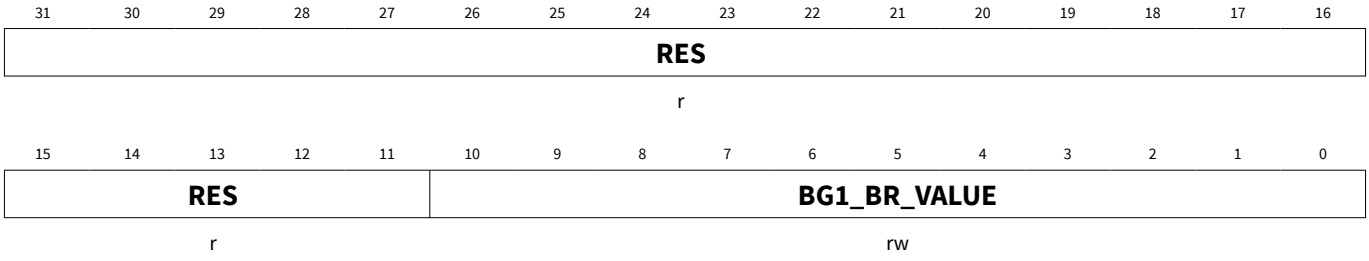


Field	Bits	Type	Description
BG1_FD_SEL	4:0	rw	**Fractional divider selection** Selects the fractional divider to be n/32, where n is the value of FD_SEL and is in the range of 0 to 31. For example, writing 0001_B to FD_SEL selects the fractional divider to be 1/32. *Note: Fractional divider has no effect if BR_VALUE = 000_H.*
RES	31:5	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.58 Baud-rate timer/reload 1 register

SCU_BG1 Offset address: 0090_H

Baud-rate timer/reload 1 register RESET_TYPE_3 value: $0000\ 0000_H$

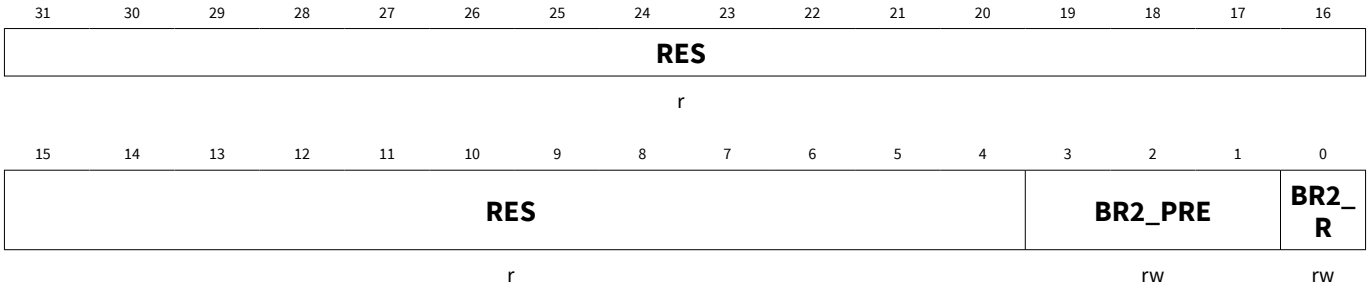


Field	Bits	Type	Description
BG1_BR_VALUE	10:0	rw	**Baud-rate timer/reload value UART1** 11-bit baud-rate timer/reload value. *Note: If the baud-rate generation is running this register shows the actual timer value.* The definition of the 11-bit reload value is as follows (other bit combinations equivalent): 000_H **BYPASSED**: Baud-rate timer is bypassed 001_H **1**: 1 002_H **2**: 2 $7FE_H$ **2046**: 2046 $7FF_H$ **2047**: 2047
RES	31:11	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.59 Baud-rate control 2 register

SCU_BCON2 Offset address: 0098_H

Baud-rate control 2 register RESET_TYPE_3 value: $0000\ 0000_H$

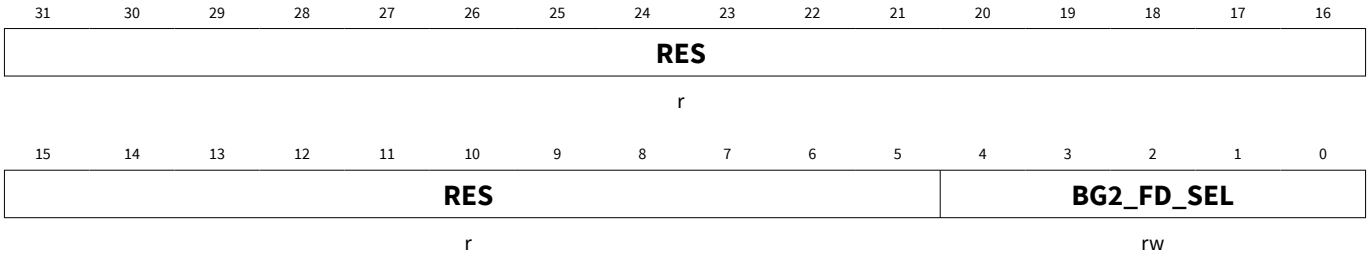


Field	Bits	Type	Description
BR2_R	0	rw	**Baud-rate generator run control bit** *Note: BR_VALUE should only be written if R = 0.* 0_B **DISABLED**: Baud-rate generator disabled 1_B **ENABLED**: Baud-rate generator enabled
BR2_PRE	3:1	rw	**Prescaler bit** Selects the input clock for which is derived from the peripheral clock. Others: reserved 000_B **1**: fDIV = fPCLK 001_B **2**: fDIV = fPCLK/2 010_B **4**: fDIV = fPCLK/4 011_B **8**: fDIV = fPCLK/8 100_B **16**: fDIV = fPCLK/16 101_B **32**: fDIV = fPCLK/32
RES	31:4	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.60 Baud-rate timer/reload, low byte 2 register

SCU_BGL2 Offset address: $009C_H$

Baud-rate timer/reload, low byte 2 register RESET_TYPE_3 value: $0000\ 0000_H$

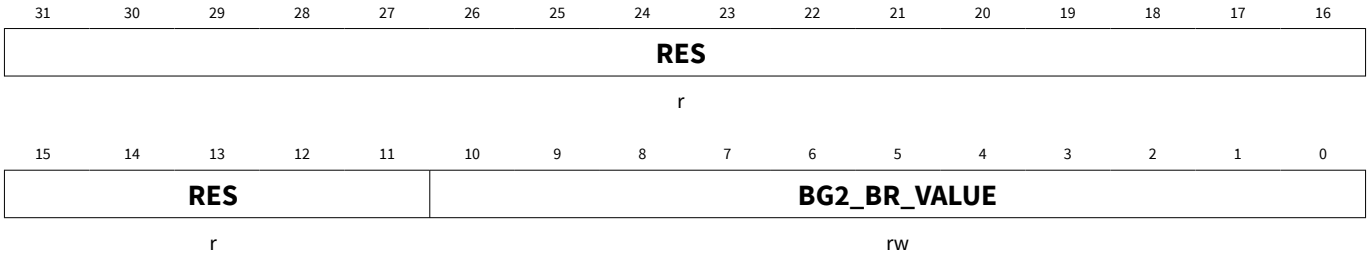


Field	Bits	Type	Description
BG2_FD_SEL	4:0	rw	**Fractional divider selection** Selects the fractional divider to be n/32, where n is the value of FD_SEL and is in the range of 0 to 31. For example, writing 0001_B to FD_SEL selects the fractional divider to be 1/32. *Note: Fractional divider has no effect if BR_VALUE = 000_H.*
RES	31:5	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.61 Baud-rate timer/reload 2 register

SCU_BG2 Offset address: $00A0_H$

Baud-rate timer/reload 2 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
BG2_BR_VALUE	10:0	rw	**Baud-rate timer/reload value UART2** 11-bit baud-rate timer/reload value. *Note: If the baud-rate generation is running this register shows the actual timer value.* The definition of the 11-bit reload value is as follows (other bit combinations equivalent): 000_H **BYPASSED**: Baud-rate timer is bypassed 001_H **1**: 1 002_H **2**: 2 $7FE_H$ **2046**: 2046 $7FF_H$ **2047**: 2047
RES	31:11	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.62 LIN status register

SCU_LINST Offset address: 0094_H

LIN status register RESET_TYPE_3 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									**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 **ENABLED**: Break/sync detection is enabled 1_B **DISABLED**: Break/sync 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 "BGSEL bit field definition for different input frequencies" for more information.
BRK	3	r	**Break field flag** This bit is set by hardware and can only be cleared by software. 0_B **NOT_DETECTED**: Break field is not detected 1_B **DETECTED**: 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 **NOT_DETECTED**: End of SYN byte is not detected 1_B **DETECTED**: 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 **NOT_DETECTED**: Error is not detected in SYN byte 1_B **DETECTED**: Error is detected in SYN byte
SYNEN	6	rw	**End of SYN byte and SYN byte error interrupts enable** 0_B **DISABLED**: End of SYN byte and SYN byte error interrupts are not enabled 1_B **ENABLED**: End of SYN byte and SYN byte error interrupts are enabled
RES	31:7	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.63 LIN status clear register

SCU_LINSCLR Offset address: $00A4_H$

LIN status clear register RESET_TYPE_3 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										ERRSYNC	EOFSYNC	BRKC	RES		
r										w	w	w	r		

Field	Bits	Type	Description
RES	2:0, 31:6	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 **NOT_CLEARED**: Break field is not cleared 1_B **CLEARED**: 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 **NOT_CLEARED**: End of SYN byte is not cleared 1_B **CLEARED**: 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 **NOT_CLEARED**: Error in SYN byte not cleared 1_B **CLEARED**: Error in SYN byte cleared

6.15.64 Error detection and correction control register

SCU_EDCCON Offset address: $00D4_H$

Error detection and correction control register RESET_TYPE_3 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													**NVMIE**	**RES**	**RIE**
r													rw	r	rw

Field	Bits	Type	Description
RIE	0	rw	**RAM double bit ECC error interrupt enable** 0_B **0**: No NMI is generated when a double bit ECC error occurs reading RAM 1_B **1**: An NMI is generated when a double bit ECC error occurs reading RAM
RES	1, 31:3	r	**Reserved** Returns 0 if read; should be written with 0.
NVMIE	2	rw	**NVM double bit ECC error interrupt enable** 0_B **0**: No NMI is generated when a double bit ECC error occurs reading NVM 1_B **1**: An NMI is generated when a double bit ECC error occurs reading NVM

6.15.65 Error detection and correction status register

SCU_EDCSTAT Offset address: $00D8_H$

Error detection and correction status register RESET_TYPE_4 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**	**RSBE**	**RES**	**NVMDBE**	**RES**	**RDBE**
r										r	r	r	r	r	r

Field	Bits	Type	Description
RDBE	0	r	**RAM double bit error** This bit is set by hardware and can be cleared only by software. 0_B **0**: No double bit error on RAM has occurred 1_B **1**: A double bit error on RAM has occurred
RES	1, 3, 5, 31:6	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 **0**: No double bit error on NVM has occurred 1_B **1**: A double bit error on NVM has occurred
RSBE	4	r	**RAM single bit error** This bit is set by hardware and can be cleared only by software. 0_B **0**: No single bit error on RAM has occurred 1_B **1**: A single bit error on RAM has occurred

6.15.66 Error detection and correction status clear register

SCU_EDCSCLR Offset address: $010C_H$

Error detection and correction status clear register RESET_TYPE_3 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	RSBEC	RES	NVMDBEC	RES	RDBEC
r										r	w	r	w	r	w

Field	Bits	Type	Description
RDBEC	0	w	**RAM double bit error clear** This bit is set by software and can be cleared only by hardware. 0_B **NOT_CLEARED**: A double bit error on RAM is not cleared 1_B **CLEARED**: A double bit error on RAM is cleared
RES	1, 3, 5, 31:6	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 **NOT_CLEARED**: A double bit error on NVM is not cleared 1_B **CLEARED**: A double bit error on NVM is cleared
RSBEC	4	w	**RAM single bit error clear** This bit is set by software and can be cleared only by hardware. 0_B **NOT_CLEARED**: A single bit error on RAM is not cleared 1_B **CLEARED**: A single bit error on RAM is cleared

6.15.67 System startup status register

It contains the main system control and status bits.

SCU_SYS_STRTUP_STS Offset address: 0074_H

System startup status register Reset values see: Table 55

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				PG100TP_CHKS_ERR	MRAMINITSTS	INIT_FAIL
r									r				rwpt	rwpt	rwpt

Field	Bits	Type	Description
INIT_FAIL	0	rwpt	**Initialization at startup failed** This bit is a logical OR between PLL_LOCK failure, Map RAM initialization failure and trimming values checksum error. *Note: This bit is affected by every RESET_TYPE.* 0_B **NO_ERROR**: No initialization error at startup 1_B **ERROR**: Initialization error at startup
MRAMINITSTS	1	rwpt	**Map RAM initialization status** Status of Map RAM initialization. *Note: This bit is affected by every RESET_TYPE.* 0_B **NO_FAIL**: Map RAM initialization was successful 1_B **FAIL**: Map RAM initialization was not successful
PG100TP_CHKS_ERR	2	rwpt	**100 TP Page checksum error** Initialization status of trimming parameters from NVM. *Note: This bit is affected by every RESET_TYPE.* 0_B **OK**: Initialization of trimming parameters from NMV was successful (checksum was correct) 1_B **NOK**: Initialization of trimming parameter from NMV was not successful (checksum was notcorrect). As a backup default values form Boot-ROM are used
RES	6:3, 31:7	r	**Reserved** Returns 0 if read; should be written with 0.

Table 55 Reset values of **SCU_SYS_STRTUP_STS**

Reset type	Reset value	Note
RESET_TYPE_3	0000 0000_H	Reset mask = “0b00000000000000000000000000110000”
RESET_TYPE_4	0000 0000_H	Reset mask = “0b11111111111111111111111111001111”

6.15.68 Identity register

The identity register identifies the product and the versioning.

SCU_ID Offset address: $00A8_H$

Identity register RESET_TYPE_3 value: $0000\ 0080_H$

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

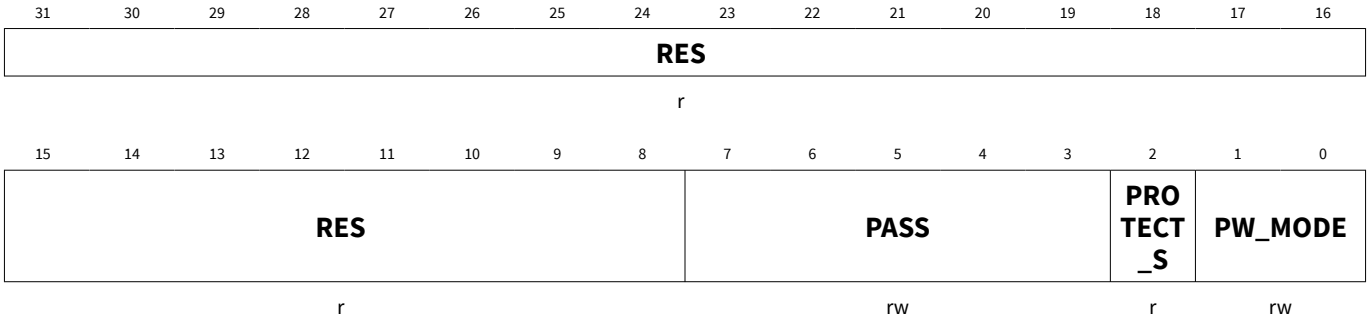
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								**PRODID**					**VERID**		
r								r					rw		

Field	Bits	Type	Description
VERID	2:0	rw	**Version ID** Defines the stepping code of the device. 001_B 010_B
PRODID	7:3	r	**Product ID** 10000_B
RES	31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.69 Password register

SCU_PASSWD Offset address: $00AC_H$

Password register RESET_TYPE_3 value: $0000\ 0007_H$



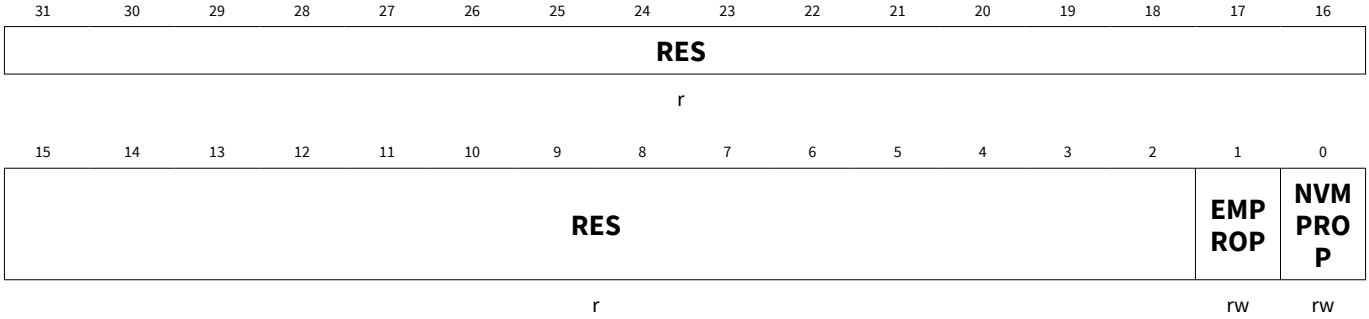
Field	Bits	Type	Description
PW_MODE	1:0	rw	**Bit protection scheme control bit** These two bits cannot be written directly. To change the value between 11_B and 00_B, the bit field PASS must be written with 11000_B, only then the MODE[1:0] will be registered. Other bit combinations: Scheme enabled 00_B **DISABLED**: Scheme disabled 11_B **ENABLED**: Scheme enabled (default)
PROTECT_S	2	r	**Bit protection signal status bit** This bit shows the status of the protection. 0_B **NOT_PROTECTED**: Software is able to write to all protected bits 1_B **PROTECTED**: Software is unable to write to any protected bits
PASS	7:3	rw	**Password bits** The bit protection scheme only recognizes three patterns. This bit field is always read as '0'. 13_H **OPENED**: Opens access to writing of all protected bits 15_H **CLOSED**: Closes access to writing of all protected bits 18_H **ENABLED**: Enables writing of the bit field MODE
RES	31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.70 Emergency and program operation status register

This register indicates the emergency and program operation status.
For Boot ROM to indicate NVM initialization status upon completion of startup:

SCU_EMOP Offset address: $00CC_H$

Emergency and program operation status register RESET_TYPE_4 value: $0000\ 0000_H$



Field	Bits	Type	Description
NVMPROP	0	rw	**NVM program operation status bit** This bit is used to monitor the status of the NVM program operation. 0_B **NOT_STARTED**: No NVM program operation is started 1_B **STARTED**: NVM program operation is started
EMPROP	1	rw	**Emergency program operation status bit** This bit is used to monitor the status of the emergency program operation. 0_B **NOT_STARTED**: No emergency program operation is started 1_B **STARTED**: Emergency program operation is started
RES	31:2	r	**Reserved** Returns 0 if read; should be written with 0.

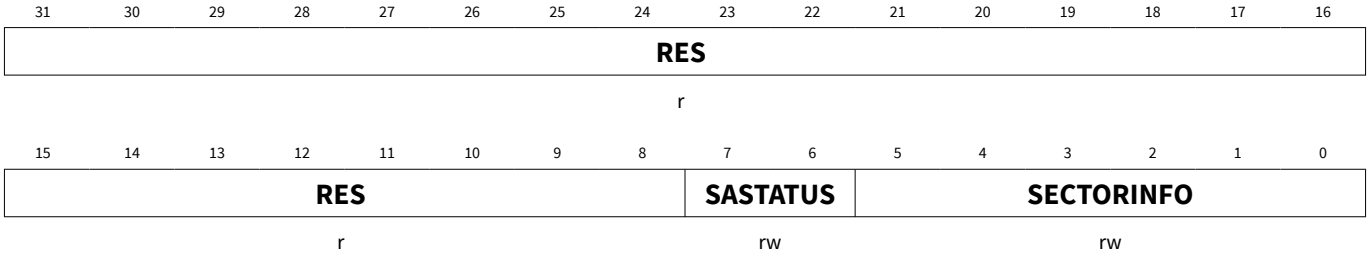
6.15.71 Memory status register

The memory status register can be used in two ways. Upon the completion of the Boot ROM startup following a reset, the register stores the NVM initialization status. Subsequently, the register can be used by the user code to store the status of the NVM program and emergency program operation status.

For Boot ROM to indicate NVM initialization status upon completion of startup:

SCU_MEMSTAT Offset address: $00DC_H$

Memory status register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
SECTORINFO	5:0	rw	**Sector information** 01_H to 10_H, which represent the different sector addresses. For values not within this range, the data will be considered invalid. Once the SA has been executed, regardless of the execution status, the last accessed sector information will be stored here.
SASTATUS	7:6	rw	**Service algorithm status** 00_B **SECTORINFO**: Depending on SECTORINFO Depending on SECTORINFO, there are two possible outcomes: For SECTORINFO = 00_H, NVM initialization is successful and no SA is executed. For SECTORINFO = values other than 00_H, SA execution is successful and only one map error is fixed 01_B **SUCCESS**: SA execution is successful 10_B **NO_SUCCESS**: SA execution is not successful. Map error exists in the mapped sector 11_B **NO_SUCCESS**: SA execution is not successful. Map error exists in the mapped sector
RES	31:8	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.72 NVM protection status register

This register reflects the NVM protection status. It is written by firmware only.

SCU_NVM_PROT_STS Offset address: $00E0_H$

NVM protection status register RESET_TYPE_4 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
NVMBSL		CBSL_PW	LIN_PW	NL_PW	DIS_RDUS_S0	DIS_RDUS	EN_RD_S0	RES		EN_RD_CBSL	EN_RD_LIN	EN_RD_NL	EN_PRG_CBSL	EN_PRG_LIN	EN_PRG_NL
rw		rw	rw	rw	rw	rw	rw	r		rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
EN_PRG_NL	0	rw	**NVM protection of data in non-linear sectors** 0_B **NO_CHANGE**: The data in sectors of the non-linearly mapped area can not be changed 1_B **CHANGE**: The data in sectors of the non-linearly mapped area can be changed (erased or written)
EN_PRG_LIN	1	rw	**NVM protection of data in linear sectors** 0_B **NO_CHANGE**: The data in sectors of the linearly mapped area can not be changed 1_B **CHANGE**: The data in sectors of the linearly mapped area can be changed (erased or written)
EN_PRG_CBSL	2	rw	**NVM protection of data in CBSL region** 0_B **NO_CHANGE**: The data in region defined by NVMBSL can not be changed 1_B **CHANGE**: The data in region defined by NVMBSL can be changed (erased or written)
EN_RD_NL	3	rw	**NVM read protection of data in non-linear sectors** 0_B **NO_READ**: The data in sectors of the non-linearly mapped area can not be read 1_B **READ**: The data in sectors of the non-linearly mapped area can be read
EN_RD_LIN	4	rw	**NVM read protection of data in linear sectors** 0_B **NO_READ**: The data in sectors of the linearly mapped area can not be read 1_B **READ**: The data in sectors of the linearly mapped area can be read
EN_RD_CBSL	5	rw	**NVM read protection of data in CBSL region** 0_B **NO_READ**: The data in region defined by NVMBSL can not be read 1_B **READ**: The data in region defined by NVMBSL sectors of can be read
RES	7:6,	r	**Reserved**

(table continues...)

(continued)

Field	Bits	Type	Description
	31:16		Returns 0 if read; should be written with 0.
EN_RD_S0	8	rw	**NVM read protection for sector 0** 0_B **NO_READ**: The data in sector 0 can not be read over AHB-Lite interface 1_B **READ**: The data in sector 0 can be read over AHB-Lite interface
DIS_RDUS	9	rw	**Configuration of NVM read protection for sector 1...n with EN_RD_* = 0** 0_B **NVM_READ_UNSAVE**: NVM read unsave 1_B **INDEPENTENT**: Write accesses to sector 1...n are prevented
DIS_RDUS_S0	10	rw	**Configuration of NVM read protection for sector 0 with EN_RD_S0 = 0** 0_B **NVM_READ_S0_UNSAVE**: NVM read S0 unsave 1_B **INDEPENDENT**: Write accesses to sector 0 are prevented
NL_PW	11	rw	**Status of non-linear region password/protection** 0_B **NOT_PROTECTED**: Non-linear region password is not installed; linear region is not protected 1_B **PROTECTED**: Non-linear region password is installed; linear region is protected
LIN_PW	12	rw	**Status of linear region password/protection** 0_B **NOT_PROTECTED**: Linear region password is not installed; linear region is not protected 1_B **PROTECTED**: Linear region password is installed; linear region is protected
CBSL_PW	13	rw	**Status of CBSL region password/protection** 0_B **NOT_PROTECTED**: CBSL region password is not installed; CBSL region is not protected 1_B **PROTECTED**: CBSL region password is installed; CBSL region is protected
NVMBSL	15:14	rw	**CBSL region size definition** Size definition of customer BSL region 00_B **4**: CBSL size is 4 K 01_B **8**: CBSL size is 8 K 10_B **12**: CBSL size is 12 K 11_B **16**: CBSL size is 16 K

6.15.73 Memory access status register

This register reflects the memory access status of all system memories. Software can only clear this register.

SCU_MEM_ACC_STS Offset address: $00E4_H$

Memory access status register RESET_TYPE_4 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											ROM_PROT_ERR	NVM_SFR_ADDR_ERR	NVM_SFR_PROT_ERR	NVM_ADDR_ERR	NVM_PROT_ERR
r											rh	rh	rh	rh	rh

Field	Bits	Type	Description
NVM_PROT_ERR	0	rh	**NVM access protection** 0_B **NO_ERROR**: No Protection error 1_B **ERROR**: Protection error
NVM_ADDR_ERR	1	rh	**NVM address protection** 0_B **NO_ERROR**: No Protection error 1_B **ERROR**: Protection error
NVM_SFR_PROT_ERR	2	rh	**NVM SFR access protection** 0_B **NO_ERROR**: No Protection error 1_B **ERROR**: Protection error
NVM_SFR_ADDR_ERR	3	rh	**NVM SFR address protection** 0_B **NO_ERROR**: No Protection error 1_B **ERROR**: Protection error
ROM_PROT_ERR	4	rh	**ROM access protection** 0_B **NO_ERROR**: No Protection error 1_B **ERROR**: Protection error
RES	31:5	r	**Reserved** Returns 0 if read; should be written with 0.

6.15.74 UART1 control/status register

Refer to register UART_SCON in Chapter 18.

SCU_SCON1 Address: xxx_H

UART1 control/status register 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														TI	RI
r														rwh	rwh

Field	Bits	Type	Description
RES	31:2	r	**Reserved** Returns 0 if read; should be written with 0.
TI	1	rwh	**Serial interface transmitter interrupt flag** Set by hardware at the end of a serial data transmission. Must be cleared by software.
RI	0	rwh	**Serial interface receiver interrupt flag** Set by hardware if a serial data byte has been received. Must be cleared by software.

6.15.75 UART2 control/status register

Refer to register UART_SCON in Chapter 18.

SCU_SCON2 Address: xxx_H

UART2 control/status register 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														TI	RI
r														rwh	rwh

Field	Bits	Type	Description
RES	31:2	r	**Reserved** Returns 0 if read; should be written with 0.
TI	1	rwh	**Serial interface transmitter interrupt flag** Set by hardware at the end of a serial data transmission. Must be cleared by software.
RI	0	rwh	**Serial interface receiver interrupt flag** Set by hardware if a serial data byte has been received. Must be cleared by software.

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

7.1 Description of the power modules system control unit

The system control unit of the power modules consists of the following submodules:

- Clock watchdog unit (CWU): Supervision of all power modules relevant clocks with NMI signaling
- Interrupt control unit (ICU): All system relevant interrupt flags and status flags
- Power control unit (PCU): Takes over control when device enters and exits sleep and stop mode
- External watchdog (WDT1): Independent system watchdog to monitor 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:

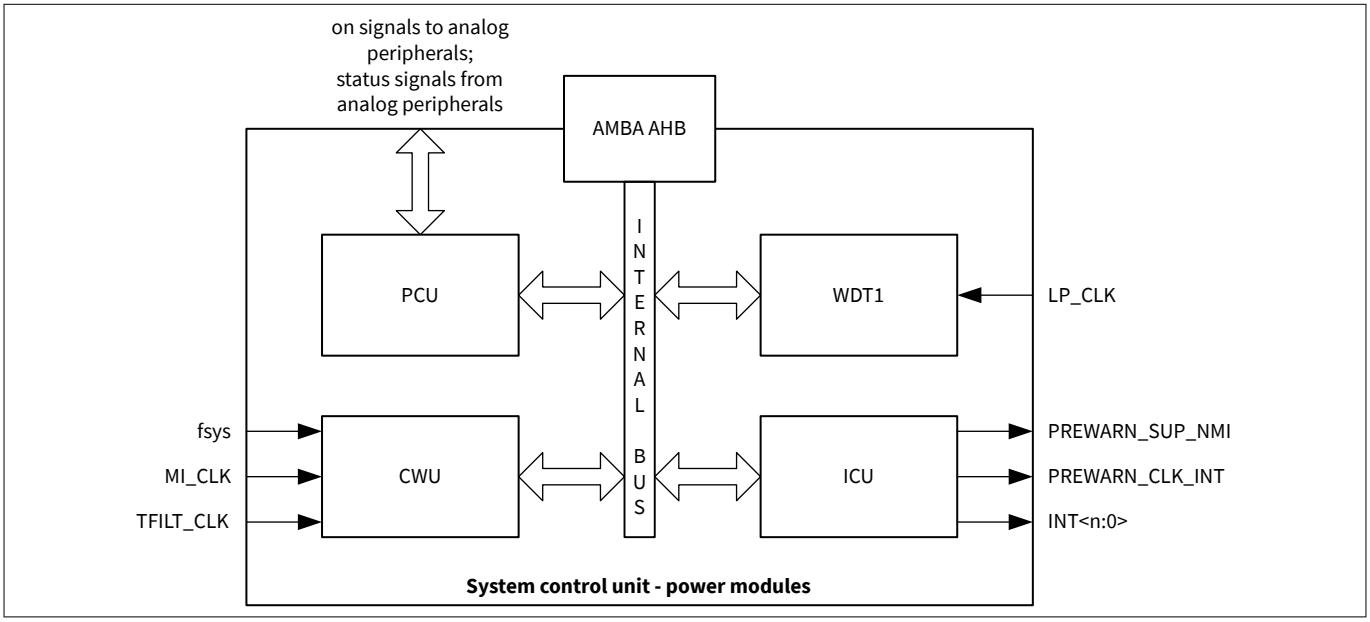


Figure 38 Block diagram of system control unit - power modules

IO description of SCU_PM

- CWU (clock watchdog unit)
  - check of f_{sys} = system frequency: output of PLL
  - check of MI_CLK = measurement interface clock (analog clock): derived out of f_{sys} by division factors 1/2/3/4
  - check of TFILT_CLK = clock used for digital filters: derived out of f_{sys} by configurable division factors
- ICU (interrupt control unit)
  - PREWARN_SUP_NMI = generation of pre-warn supply NMI
  - PREWARN_CLK_INT = generation of pre-warn clock watchdog NMI
  - INT = generation of MISC interrupts

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:
  - Low sides
  - High sides
  - LIN

As shown in the following figure 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. 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:

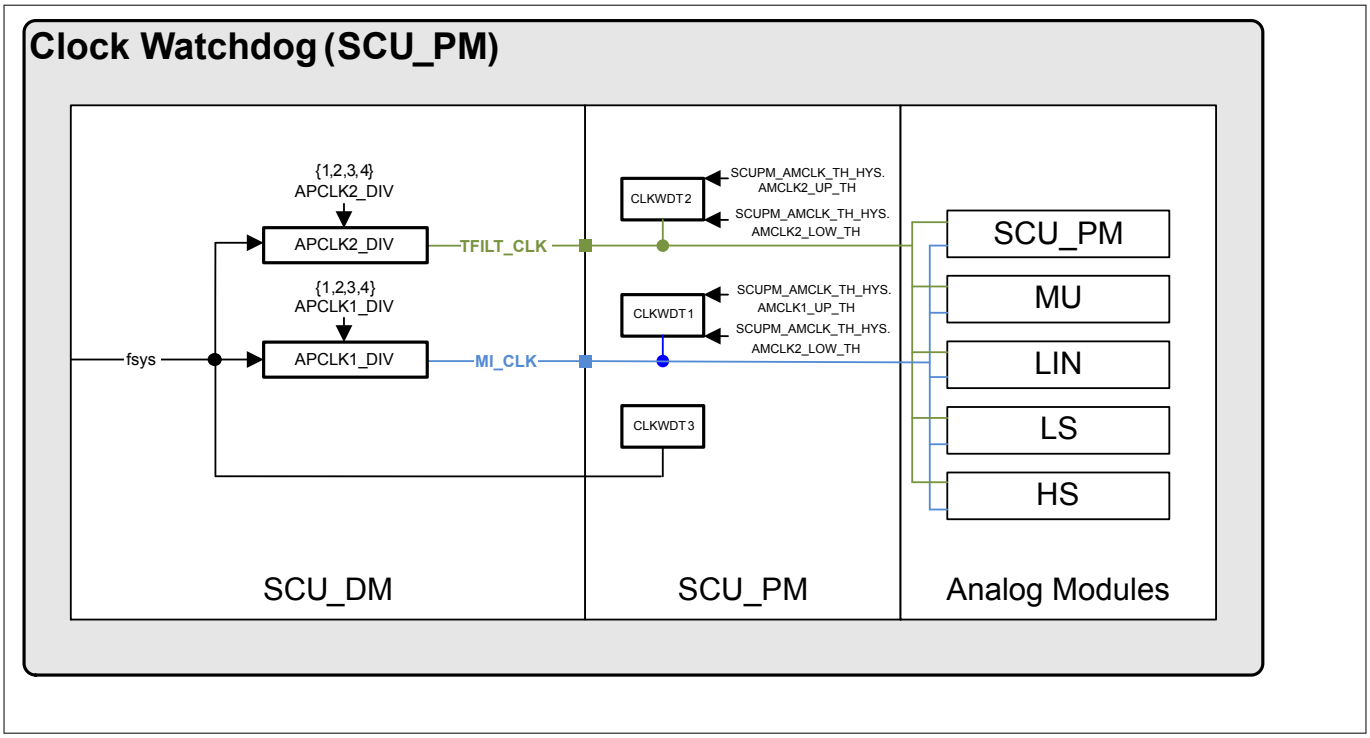


Figure 39 Block diagram of CGU including clock watchdogs

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 registers

The analog module clock generation unit is fully controllable by the registers listed below.

The registers are addressed wordwise.

7.3.2.1 Register overview - Clock Watchdog Unit registers (ascending offset address)

Table 56 Register overview - Clock Watchdog Unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCUPM_AMCLK_FREQ_STS	Analog module clock frequency status register	0000_H	253
SCUPM_AMCLK_CTRL	Analog module clock control register	0004_H	254
SCUPM_AMCLK_TH_HYS	Analog module clock limit register	$000C_H$	255
SCUPM_STCALIB	System tick calibration register	$006C_H$	256

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 and 10-bit ADC

  The supply voltages VS, VBAT_SENSE, VDDP and VDDC are monitored by the measurement unit and the 10-bit ADC module. The measurement unit can be considered as an independent monitoring instance for external supply voltages and internal voltages generated by PMU. This monitoring is done with an independent reference and supply voltage to ensure fail safe operation.
- 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 are detected.

The following figure shows the structure of the PREWARN_SUP:

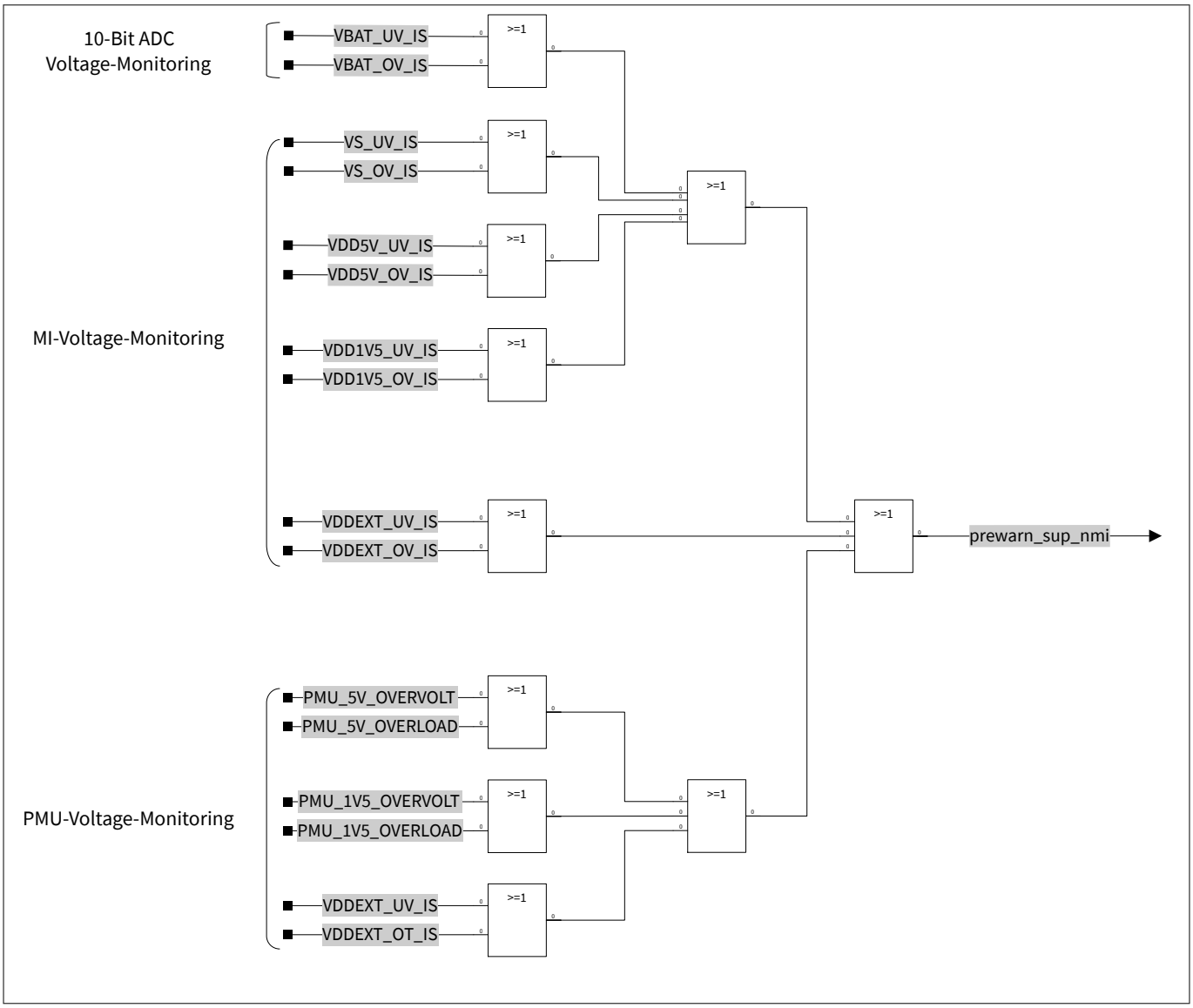


Figure 40 Structure of PREWARN_SUP

All PREWARN_SUP related flags are grouped in register SCUPM_SYS_SUPPLY_IRQ_STS. All measurement interface related flags are edge triggered (attribute rwhe). 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 registers

All analog modules interrupt functionality is is fully controllable by the registers listed below.

The registers are addressed wordwise.

7.4.2.1 Register overview - Interrupt Control Unit registers (ascending offset address)

Table 57 Register overview - Interrupt Control Unit registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCUPM_SYS_ISCLR	System interrupt status clear register	0014_H	257
SCUPM_SYS_IS	System interrupt status register	0018_H	258
SCUPM_SYS_SUPPLY_IRQ_STS	System supply interrupt status register	$001C_H$	261
SCUPM_SYS_SUPPLY_IRQ_CTRL	System supply interrupt control register	0020_H	263
SCUPM_SYS_SUPPLY_IRQ_CLR	System supply interrupt status clear register	0024_H	265
SCUPM_SYS_IRQ_CTRL	System interrupt control register	0028_H	267

7.4.2.2 Interrupt control unit status overview registers

Due to the large variety of diagnosis possibilities of MOTIX™ TLE984xQX, the system offers several overview registers, to help the user finding the right source of interrupt.

Overview register, switches interrupt status register and system supply interrupt status register

- SCUPM_SYS_SUPPLY_IRQ_STS: Flags for undervoltage and overvoltage detection for all system relevant supplies. These interrupts are edge triggered interrupts.
- SCUPM_SYS_IS: Interrupts for analog modules.

7.4.2.3 Interrupt control unit - interrupt clear registers

The analog module Interrupts can be cleared by their corresponding enable bits which are located in the registers:

- SCUPM_SYS_SUPPLY_IRQ_CLR: Clear of interrupts for undervoltage and overvoltage detection for all system relevant supplies. These interrupts are edge triggered interrupts to reduce interrupt load of the µC.
- SCUPM_SYS_ISCLR: Clear of interrupts related to analog modules.

7.4.2.4 Interrupt control unit - interrupt enable registers

The analog module interrupts can be enabled and disabled by there corresponding enable bits which are located in registers:

- SCUPM_SYS_SUPPLY_IRQ_CTRL: Enable of interrupts for undervoltage and overvoltage detection for all system relevant supplies. These interrupts are edge triggered interrupts to reduce interrupt load of the µC.
- SCUPM_SYS_IRQ_CTRL: Enable of interrupts for analog modules.

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
- Low-side drivers
- High-side drivers

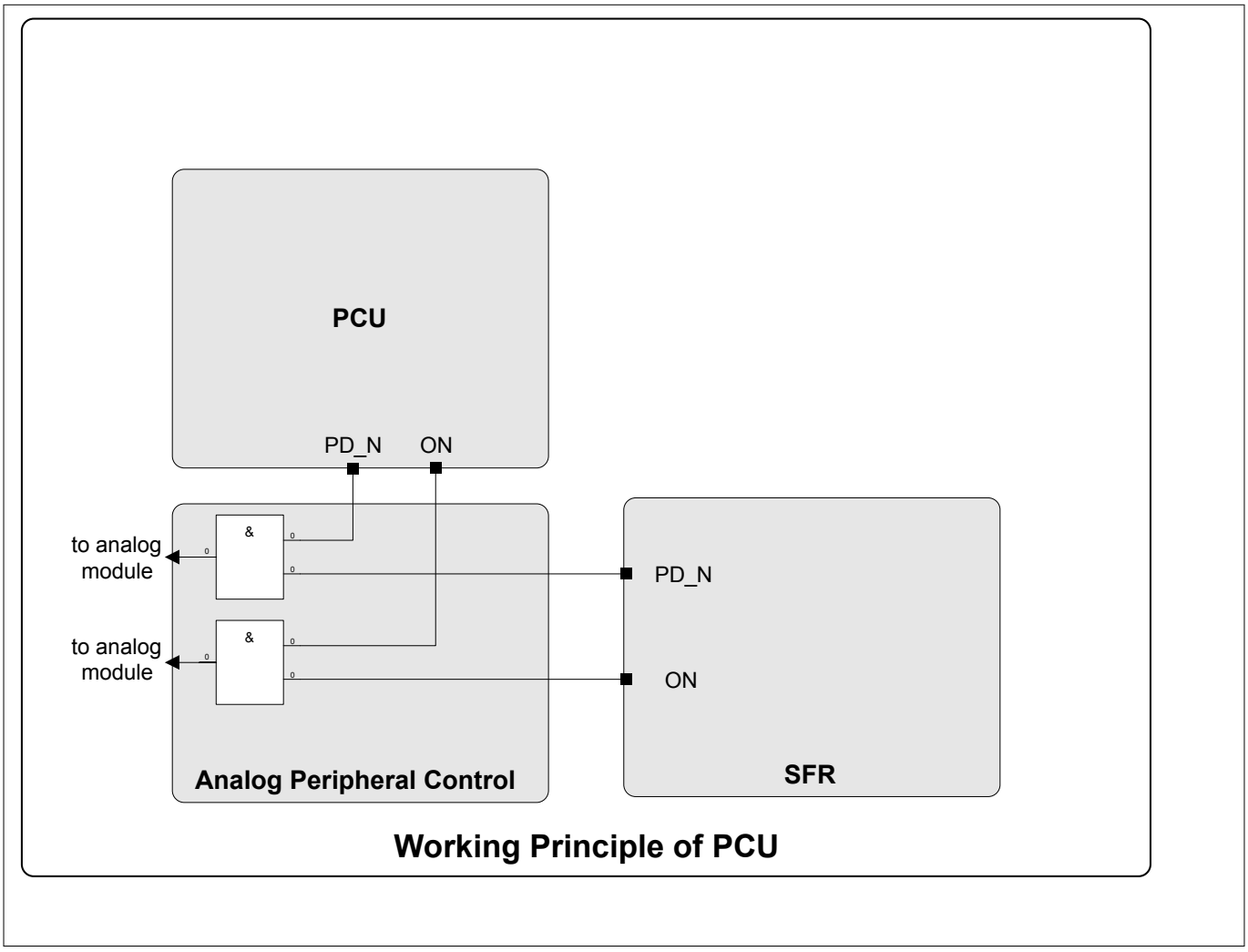


Figure 41 Function of AP_SUB_CTRL

If the device will power up the analog modules state machine will start-up 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 Overtemperature system shutdown

In case of overtemperature (T_j > 150 °C) the system will be set to sleep mode. This functionality is used to protect the system from thermal overstress. One possibility of avoiding this thermal shutdown is to stick to an emergency procedure, which helps to minimize the power dissipation in the system. This routine would require to shutdown all modules which have big contribution to power dissipation (e.g. low sides, high sides). This procedure has to be implemented in user software. Another possibility is to use the implemented hardware shutdown procedure. All power dissipation contributors will be automatically shutdown.

Main power dissipation contributors are:

- Low-side drivers
- High-side drivers
- LIN

1 ms after the indication the system will be set into sleep mode.

7.5.2 Power control unit registers

The power control unit (PCU) is fully controllable by the below listed SFR registers.

The registers are addressed wordwise.

7.5.2.1 Register overview - Power Control Unit for Power Modules registers (ascending offset address)

Table 58 Register overview - Power Control Unit for Power Modules registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCUPM_PCU_CTRL_STS	Power control unit control status register	0030_H	268

7.6 System control unit - power modules (SCUPM) register definition

Note: HS2 and MON5 are device variant specific. In devices featuring only HS1 the HS2_XXX bitfields can be ignored. In devices featuring only MON1-4 the HS MON5_XXX bitfields can be ignored. Writing to these bitfields has no effect.

7.6.1 Register address space - SCUPM

Table 59 Registers address space - SCUPM

Module	Base address	End address	Note
SCUPM	50006000_H	$50006FFF_H$	System Control Unit - Power Modules (SCU-PM)

7.6.2 Register overview - SCUPM (ascending offset address)

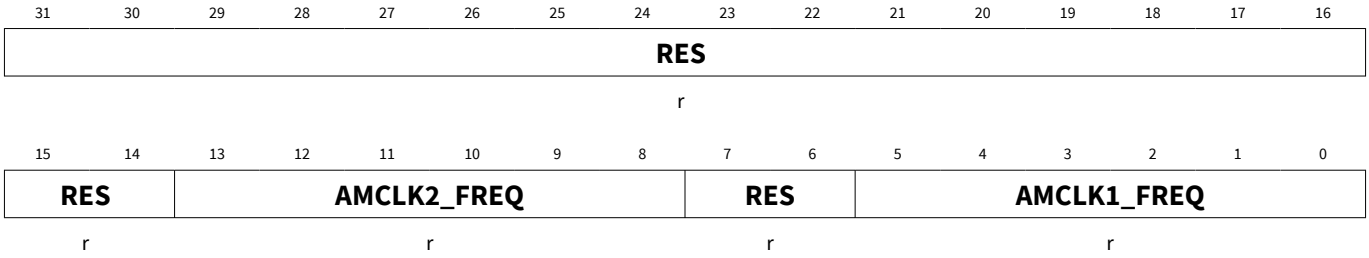
Table 60 Register overview - SCUPM (ascending offset address)

Short name	Long name	Offset address	Page number
SCUPM_AMCLK_FREQ_STS	Analog module clock frequency status register	0000_H	253
SCUPM_AMCLK_CTRL	Analog module clock control register	0004_H	254
SCUPM_AMCLK_TH_HYS	Analog module clock limit register	$000C_H$	255
SCUPM_SYS_ISCLR	System interrupt status clear register	0014_H	257
SCUPM_SYS_IS	System interrupt status register	0018_H	258
SCUPM_SYS_SUPPLY_IRQ_STS	System supply interrupt status register	$001C_H$	261
SCUPM_SYS_SUPPLY_IRQ_CTRL	System supply interrupt control register	0020_H	263
SCUPM_SYS_SUPPLY_IRQ_CLR	System supply interrupt status clear register	0024_H	265
SCUPM_SYS_IRQ_CTRL	System interrupt control register	0028_H	267
SCUPM_PCU_CTRL_STS	Power control unit control status register	0030_H	268
SCUPM_WDT1_TRIG	WDT1 watchdog control register	0034_H	269
SCUPM_STCALIB	System tick calibration register	$006C_H$	256

7.6.3 Analog module clock frequency status register

SCUPM_AMCLK_FREQ_STS Offset address: 0000_H

Analog module clock frequency status register RESET_TYPE_4 value: $0000\ 0000_H$



Field	Bits	Type	Description
AMCLK1_FREQ	5:0	r	**Current frequency of analog module clock system clock (MI_CLK)** 0.75 MHz × AMCLK1_FREQ
RES	7:6, 15:14, 31:16	r	**Reserved** Always read as 0.
AMCLK2_FREQ	13:8	r	**Current frequency of analog module clock 2 (TFILT_CLK)** 0.09375 MHz × AMCLK2_FREQ

7.6.4 Analog module clock control register

SCUPM_AMCLK_CTRL Offset address: 0004_H

Analog module clock control register RESET_TYPE_4 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
CLKWDT_PD_N	0	rw	**Clock watchdog power down** 0_B **DISABLE**: Clock watchdog disabled 1_B **ENABLE**: Clock watchdog enabled
RES	31:1	r	**Reserved** Always read as 0.

7.6.5 Analog module clock limit register

SCUPM_AMCLK_TH_HYS Offset address: $000C_H$

Analog module clock limit register RESET_TYPE_4 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					

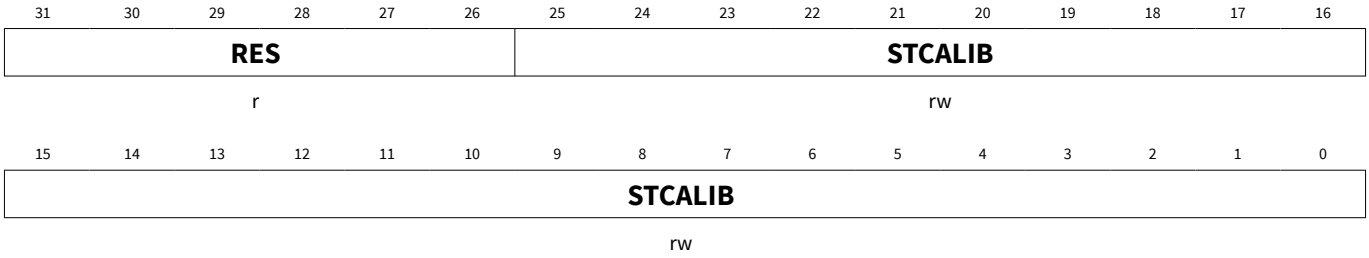
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AMCLK1_LOW_HYS		**AMCLK1_LOW_TH**						**AMCLK1_UP_HYS**		**AMCLK1_UP_TH**					
rw		rw						rw		rw					

Field	Bits	Type	Description
AMCLK1_UP_TH	5:0	rw	**Analog module clock 1 (MI_CLK) upper limit threshold** 0.75 MHz × AMCLK1_UP_TH
AMCLK1_UP_HYS	7:6	rw	**Analog module clock 1 (MI_CLK) upper hysteresis**
AMCLK1_LOW_TH	13:8	rw	**Analog module clock 1 (MI_CLK) lower limit threshold** 0.75 MHz × AMCLK1_LOW_TH
AMCLK1_LOW_HYS	15:14	rw	**Analog module clock 1 (MI_CLK) lower hysteresis**
AMCLK2_UP_TH	21:16	rw	**Analog module clock 2 (TFILT_CLK) upper limit threshold** 0.09375 MHz × AMCLK2_UP_TH
AMCLK2_UP_HYS	23:22	rw	**Analog module clock 2 (TFILT_CLK) upper hysteresis**
AMCLK2_LOW_TH	29:24	rw	**Analog module clock 2 (TFILT_CLK) lower limit threshold** 0.09375 MHz × AMCLK2_LOW_TH
AMCLK2_LOW_HYS	31:30	rw	**Analog module clock 2 (TFILT_CLK) lower hysteresis**

7.6.6 System tick calibration register

SCUPM_STCALIB Offset address: $006C_H$

System tick calibration register RESET_TYPE_4 value: $0000\ 0000_H$

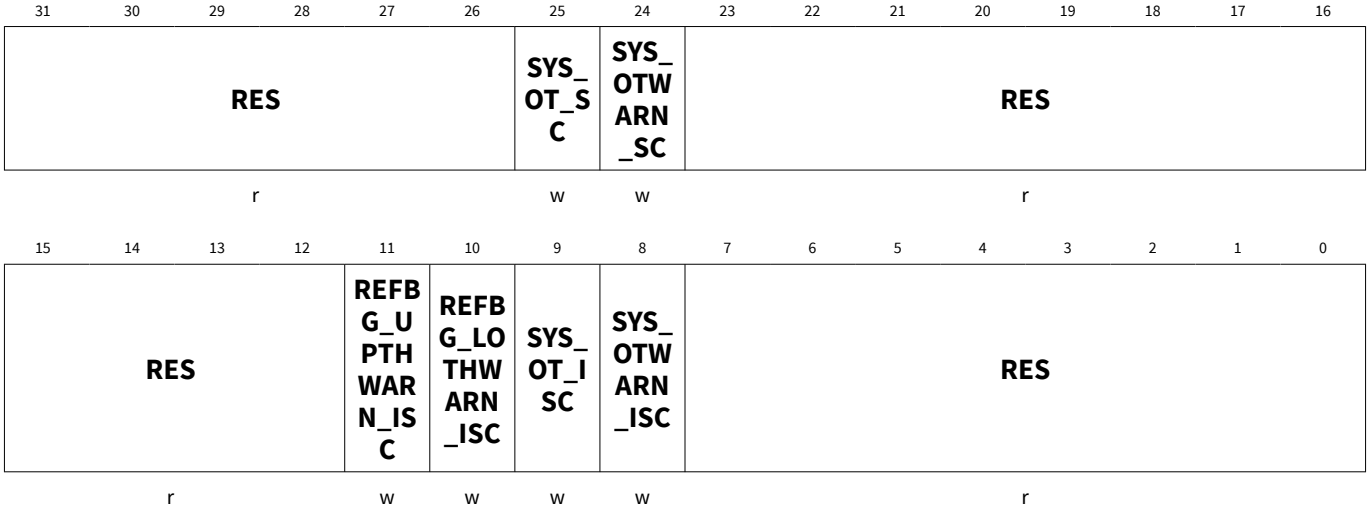


Field	Bits	Type	Description
STCALIB	25:0	rw	**System tick calibration** [25]: Noref [24]: Skew [23:0]: Reload value to use for 10 ms (100 Hz) timing STCALIB[23:0] = HCLK (in Hz) / 100 Hz − 1, e.g. 0x7A11F
RES	31:26	r	**Reserved** Always read as 0.

7.6.7 System interrupt status clear register

SCUPM_SYS_ISCLR Offset address: 0014_H

System interrupt status clear register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
RES	7:0, 23:12, 31:26	r	**Reserved** Always read as 0
SYS_OTWARN_ISC	8	w	**System overtemperature pre-warning status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
SYS_OT_ISC	9	w	**System overtemperature shutdown status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
REFBG_LOTHWARN_ISC	10	w	**8-bit ADC2 reference undervoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
REFBG_UPTHWARN_ISC	11	w	**8-bit ADC2 reference overvoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
SYS_OTWARN_SC	24	w	**System overtemperature pre-warning status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
SYS_OT_SC	25	w	**System overtemperature shutdown status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared

7.6.8 System interrupt status register

SCUPM_SYS_IS Offset address: 0018_H

System interrupt status register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			RES	RES		SYS_OT_STS	SYS_OTWARN_STS	RES	LIN_FAIL_STS	RES		HS2_FAIL_STS	HS1_FAIL_STS	LS2_FAIL_STS	LS1_FAIL_STS
r			r	r		rwhxr	rwhxr	r	r	r		r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			RES	REFBG_UPTHWARN_IS	REFBG_LOTHWARN_IS	SYS_OT_IS	SYS_OTWARN_IS	RES	LIN_FAIL_IS	RES		HS2_FAIL_IS	HS1_FAIL_IS	LS2_FAIL_IS	LS1_FAIL_IS
r			r	rwhxr	rwhxr	rwhxre	rwhxre	r	r	r		r	r	r	r

Field	Bits	Type	Description
LS1_FAIL_IS	0	r	**Low-side driver 1 fail interrupt status** *Note: This flag is an OR combination of LS1_OC_IS, LS1_OT_IS, LS1_OT_PREWARN_IS, and LS1_OL_IS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
LS2_FAIL_IS	1	r	**Low-side driver 2 fail interrupt status** *Note: This flag is an OR combination of LS2_OC_IS, LS2_OT_IS, LS2_OT_PREWARN_IS, and LS2_OL_IS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
HS1_FAIL_IS	2	r	**High-side driver 1 fail interrupt status** *Note: This flag is an OR combination of HS1_OC_IS, HS1_OT_IS, and HS1_OL_IS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
HS2_FAIL_IS	3	r	**High-side driver 2 fail interrupt status** *Note: This flag is an OR combination of HS2_OC_IS, HS2_OT_IS, and HS2_OL_IS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
RES	5:4, 7,	r	**Reserved** Always read as 0.

(table continues...)

(continued)

Field	Bits	Type	Description
	12, 15:13, 21:20, 23, 27:26, 28, 31:29		
LIN_FAIL_IS	6	r	**LIN fail interrupt status** *Note: This flag is an OR combination of LIN_IRQS.OC_IS and LIN_IRQS.OT_IS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
SYS_OTWARN_IS	8	rwhxre	**System overtemperature prewarning (ADC2, channel 6) interrupt status** 0_B **INACTIVE**: No interrupt status set 1_B **ACTIVE**: At least one interrupt status set
SYS_OT_IS	9	rwhxre	**System overtemperature shutdown (ADC2, channel 6) interrupt status** 0_B **INACTIVE**: No interrupt status set 1_B **ACTIVE**: At least one interrupt status set
REFBG_LOTHWARN_IS	10	rwhxr	**8-bit ADC2 reference undervoltage (ADC2, channel 3) interrupt status** 0_B **INACTIVE**: No interrupt status set 1_B **ACTIVE**: At least one interrupt status set
REFBG_UPTHWARN_IS	11	rwhxr	**8-bit ADC2 reference overvoltage (ADC2, channel 3) interrupt status** 0_B **INACTIVE**: No interrupt status set 1_B **ACTIVE**: At least one interrupt status set
LS1_FAIL_STS	16	r	**Low-side driver 1 fail status** *Note: This flag is an OR combination of LS1_OT_STS, LS1_OT_PREWARN_STS, and LS1_OL_STS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
LS2_FAIL_STS	17	r	**Low-side driver 2 fail status** *Note: This flag is an OR combination of LS2_OT_STS, LS2_OT_PREWARN_STS, and LS2_OL_STS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
HS1_FAIL_STS	18	r	**High-side driver 1 fail status** *Note: This flag is an OR combination of HS1_OT_STS, and HS1_OL_STS.*

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
HS2_FAIL_STS	19	r	**High-side driver 2 fail status** *Note: This flag is an OR combination of HS2_OT_STS and HS2_OL_STS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
LIN_FAIL_STS	22	r	**LIN fail status** *Note: This flag is the LIN_IRQS.OT_STS.* 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
SYS_OTWARN_STS	24	rwhxr	**System overtemperature pre-warning (ADC2, channel 6) status** 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set
SYS_OT_STS	25	rwhxr	**System overtemperature shutdown (ADC2, channel 6) status** 0_B **INACTIVE**: No status set 1_B **ACTIVE**: At least one status set

7.6.9 System supply interrupt status register

SCUPM_SYS_SUPPLY_IRQ_STS Offset address: $001C_H$

System supply interrupt status register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES						VDD_EXT_OV_STS	VDD_1V5_OV_STS	VDD_5V_OV_STS	VS_OV_STS	VBAT_OV_STS	VDD_EXT_UV_STS	VDD_1V5_UV_STS	VDD_5V_UV_STS	VS_UV_STS	VBAT_UV_STS
r						rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						VDD_EXT_OV_IS	VDD_1V5_OV_IS	VDD_5V_OV_IS	VS_OV_IS	VBAT_OV_IS	VDD_EXT_UV_IS	VDD_1V5_UV_IS	VDD_5V_UV_IS	VS_UV_IS	VBAT_UV_IS
r						rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre

Field	Bits	Type	Description
VBAT_UV_IS	0	rwhxre	**VBAT undervoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VS_UV_IS	1	rwhxre	**VS undervoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VDD5V_UV_IS	2	rwhxre	**VDDP undervoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VDD1V5_UV_IS	3	rwhxre	**VDDC undervoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VDDEXT_UV_IS	4	rwhxre	**VDDEXT undervoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VBAT_OV_IS	5	rwhxre	**VBAT overvoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VS_OV_IS	6	rwhxre	**VS overvoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VDD5V_OV_IS	7	rwhxre	**VDDP overvoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred

(table continues...)

(continued)

Field	Bits	Type	Description
VDD1V5_OV_IS	8	rwhxre	**VDDC overvoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
VDDEXT_OV_IS	9	rwhxre	**VDDEXT overvoltage interrupt status** 0_B **NO_INTERRUPT**: No undervoltage interrupt occurred 1_B **INTERRUPT**: Undervoltage interrupt occurred
RES	15:10, 31:26	r	**Reserved** Always read as 0.
VBAT_UV_STS	16	rwhxr	**VBAT undervoltage status** 0_B **NO_UNDERVOLTAGE**: No undervoltage occurred 1_B **UNDERVOLTAGE**: Undervoltage occurred
VS_UV_STS	17	rwhxr	**VS undervoltage status** 0_B **NO_UNDERVOLTAGE**: No undervoltage occurred 1_B **UNDERVOLTAGE**: Undervoltage occurred
VDD5V_UV_STS	18	rwhxr	**VDDP undervoltage status** 0_B **NO_UNDERVOLTAGE**: No undervoltage occurred 1_B **UNDERVOLTAGE**: Undervoltage occurred
VDD1V5_UV_STS	19	rwhxr	**VDDC undervoltage status** 0_B **NO_UNDERVOLTAGE**: No undervoltage occurred 1_B **UNDERVOLTAGE**: Undervoltage occurred
VDDEXT_UV_STS	20	rwhxr	**VDDEXT undervoltage status** 0_B **NO_UNDERVOLTAGE**: No undervoltage occurred 1_B **UNDERVOLTAGE**: Undervoltage occurred
VBAT_OV_STS	21	rwhxr	**VBAT overvoltage status** 0_B **NO_OVERVOLTAGE**: No overvoltage occurred 1_B **OVERVOLTAGE**: Overvoltage occurred
VS_OV_STS	22	rwhxr	**VS overvoltage status** 0_B **NO_OVERVOLTAGE**: No overvoltage occurred 1_B **OVERVOLTAGE**: Overvoltage occurred
VDD5V_OV_STS	23	rwhxr	**VDDP overvoltage status** 0_B **NO_OVERVOLTAGE**: No overvoltage occurred 1_B **OVERVOLTAGE**: Overvoltage occurred
VDD1V5_OV_STS	24	rwhxr	**VDDC overvoltage status** 0_B **NO_OVERVOLTAGE**: No overvoltage occurred 1_B **OVERVOLTAGE**: Overvoltage occurred
VDDEXT_OV_STS	25	rwhxr	**VDDEXT overvoltage status** 0_B **NO_OVERVOLTAGE**: No overvoltage occurred 1_B **OVERVOLTAGE**: Overvoltage occurred

7.6.10 System supply interrupt control register

SCUPM_SYS_SUPPLY_IRQ_CTRL Offset address: 0020_H

System supply interrupt control register RESET_TYPE_4 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						VDD EXT_ OV_I E	VDD 1V5_ OV_I E	VDD 5V_O V_IE	VS_O V_IE	VBAT _OV_ IE	VDD EXT_ UV_I E	VDD 1V5_ UV_I E	VDD 5V_U V_IE	VS_U V_IE	VBAT _UV_ IE
r						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
VBAT_UV_IE	0	rw	**VBAT undervoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VS_UV_IE	1	rw	**VS undervoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VDD5V_UV_IE	2	rw	**VDDP undervoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VDD1V5_UV_IE	3	rw	**VDDC undervoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VDDEXT_UV_IE	4	rw	**VDDEXT undervoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VBAT_OV_IE	5	rw	**VBAT overvoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VS_OV_IE	6	rw	**VS overvoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VDD5V_OV_IE	7	rw	**VDDP overvoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
VDD1V5_OV_IE	8	rw	**VDDC overvoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLED**: Interrupt is enabled
VDDEXT_OV_IE	9	rw	**VDDEXT overvoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
RES	31:10	r	**Reserved** Always read as 0

7.6.11 System supply interrupt status clear register

SCUPM_SYS_SUPPLY_IRQ_CLR Offset address: 0024_H

System supply interrupt status clear register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES						VDD EXT_ OV_S C	VDD 1V5_ OV_S C	VDD 5V_O V_SC	VS_O V_SC	VBAT _OV_ SC	VDD EXT_ UV_S C	VDD 1V5_ UV_S C	VDD 5V_U V_SC	VS_U V_SC	VBAT _UV_ SC
r						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
RES						VDD EXT_ OV_I SC	VDD 1V5_ OV_I SC	VDD 5V_O V_IS C	VS_O V_IS C	VBAT _OV_ ISC	VDD EXT_ UV_I SC	VDD 1V5_ UV_I SC	VDD 5V_U V_IS C	VS_U V_IS C	VBAT _UV_ ISC
r						w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
VBAT_UV_ISC	0	w	**VBAT undervoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VS_UV_ISC	1	w	**VS undervoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD5V_UV_ISC	2	w	**VDDP undervoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD1V5_UV_ISC	3	w	**VDDC undervoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDDEXT_UV_ISC	4	w	**VDDEXT undervoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VBAT_OV_ISC	5	w	**VBAT overvoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VS_OV_ISC	6	w	**VS overvoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD5V_OV_ISC	7	w	**VDDP overvoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared

(table continues...)

(continued)

Field	Bits	Type	Description
VDD1V5_OV_ISC	8	w	**VDDC overvoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDDEXT_OV_ISC	9	w	**VDDEXT overvoltage interrupt status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
RES	15:10, 31:26	r	**Reserved** Always read as 0
VBAT_UV_SC	16	w	**VBAT undervoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VS_UV_SC	17	w	**VS undervoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD5V_UV_SC	18	w	**VDDP undervoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD1V5_UV_SC	19	w	**VDDC undervoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDDEXT_UV_SC	20	w	**VDDEXT undervoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VBAT_OV_SC	21	w	**VBAT overvoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VS_OV_SC	22	w	**VS overvoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD5V_OV_SC	23	w	**VDDP overvoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDD1V5_OV_SC	24	w	**VDDC overvoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared
VDDEXT_OV_SC	25	w	**VDDEXT overvoltage status clear** 0_B **NO_CLEAR**: The interrupt status is not cleared 1_B **CLEAR**: The interrupt status is cleared

7.6.12 System interrupt control register

SCUPM_SYS_IRQ_CTRL Offset address: 0028_H

System interrupt control register RESET_TYPE_4 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				REFBG_UPTHWARN_IE	REFBG_LOTHWARN_IE	SYS_OT_IE	SYS_OTWARN_IE	RES							
r				rw	rw	rw	rw	r							

Field	Bits	Type	Description
RES	7:0, 31:12	r	**Reserved** Always read as 0
SYS_OTWARN_IE	8	rw	**System overtemperature warning interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
SYS_OT_IE	9	rw	**System overtemperature shutdown interrupt enable (leads to shutdown of system)** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
REFBG_LOTHWARN_IE	10	rw	**Reference voltage undervoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled
REFBG_UPTHWARN_IE	11	rw	**Reference voltage overvoltage interrupt enable** 0_B **DISABLED**: Interrupt is disabled 1_B **ENABLED**: Interrupt is enabled

7.6.13 Power control unit control status register

SCUPM_PCU_CTRL_STS Offset address: 0030_H

Power control unit control status register RESET_TYPE_4 value: $0EE3\ 7EF3_H$

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

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

Field	Bits	Type	Description
RES	0, 4:2, 5, 7:6, 16:9, 17, 27:18, 31:28	r	**Reserved** Always read as 0
CLKWDT_SD_DIS	1	rw	**Power modules clock watchdog shutdown disable** 0_B **ENABLE**: Power devices will be switched off when clock watchdog occurs 1_B **DISABLE**: Power devices will not be shut down when clock watchdog occurs
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

7.6.14 WDT1 watchdog control register

SCUPM_WDT1_TRIG Offset address: 0034_H

WDT1 watchdog control register RESET_TYPE_3 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								**SOWCONF**		**WDP_SEL**					
r								rw		rw					

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 (1*16) … $3F_H$ **WP_63**: Watchdog period 1008 ms (63*16)
SOWCONF	7:6	rw	**Short open window configuration** 00_B **DIS**: Short open windows disabled Writing 00_H to the WDT_TRIG register will cause a reset. 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
RES	31:8	r	**Reserved** Always read as 0.

8 Arm® Cortex®-M0 core

8.1 Features

The key features of the Arm® Cortex®-M0 implemented are listed below.

Processor core – a low gate count core, with low latency interrupt processing:

- Thumb® + Thumb-2® instruction set
- Banked stack pointer (SP) only
- 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 v6-M style
- Arm®v6 unaligned accesses
- Systick (typical 1 ms)

Nested vectored interrupt controller (NVIC) closely integrated with the processor core to achieve low latency interrupt processing:

- External interrupts, configurable from 1 to 24
- 7 interrupt priority registers for levels from 0 up to 192 in steps of 64
- Dynamic re-prioritization of interrupts
- Priority grouping. This enables selection of preempting interrupt levels and non preempting 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

8.2 Introduction

The Arm® Cortex®-M0 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®-M0 processor implements the Thumb®-2 instruction set architecture. With the optimized feature set the Arm® Cortex®-M0 delivers 32-bit performance in an application space that is usually associated with 8-bit and 16-bit microcontrollers.

8.2.1 Block diagram

The following figure shows the functional blocks of the Arm® Cortex®-M0.

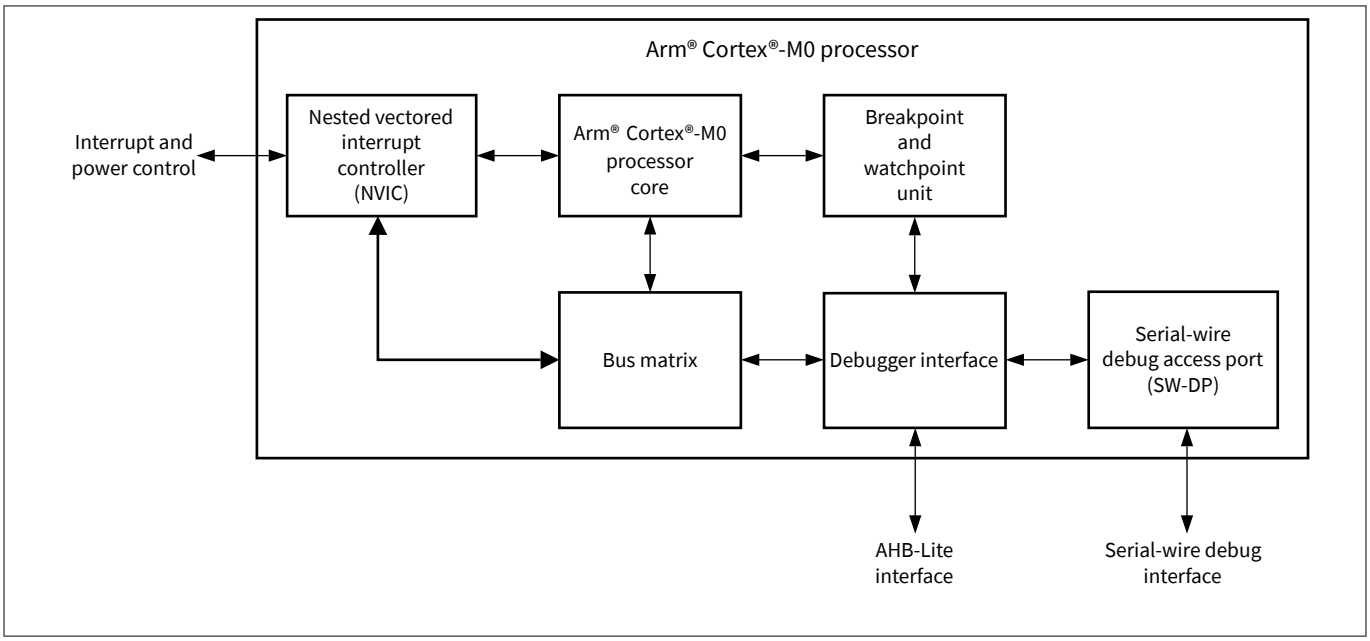


Figure 42 Arm® Cortex®-M0 block diagram

8.3 Functional description

8.3.1 Processor 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 registers

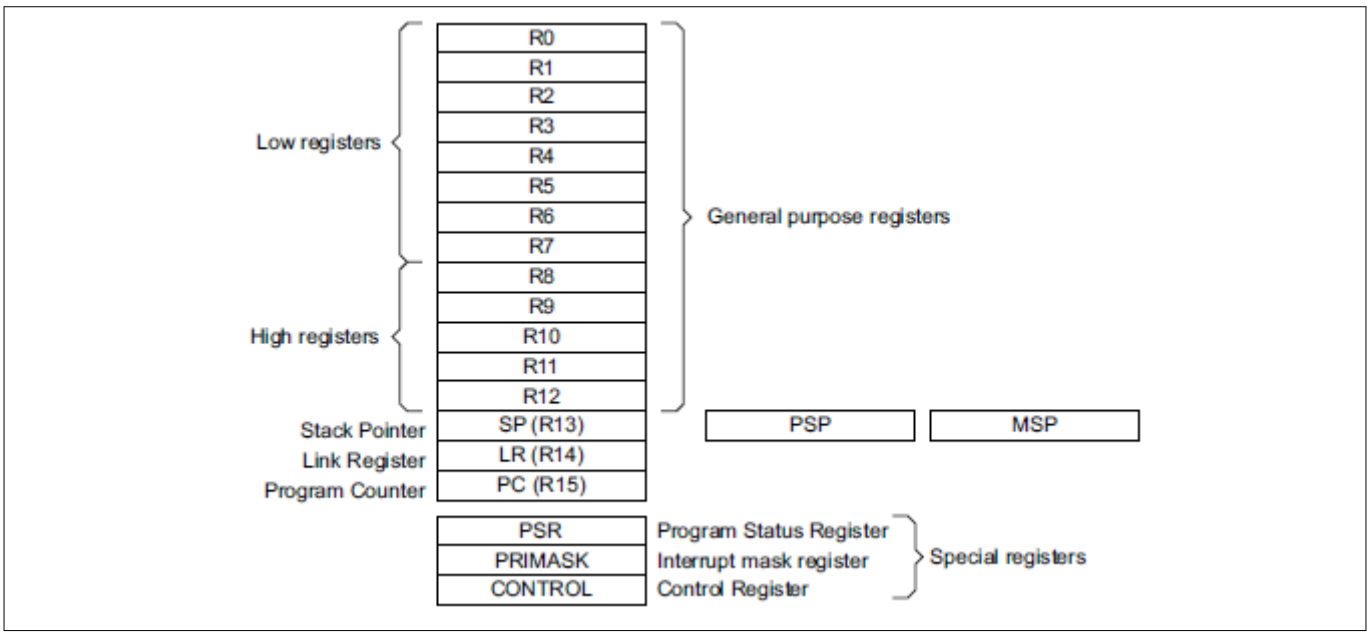


Figure 43 Processor register set

8.3.1.1 General-purpose registers

The general-purpose registers R0-R12 are 32-bit registers for data operations.

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 registers

The special-purpose registers have the following functions:

- Program status register
  Register PSR is the program status register.
- Interrupt mask register

Register PRIMASK is the interrupt mask register.

- Control register
  Register CONTROL is the control register.

8.3.2 Processor (CPU) register definition

The processor has the following listed 32-bit registers that control functionality.

Note: HS2 and MON5 are device variant specific. In devices featuring only HS1 the HS2_XXX bitfields can be ignored. In devices featuring only MON1-4 the HS MON5_XXX bitfields can be ignored. Writing to these bitfields has no effect.

The registers are addressed wordwise.

8.3.2.1 Register address space - CPU

Table 61 Registers address space - CPU

Module	Base address	End address	Note
CPU	$E000E000_H$	$E000EFFF_H$	Arm® Cortex®-M0 Core SCS (system control space), Systick, NVIC processor registers

8.3.2.2 Register overview - CPU (ascending offset address)

Table 62 Register overview - CPU (ascending offset address)

Short name	Long name	Offset address	Page number
CPU_SYSTICK_CSR	SysTick control and status register	0010_H	275
CPU_SYSTICK_RVR	SysTick reload value register	0014_H	276
CPU_SYSTICK_CVR	SysTick current value register	0018_H	277
CPU_SYSTICK_CALIB	SysTick calibration value register	$001C_H$	278
CPU_NVIC_ISER	Interrupt set-enable register	0100_H	279
CPU_NVIC_ICER	Interrupt clear-enable register	0180_H	282
CPU_NVIC_ISPR	Interrupt set-pending register	0200_H	285
CPU_NVIC_ICPR	Interrupt clear-pending register	0280_H	288
CPU_NVIC_IPR0	Interrupt priority 0 register	0400_H	291
CPU_NVIC_IPR1	Interrupt priority 1 register	0404_H	292
CPU_NVIC_IPR2	Interrupt priority 2 register	0408_H	293
CPU_NVIC_IPR3	Interrupt priority 3 register	$040C_H$	294
CPU_NVIC_IPR4	Interrupt priority 4 register	0410_H	295
CPU_NVIC_IPR5	Interrupt priority 5 register	0414_H	296
CPU_CPUID	CPU ID base register	$0D00_H$	297
CPU_ICSR	Interrupt control and state register	$0D04_H$	298
CPU_AIRCR	Application interrupt/reset control register	$0D0C_H$	300
CPU_SCR	System control register	$0D10_H$	301
CPU_CCR	Configuration control register	$0D14_H$	302
CPU_SHPR2	System handler priority 2 register	$0D1C_H$	303
CPU_SHPR3	System handler priority 3 register	$0D20_H$	304

8.3.2.3 SysTick control and status register

CPU_SYSTICK_CSR Offset address: 0010_H

SysTick control and status register RESET_TYPE_3 value: $0000\ 0000_H$

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

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
ENABLE	0	rw	**Enable** 0_B **DISABLE**: Counter disabled 1_B **ENABLE**: Counter enabled
TICKINT	1	rw	**TICKINT** Enables SysTick exception request. 0_B **DISABLE**: Counting down to 0 does not assert the SysTick exception request 1_B **ENABLE**: Counting down to 0 asserts the SysTick exception request
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 (from fSYS/4) 1_B **HCLK**: Core clock (from fSYS)
RES	15:3, 31:17	r	**Reserved**
COUNTFLAG	16	r	**Count flag** Returns 1 if timer counted to 0 since the last read of this register.

8.3.2.4 SysTick reload value register

Calculating the RELOAD value

The RELOAD value can be any value in the range 00000001_H to $00FFFFFF_H$. You can program a value of 0, but this has no effect because the SysTick exception request and COUNTFLAG are activated when counting from 1 to 0.

To generate a multi-shot timer with a period of N processor clock cycles, use a RELOAD value of N-1. For example, if the SysTick interrupt is required every 100 clock pulses, set RELOAD to 99.

CPU_SYSTICK_RVR Offset address: 0014_H

SysTick reload value register RESET_TYPE_3 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
RELOAD	23:0	rw	**Reload** Value to load into the SysTick current value register when the counter is enabled and when it reaches 0, see "Calculating the RELOAD value" above.
RES	31:24	r	**Reserved**

8.3.2.5 SysTick current value register

CPU_SYSTICK_CVR Offset address: 0018_H

SysTick current value register RESET_TYPE_3 value: $00XX\ XXXX_H$

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

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

Field	Bits	Type	Description
CURRENT	23:0	rw	**Current** Reads return the current value of the SysTick counter. A write of any value clears the field to 0, and also clears the SYST_CSR.COUNTFLAG bit to 0.
RES	31:24	r	**Reserved**

8.3.2.6 SysTick calibration value register

CPU_SYSTICK_CALIB Offset address: $001C_H$

SysTick calibration value register RESET_TYPE_3 value: $X0XX\ XXXX_H$

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
TENMS	23:0	r	**Tenms** Indicates calibration value is not known. If calibration information is not known, calculate the calibration value required from the frequency of the processor clock or external clock. Reads as 0.
RES	29:24	r	**Reserved**
SKEW	30	r	**Skew** Calibration value for the 10 ms inexact timing is not known because TENMS is not known. This can affect the suitability of SysTick as a software real time clock. Reads as 0_b.
NOREF	31	r	**No reference clock** Indicates that no separate reference clock is provided. Reads as 0_b.

8.3.2.7 Interrupt set-enable register

CPU_NVIC_ISER Offset address: 0100_H

Interrupt set-enable register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								Int_POR T2	Int_MON	Int_DU	Int_HS2	Int_HS1	Int_LS2	Int_LS1	RES
r								rw	rw	rw	rw	rw	rw	rw	r

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

Field	Bits	Type	Description
Int_GPT1	0	rw	**Interrupt set for GPT1** 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_ADC2	2	rw	**Interrupt set for MU, ADC2** 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_CCU6SR0	4	rw	**Interrupt set for CCU6 SR0** 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_CCU6SR2	6	rw	**Interrupt set for CCU6 SR2** 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
Int_SSC1	8	rw	**Interrupt set for SSC1**

(table continues...)

(continued)

Field	Bits	Type	Description
			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_UART1	10	rw	**Interrupt set for UART1** 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_EXINT0	12	rw	**Interrupt set for external Int 0** 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_WAKEUP	14	rw	**Interrupt set for WAKEUP** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt
RES	15, 16, 31:24	r	**Reserved for future use**
Int_LS1	17	rw	**Interrupt set for LS1** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt
Int_LS2	18	rw	**Interrupt set for LS2** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt
Int_HS1	19	rw	**Interrupt set for HS1** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt
Int_HS2	20	rw	**Interrupt set for HS2** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt
Int_DU	21	rw	**Interrupt set for differential unit** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt
Int_MON	22	rw	**Interrupt set for MON** 0_B **DISABLED**: No effect on write

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLE**: Enables the associated interrupt
Int_PORT2	23	rw	**Interrupt set for PORT2** 0_B **DISABLED**: No effect on write 1_B **ENABLE**: Enables the associated interrupt

8.3.2.8 Interrupt clear-enable register

CPU_NVIC_ICER Offset address: 0180_H

Interrupt clear-enable register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								Int_POR T2	Int_MON	Int_DU	Int_HS2	Int_HS1	Int_LS2	Int_LS1	RES
r								rw	rw	rw	rw	rw	rw	rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	Int_WAK EUP	Int_E XINT 1	Int_E XINT 0	Int_UAR T2	Int_UAR T1	Int_S SC2	Int_S SC1	Int_CCU6 SR3	Int_CCU6 SR2	Int_CCU6 SR1	Int_CCU6 SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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
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
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_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_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_CCU6SR1	5	rw	**Interrupt clear for CCU6 SR1** 0_B **DISABLE**: On reads the associated interrupt is disabled, no effect on write

(table continues...)

(continued)

Field	Bits	Type	Description
			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_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_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_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_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
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_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_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_WAKEUP	14	rw	**Interrupt clear for WAKEUP**

(table continues...)

(continued)

Field	Bits	Type	Description
			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
RES	15, 16, 31:24	r	**Reserved for future use**
Int_LS1	17	rw	**Interrupt clear for LS1** 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_LS2	18	rw	**Interrupt clear for LS2** 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_HS1	19	rw	**Interrupt clear for HS1** 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_HS2	20	rw	**Interrupt clear for HS2** 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_DU	21	rw	**Interrupt clear for differential unit** 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_MON	22	rw	**Interrupt clear for MON** 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_PORT2	23	rw	**Interrupt Clear for PORT2** 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

8.3.2.9 Interrupt set-pending register

CPU_NVIC_ISPR Offset address: 0200_H

Interrupt set-pending register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								Int_POR T2	Int_MON	Int_DU	Int_HS2	Int_HS1	Int_LS2	Int_LS1	RES
r								rw	rw	rw	rw	rw	rw	rw	r

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

Field	Bits	Type	Description
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
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_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_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_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_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_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

(table continues...)

(continued)

Field	Bits	Type	Description
			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_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_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
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_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_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_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_WAKEUP	14	rw	**Interrupt set pending for WAKEUP** 0_B **Not_pending**: On reads the associated interrupt is not pending, no effect on writes 1_B **Pending**: The associated interrupt is pending
RES	15, 16, 31:24	r	**Reserved for future use**
Int_LS1	17	rw	**Interrupt set pending for LS1** 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_LS2	18	rw	**Interrupt set pending for LS2**

(table continues...)

(continued)

Field	Bits	Type	Description
			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_HS1	19	rw	**Interrupt set pending for HS1** 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_HS2	20	rw	**Interrupt set pending for HS2** 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_DU	21	rw	**Interrupt set pending for differential unit** 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_MON	22	rw	**Interrupt set pending for MON** 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_PORT2	23	rw	**Interrupt set pending for PORT2** 0_B **Not_pending**: On reads the associated interrupt is not pending, no effect on writes 1_B **Pending**: The associated interrupt is pending

8.3.2.10 Interrupt clear-pending register

CPU_NVIC_ICPR Offset address: 0280_H

Interrupt clear-pending register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								Int_POR T2	Int_MON	Int_DU	Int_HS2	Int_HS1	Int_LS2	Int_LS1	RES
r								rw	rw	rw	rw	rw	rw	rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	Int_WAK EUP	Int_E XINT 1	Int_E XINT 0	Int_UAR T2	Int_UAR T1	Int_S SC2	Int_S SC1	Int_CCU6 SR3	Int_CCU6 SR2	Int_CCU6 SR1	Int_CCU6 SR0	Int_ADC1	Int_ADC2	Int_GPT2	Int_GPT1
r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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
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
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_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_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_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

(table continues...)

(continued)

Field	Bits	Type	Description
			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_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_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_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_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
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_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_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_WAKEUP	14	rw	**Interrupt clear pending for WAKEUP**

(table continues...)

(continued)

Field	Bits	Type	Description
			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
RES	15, 16, 31:24	r	**Reserved for future use**
Int_LS1	17	rw	**Interrupt clear pending for LS1** 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_LS2	18	rw	**Interrupt clear pending for LS2** 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_HS1	19	rw	**Interrupt clear pending for HS1** 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_HS2	20	rw	**Interrupt clear pending for HS2** 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_DU	21	rw	**Interrupt clear pending for differential unit** 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_MON	22	rw	**Interrupt clear pending for MON** 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_PORT2	23	rw	**Interrupt clear pending for PORT2** 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

8.3.2.11 Interrupt priority 0 register

CPU_NVIC_IPR0 Offset address: 0400_H

Interrupt priority 0 register RESET_TYPE_3 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
RES	5:0, 13:8, 21:16, 29:24	r	**Reserved**
PRI_GPT1	7:6	rw	**Priority for GPT1**
PRI_GPT2	15:14	rw	**Priority for GPT2**
PRI_ADC2	23:22	rw	**Priority for MU, ADC2**
PRI_ADC1	31:30	rw	**Priority for ADC1**

8.3.2.12 Interrupt priority 1 register

CPU_NVIC_IPR1 Offset address: 0404_H

Interrupt priority 1 register RESET_TYPE_3 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
RES	5:0, 13:8, 21:16, 29:24	r	**Reserved**
PRI_CCU6SR0	7:6	rw	**Priority for CCU6 SR0**
PRI_CCU6SR1	15:14	rw	**Priority for CCU6 SR1**
PRI_CCU6SR2	23:22	rw	**Priority for CCU6 SR2**
PRI_CCU6SR3	31:30	rw	**Priority for CCU6 SR3**

8.3.2.13 Interrupt priority 2 register

CPU_NVIC_IPR2 Offset address: 0408_H

Interrupt priority 2 register RESET_TYPE_3 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
RES	5:0, 13:8, 21:16, 29:24	r	**Reserved**
PRI_SSC1	7:6	rw	**Priority for CCU6 SSC1**
PRI_SSC2	15:14	rw	**Priority for CCU6 SSC2**
PRI_UART1	23:22	rw	**Priority for CCU6 UART1**
PRI_UART2	31:30	rw	**Priority for CCU6 UART2**

8.3.2.14 Interrupt priority 3 register

CPU_NVIC_IPR3 Offset address: $040C_H$

Interrupt priority 3 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		**RES**						**PRI_WAKEUP**		**RES**					
r		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
RES	5:0, 13:8, 21:16, 29:24, 31:30	r	**Reserved**
PRI_EXINT0	7:6	rw	**Priority for external Int 0**
PRI_EXINT1	15:14	rw	**Priority for external Int 1**
PRI_WAKEUP	23:22	rw	**Priority for WAKEUP**

8.3.2.15 Interrupt priority 4 register

CPU_NVIC_IPR4 Offset address: 0410_H

Interrupt priority 4 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_HS1		**RES**						**PRI_LS2**		**RES**					
rw		r						rw		r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_LS1		**RES**						**RES**		**RES**					
rw		r						r		r					

Field	Bits	Type	Description
RES	5:0, 7:6, 13:8, 21:16, 29:24	r	**Reserved**
PRI_LS1	15:14	rw	**Priority for LS1**
PRI_LS2	23:22	rw	**Priority for LS2**
PRI_HS1	31:30	rw	**Priority for HS1**

8.3.2.16 Interrupt priority 5 register

CPU_NVIC_IPR5 Offset address: 0414_H

Interrupt priority 5 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRI_PORT2		**RES**						**PRI_MON**		**RES**					
rw		r						rw		r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_DU		**RES**						**PRI_HS2**		**RES**					
rw		r						rw		r					

Field	Bits	Type	Description
RES	5:0, 13:8, 21:16, 29:24	r	**Reserved**
PRI_HS2	7:6	rw	**Priority for HS2**
PRI_DU	15:14	rw	**Priority for differential unit**
PRI_MON	23:22	rw	**Priority for MON**
PRI_PORT2	31:30	rw	**Priority for PORT2**

8.3.2.17 CPU ID base register

CPU_CPUID

CPU ID base register

Offset address: $0D00_H$

RESET_TYPE_3 value: $410C\ C200_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IMPLEMENTER								VARIANT				CONSTANT			
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
REVISION	3:0	r	**Revision number** Implementation defined.
PARTNO	15:4	r	**Part number** Implementation defined.
CONSTANT	19:16	r	**Constant** Defines the architecture of the processor.
VARIANT	23:20	r	**Variant number** Implementation defined.
IMPLEMENTER	31:24	r	**Implementer code** Assigned by Arm®. Read as 41_H for a processor implemented by Arm®.

8.3.2.18 Interrupt control and state register

CPU_ICSR Offset address: $0D04_H$

Interrupt control and state register RESET_TYPE_3 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**		**ISRPENDING**	**RES**				**VECTPENDING**	
rw	r		rw	w	rw	w	r		r	r				r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VECTPENDING				**RES**						**VECTACTIVE**					
r				r						r					

Field	Bits	Type	Description
VECTACTIVE	5:0	r	**VECTACTIVATE** Contains the active exception number. Nonzero is the exception number of the currently active exception. This is the same value as IPSR bits 5:0. *Note: Subtract 16 from this value to obtain the CMSIS IRQ number that identifies the corresponding bit in the interrupt clear-enable, set-enable, clear-pending, set-pending, and priority register.* When you write to the ICSR, the effect is unpredictable if you: • Write 1 to the PENDSVSET bit and write 1 to the PENDSVCLR bit. • Write 1 to the PENDSTSET bit and write 1 to the PENDSTCLR bit. 00_H **THREAD**: Thread mode
RES	11:6, 21:18, 24:23, 30:29	r	**Reserved**
VECTPENDING	17:12	r	**VECTPENDING** Indicates the exception number of the highest priority pending enabled exception. Nonzero is the exception number of the highest priority pending enables exception. 00_H **NOT_PENDING**: No pending exceptions
ISRPENDING	22	r	**Interrupt pending flag** Excluding NMI and faults. 0_B **NOT_PENDING**: Interrupt not pending 1_B **PENDING**: Interrupt is pending
PENDSTCLR	25	w	**SysTick exception clear pending** Removes the pending status of the SysTick exception.

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: This bit is write-only. On a register read its value is unknown. 0_B **NO_EFFECT**: No effect 1_B **REMOVE**: Removes the pending state from the SysTick exception
PENDSTSET	26	rw	**SysTick exception set pending** On writes, sets the SysTick exception as pending. On reads, indicates the current state of the exception. 0_B **NOT_PENDING**: On writes, has no effect. On reads, SysTick exception is not pending 1_B **PENDING**: On writes, changes SysTick exception state to pending. On reads, SysTick exception is pending
PENDSVCLR	27	w	**PendSV clear pending** Removes the pending status of the PendSV exception. 0_B **NO_EFFECT**: No effect 1_B **CLEAR**: Remove pending state from the PENDSV exception
PENDSVSET	28	rw	**PendSV set pending** On writes, sets the PendSV exception as pending. On reads, indicates the current state of the exception. *Note: Writing 1 to this bit is the only way to set the PenDSV exception state to pending.* 0_B **NOT_PENDING**: On writes, has no effect. On reads, PendSV exception is not pending 1_B **PENDING**: On writes, changes PendSV exception state to pending. On reads, PendSV is pending
NMIPENDSET	31	rw	**NMI set pending** On writes, makes the NMI exception state pending. On reads, indicates the state of the exception. *Note: Because NMI is the highest-priority exception, normally the processor enters the NMI exception handler as soon as it detects a write of 1 to this bit. Entering the handler then clears this bit to 0. This means a read of this bit by the NMI exception handler returns 1 only if the NMI signal is reasserted while the processor is executing that handler.* 0_B **NOT_PENDING**: On writes, has no effect. On reads, NMI exception is not pending 1_B **PENDING**: On writes, changes the NMI exception state to pending. On reads, NMI exception is pending

8.3.2.19 Application interrupt/reset control register

CPU_AIRCR Offset address: $0D0C_H$

Application interrupt/reset control register RESET_TYPE_3 value: $FA05\ 0000_H$

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

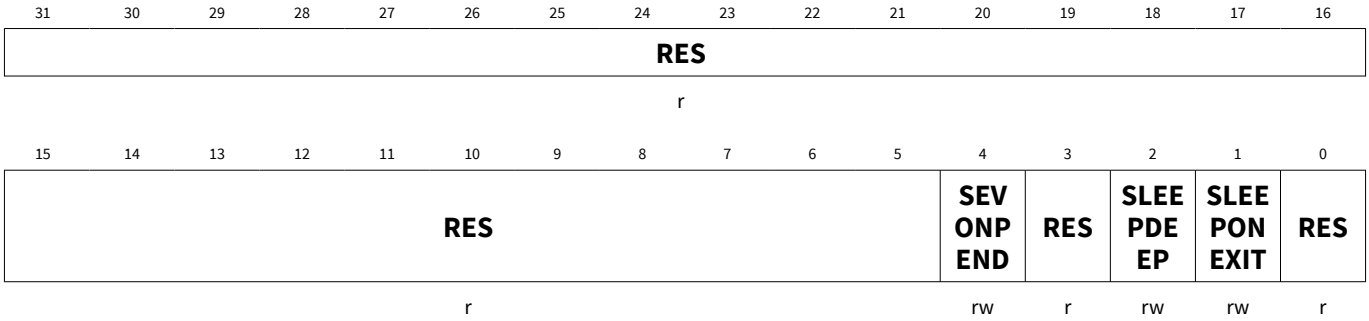
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENDIANNESS	**RES**												**SYSRESETREQ**	**VECTCLRACTIVE**	**RES**
r	r												w	w	r

Field	Bits	Type	Description
RES	0, 14:3	r	**Reserved**
VECTCLRACTIVE	1	w	**VECTCLRACTIVE** Reserved for debug use. This bit reads as 0_B. *Note: When writing to this register you must write to this bit, otherwise the behavior is unpredictable.*
SYSRESETREQ	2	w	**System reset request** This bit reads as 0_B. 0_B **NO_EFFECT**: No effect 1_B **RESET**: Request a system level reset
ENDIANNESS	15	r	**Data endianness** 0_B **LITTLE_ENDIAN**: Little endian 1_B **BIG_ENDIAN**: Big endian
VECTKEY	31:16	rw	**Vector key** Register writes must write $FA05_H$ to this field, otherwise the write is ignored. On reads, returns Unknown.

8.3.2.20 System control register

CPU_SCR Offset address: $0D10_H$

System control register RESET_TYPE_3 value: $0000\ 0000_H$

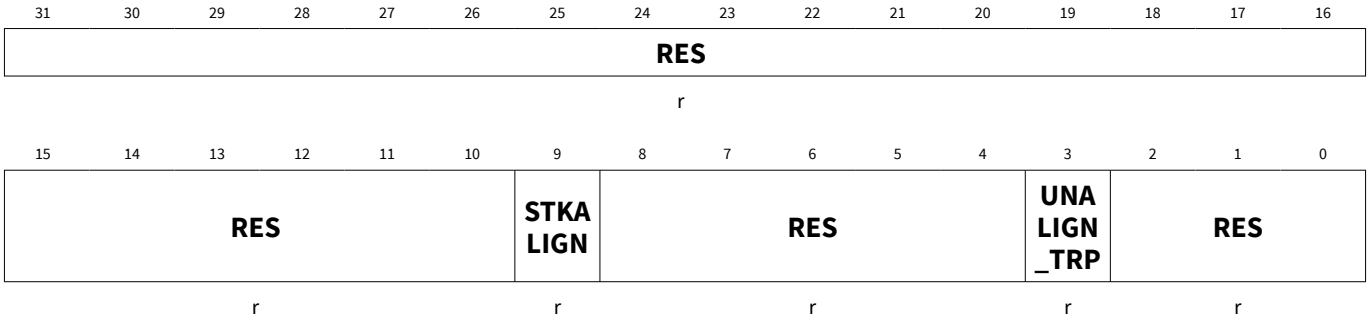


Field	Bits	Type	Description
RES	0, 3, 31:5	r	**Reserved**
SLEEPONEXIT	1	rw	**Sleep on exit** Indicates sleep-on-exit when returning from Handler mode to Thread mode. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application. 0_B **NO_SLEEP**: Do not sleep when returning to Thread mode 1_B **SLEEP**: Enter sleep or deep sleep on return from an ISR to Thread mode
SLEEPDEEP	2	rw	**Sleep deep** Controls whether the processor uses sleep or deep sleep as its low power mode. 0_B **SLEEP**: Sleep 1_B **DEEP_SLEEP**: Deep sleep
SEVONPEND	4	rw	**Send event on pending bit** When an event or interrupt enters pending state, the event signal wakes up the processor from WFE. If the processor is not waiting for an event, the event is registered and affects the next WFE. The processor also wakes up on execution of an SEV instruction or an external event. 0_B **SOME**: Only enabled interrupts or events can wake-up the processor, disabled interrupts are excluded 1_B **ALL**: Enabled events and all interrupts, including disabled interrupts, can wake up the processor

8.3.2.21 Configuration control register

CPU_CCR Offset address: $0D14_H$

Configuration control register RESET_TYPE_3 value: $0000\ 0208_H$

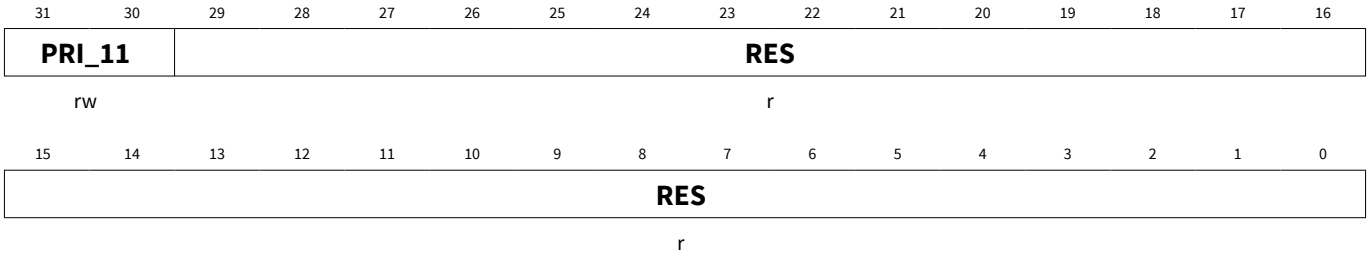


Field	Bits	Type	Description
RES	2:0, 8:4, 31:10	r	**Reserved**
UNALIGN_TRP	3	r	**UNALIGN_TRP** Indicates that all unaligned accesses generate a Hardfault. Always reads as 1_B.
STKALIGN	9	r	**STKALIGN** Always reads as 1_B, indicates 8-byte stack alignment on exception entry. On exception entry, the processor uses bit[9] of the stacked PSR to indicate the stack alignment. On return from the exception it uses this stacked bit to restore the correct stack alignment.

8.3.2.22 System handler priority 2 register

CPU_SHPR2 Offset address: $0D1C_H$

System handler priority 2 register RESET_TYPE_3 value: $0000\ 0000_H$

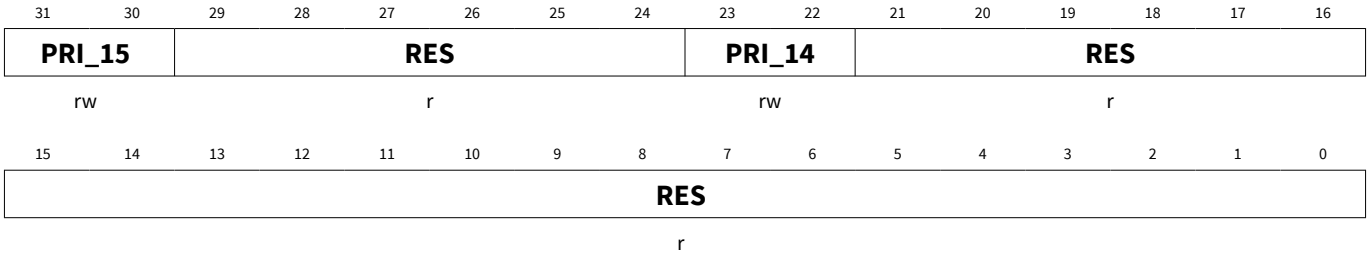


Field	Bits	Type	Description
RES	29:0	r	**Reserved**
PRI_11	31:30	rw	**Priority of system handler 11, SVCall**

8.3.2.23 System handler priority 3 register

CPU_SHPR3 Offset address: $0D20_H$

System handler priority 3 register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
RES	21:0, 29:24	r	**Reserved**
PRI_14	23:22	rw	**Priority of system handler 14, PendSV**
PRI_15	31:30	rw	**Priority of system handler 15, SysTick**

8.4 Instruction set summary

This chapter provides the instruction set. The following table 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.

The following table 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 63 Instruction set summary

Operation	Description	Mnemonic	Cycles (without wait states)
Move	Register	MOV Rd, Rm	1
Add	Add	ADD Rd, Rn, <op2>	1
	Add with carry	ADCS Rd, Rn, Rm	1
ADR	Address to register	ADR Rd, <label>	1
Subtract	Subtract	SUB Rd, Rn, <op2>	1
	Subtract with carry	SBCS Rd, Rn, Rm	1
	Reverse	RSBS Rd, Rn, #0	1
Multiply	Multiply, 32-bit result	MULS Rd, Rn, Rm	1
Compare	Compare	CMP Rn, <op2>	1
	Negative	CMN Rn, Rm	1
Logical	AND bitwise	ANDS Rd, Rn, <op2>	1
	Exclusive OR	EORS Rd, Rn, Rm	1
	OR	ORRS Rd, Rn, Rm	1
	Bit clear	BICS Rd, Rn, <op2>	1
	Move NOT bitwise	MVNS Rd, Rm	1
	AND test	TST Rn, Rm	1
Shift	Logical shift left	LSLS Rd, Rn, #<imm>	1
	Logical shift left	LSLS Rd, Rn, Rs	1
	Logical shift right	LSRS Rd, Rn, #<imm>	1
	Logical shift right	LSRS Rd, Rn, Rs	1

(table continues...)

Table 63 (continued) Instruction set summary

Operation	Description	Mnemonic	Cycles (without wait states)
	Arithmetic shift right	ASRS Rd, Rn, #<imm>	1
	Arithmetic shift right	ASRS Rd, Rn, Rs	1
Rotate	Rotate right	ROR Rd, Rn, Rs	1
Load	Word	LDR Rt, [Rn, <op2>]	2[1]
	Halfword	LDRH Rt, [Rn, <op2>]	2[1]
	Byte	LDRB Rt, [Rn, <op2>]	2[1]
	Signed halfword	LDRSH Rt, [Rn, <op2>]	2[1]
	Signed byte	LDRSB Rt, [Rn, <op2>]	2[1]
	Register from PC relative address	LDR Rt, label	2[1]
	Multiple register, increment after	LDM Rn, {<reglist>}	1 + N
Store	Word	STR Rt, [Rn, <op2>]	2[1]
	Halfword	STRH Rt, [Rn, <op2>]	2[1]
	Byte	STRB Rt, [Rn, <op2>]	2[1]
	Multiple register, increment after	STM Rn, {<reglist>}	1 + N
Push	Push registers onto stack	PUSH {<reglist>}	1 + N
Pop	Pop registers from stack	POP {<reglist>}	1 + N
Branch	Conditional	B <cc> <label>	1 or 1 + P[2]
	Unconditional	B <label>	1 + P
	With link	BL <label>	1 + P
	Indirect	BX Rm	1 + P
	Indirect with link	BLX Rm	1 + P
State change	Supervisor call	SVC #<imm>	–
	Disable interrupts	CPSID i	1 or 2
	Enable interrupts	CPSIE i	1 or 2
	Move to general register from special register	MRS Rd, <specreg>	1 or 2
	Move to special register from general register	MSR <specreg>, Rn	1 or 2
	Breakpoint	BKPT #<imm>	–
Extend	Signed halfword to word	SXTH Rd, Rm	1
	Signed byte to word	SXTB Rd, Rm	1
	Unsigned halfword	UXTH Rd, Rm	1
	Unsigned byte	UXTB Rd, Rm	1
Bit field	Clear	BICS Rd, Rn, Rm	1

(table continues...)

Table 63 (continued) Instruction set summary

Operation	Description	Mnemonic	Cycles (without wait states)
Reverse	Bytes in word	REV Rd, Rm	1
	Bytes in both halfwords	REV16 Rd, Rm	1
	Signed bottom halfword	REVSH Rd, Rm	1
	Subtract	RSBS Rd, Rn, #0	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	1 + B

1) Neighboring load and store single instructions can pipeline their address and data phases. This enables these instructions to complete in a singleexecution cycle.
2) Conditional branch completes in a single cycle if the branch is not taken.

9 Address space organization

The embedded Arm® Cortex®-M0 MCU offers the following address space organization:

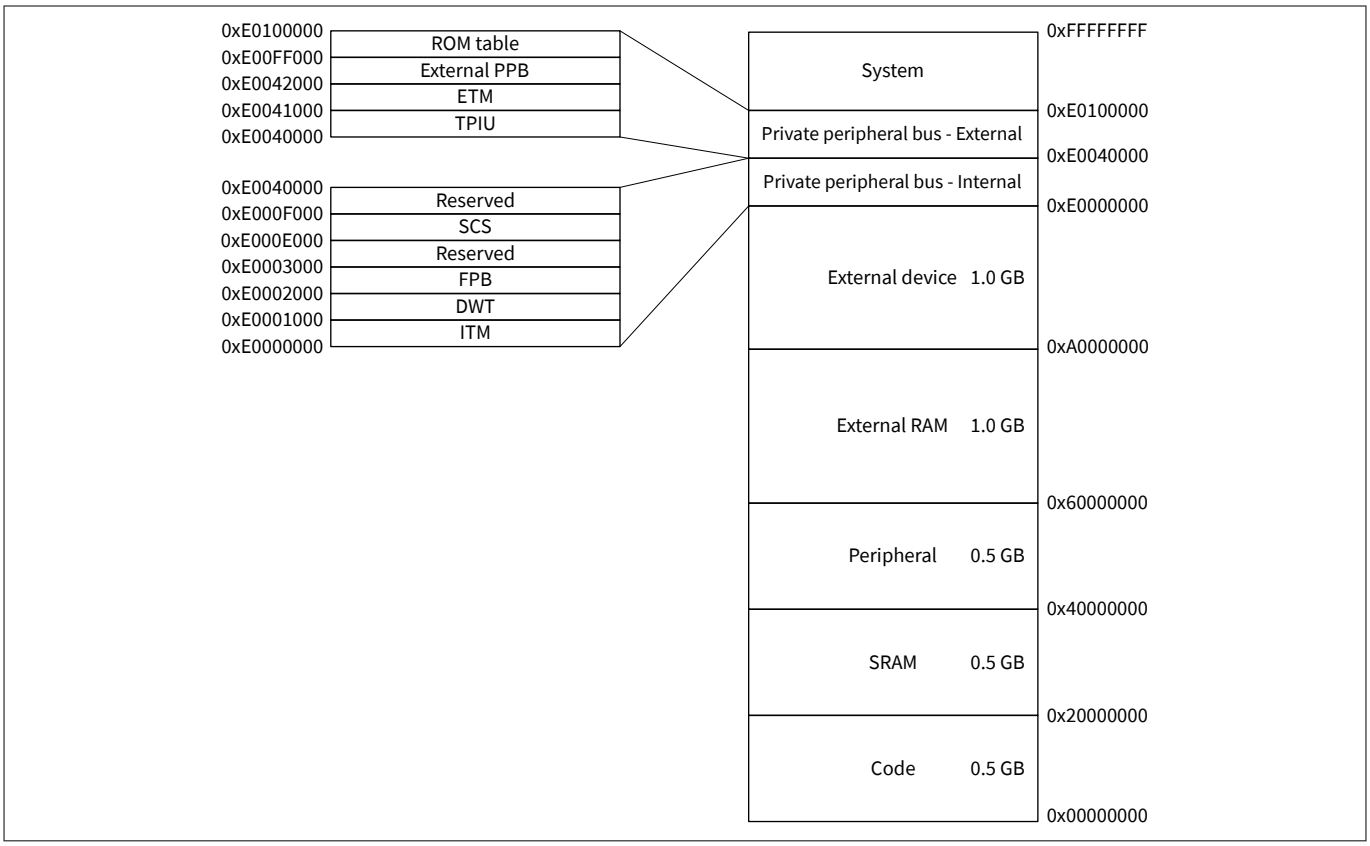


Figure 44 Original Arm® Cortex®-M0 memory map

The MOTIX™ TLE984xQX manipulates operands in the following memory spaces:

- 36, 40, 48, 52 or 64 KB of flash memory (product variant dependent) in code space (including 4 KB EEPROM emulation)
- 24 KB boot ROM memory in code space (used for boot code and IP storage)
- 2 or 4 KB (product variant dependent) 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 linear address space, up to 0.5 GB

The figure below shows the detailed address alignment of MOTIX™ TLE984xQX.

The on-chip memory modules available in the MOTIX™ TLE984xQX are:

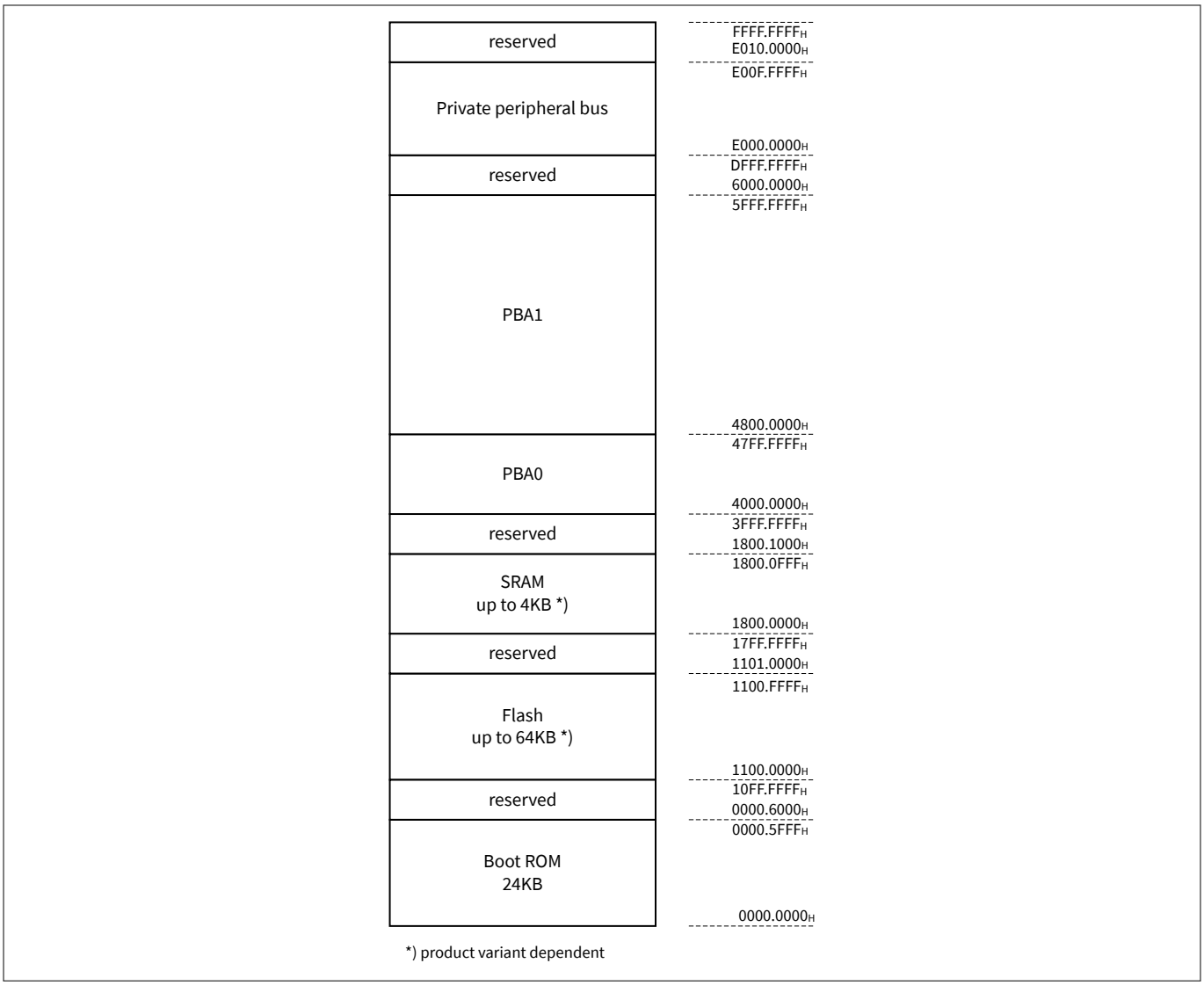


Figure 45 MOTIX™ TLE984xQX memory map

Each module provides, beside the physical memory implementation, standard AHB-Lite interface interface and error correction code (ECC) logic if needed.

Table 64 Memory map

Start (hex)	End (hex)	Space name	Usage
0000_0000	0000_5FFF	Code/data	Boot-ROM, 24 Kbytes
0000_6000	10FF_FFFF	Reserved	Reserved
1100_0000	1100_FFFF	Code/data	Flash, 36, 40, 48, 52 or 64 KB (product variant dependent)
1101_0000	17FF_FFFF	Reserved	Reserved
1800_0000	1800_0FFF	Code/data	SRAM, 2 or 4 KB (product variant dependent)
1800_1000	3FFF_FFFF	Reserved	Reserved
4000_0000	47FF_FFFF	Peripheral 0	Peripheral 0 (PBA0)
4800_0000	5FFF_FFFF	Peripheral 1	Peripheral 1 (PBA1)
6000_0000	DFFF_FFFF	Reserved	Reserved
E000_0000	E00F_FFFF	PPB, private peripheral bus	CPU
E010_0000	FFFF_FFFF	Vendor specific	Reserved

Table 65 Peripheral memory map

Bus structure	Modules	Start address	End address
Peripherals 0	Reserved	40000000_H	$40003FFF_H$
	ADC1	40004000_H	$40007FFF_H$
	Reserved	40008000_H	$4000BFFF_H$
	CCU6	$4000C000_H$	$4000FFFF_H$
	GPT12	40010000_H	$40013FFF_H$
	Reserved	40014000_H	$4001BFFF_H$
	LS	$4001C000_H$	$4001FFFF_H$
	Reserved	40020000_H	$40023FFF_H$
	HS	40024000_H	$40027FFF_H$
	Reserved	40028000_H	$47FFFFFF_H$
Peripherals 1	Reserved	48000000_H	$48003FFF_H$
	T2	48004000_H	$48004FFF_H$
	T21	48005000_H	$48005FFF_H$
	Reserved	48006000_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	$4802A000_H$	$50003FFF_H$
	PMU	50004000_H	$50004FFF_H$
	SCU	50005000_H	$50005FFF_H$
	SCUPM	50006000_H	$50006FFF_H$
	Reserved	50007000_H	$5FFFFFFF_H$

10 Memory control unit

10.1 Features

- Provides memory access to ROM, RAM, NVM, config sector through AHB-Lite interface
- MBIST for RAM
- MBIST for ROM
- NVM configuration with special function registers through AHB-Lite interface
- Hardware memory protection logic

10.2 Introduction

10.2.1 Block diagram

The memory control unit is divided in the following submodules:

- NVM memory module (embedded flash memory)
- RAM memory module
- BootROM memory module
- Memory protection unit (MPU) module
- LMB (local memory bus) interface logic

A block diagram view of the memory control unit, together with the main interface signals, is shown in the following figure:

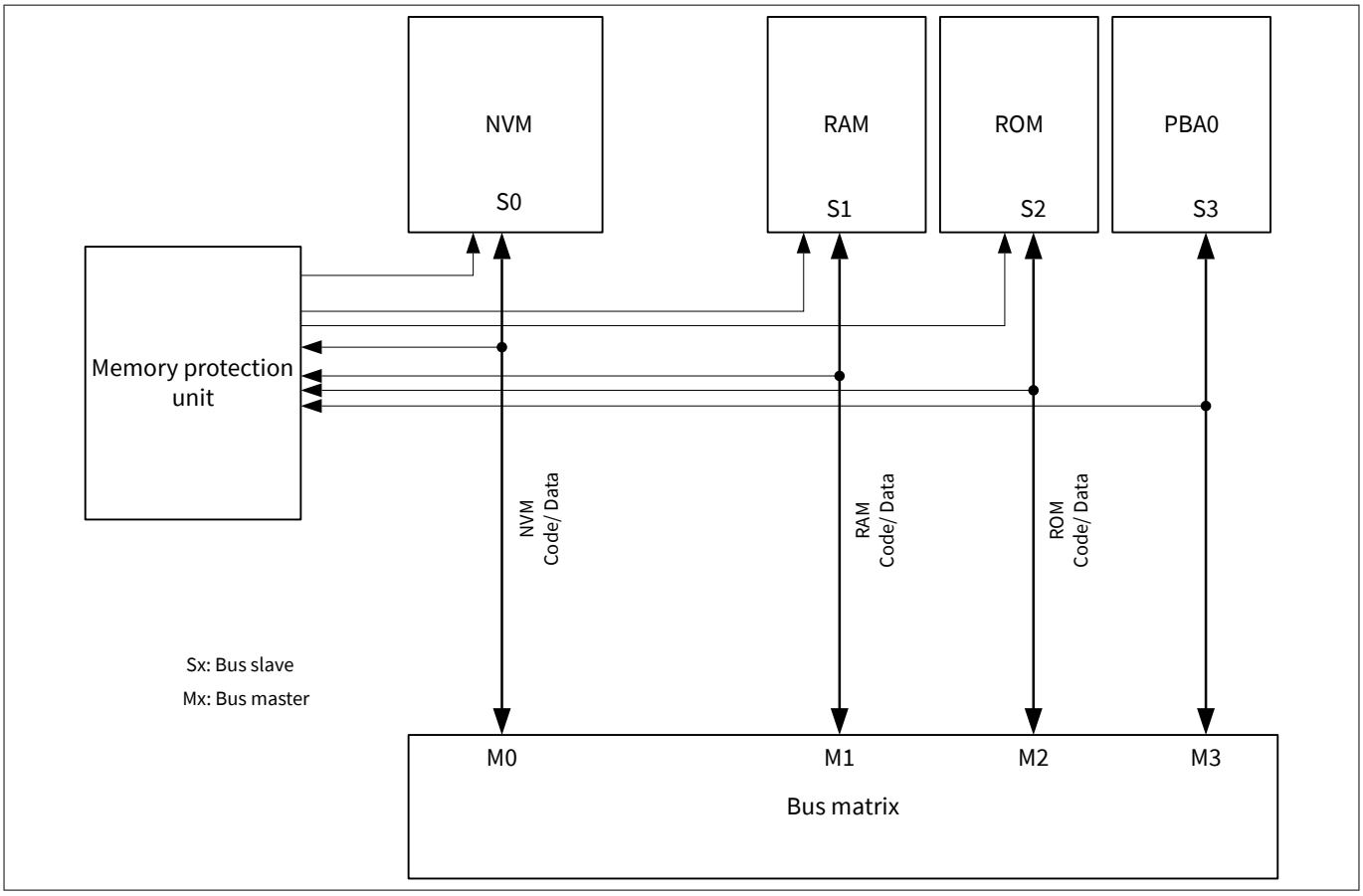


Figure 46 Memory control unit block view

Functional features for RAM

- 2 or 4 Kbytes (product variant dependent) RAM
- Error correction code (ECC) for detection of single bit and double bit errors and dynamic correction of single bit errors
- Single byte access

As shown in Figure 46, the memory control unit interface communicates with the external world, mainly the core, through 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 memory control unit has access to further core specific signals, relevant for memory protection.

10.3 NVM module (flash memory)

The flash memory provides an embedded user-programmable non-volatile memory, allowing fast and reliable storage of user code and data.

A complete description of the flash memory features can be found in Chapter 11.

10.4 BootROM module

The MOTIX™ TLE984xQX BootROM module provides physical implementation of the memory module as well as needed complementary features and interface towards the core.

The module provides proper access through a 32-bit AHB-Lite data interface multiplexed on Arm® Cortex®-M0 system bus for code/data access.

The BootROM module in MOTIX™ TLE984xQX 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

10.4.1 BootROM addressing

The BootROM, as visible from the memory map, is mapped starting at the address range 00000000_H - $00005FFF_H$. After any reset, the device hardware-controlled start address is 00000000_H. At this location, the default VTOR to be used is stored.

10.4.2 BootROM firmware program structure

The BootROM firmware provides basic functionality required to be executed after reset and routines for specific operation, such as:

- Start-up 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 through 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

10.5 RAM module

The MOTIX™ TLE984xQX RAM module provides physical implementation of the memory module as well as needed complementary features and interface towards the core.

The module provides proper access through a 32-bit AHB-Lite data interface multiplexed on Arm® Cortex®-M0 system bus for code/data access.

The RAM module in MOTIX™ TLE984xQX has a capacity of 4 Kbyte, organized with words of 32 bits.

The module support 1-bit error correction and 2-bit 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 through the interrupt enable bit in register EDCCON to trigger an exception.

10.5.1 RAM addressing

The RAM, as visible from the memory map, is mapped at the address range 18000000_H - $18000FFF_H$. The module is mapped in the code area of the Arm® Cortex®-M0 map regions and can be used as program memory for code fetching as well as data storing.

10.6 Memory protection unit (MPU)

The target of the memory protection scheme is to prevent unauthorized read out of critical data and user IPs from the BootROM and NVM as well as to prevent accidental memory data modification.

The MOTIX™ TLE984xQX protection scheme is divided in 2 parts interacting together.

The first memory protection scheme is firmware based and involves the blocking of all external access to the device. More information on the firmware based protection scheme can be found in Chapter 10.6.3.

The second memory protection scheme is hardware based; The “source” address, from which a memory read instruction is fetched, and the “target” address, where addressed data are stored, are checked by the memory protection unit (MPU) to determine if the access must be blocked. Read instructions executed from an unsafe memory address (for example RAM) that target the BootROM or NVM are blocked when the respective protection mode is enabled. The hardware protection scheme is further described in Chapter 10.6.2.

10.6.1 Memory protection regions

The MOTIX™ TLE984xQX provides the following protection regions:

- BootROM region
- Customer BSL region BootROM
- Linear NVM region
- Non-linear NVM region

The protection scheme implemented for the NVM memory module supports 3 different protection regions. On each region the protection feature can be enabled or disabled independently according to the mechanism and limitation further explained in the Chapter 10.6.2.2.

The following figure shows the NVM memory regions supported by the protection mechanism.

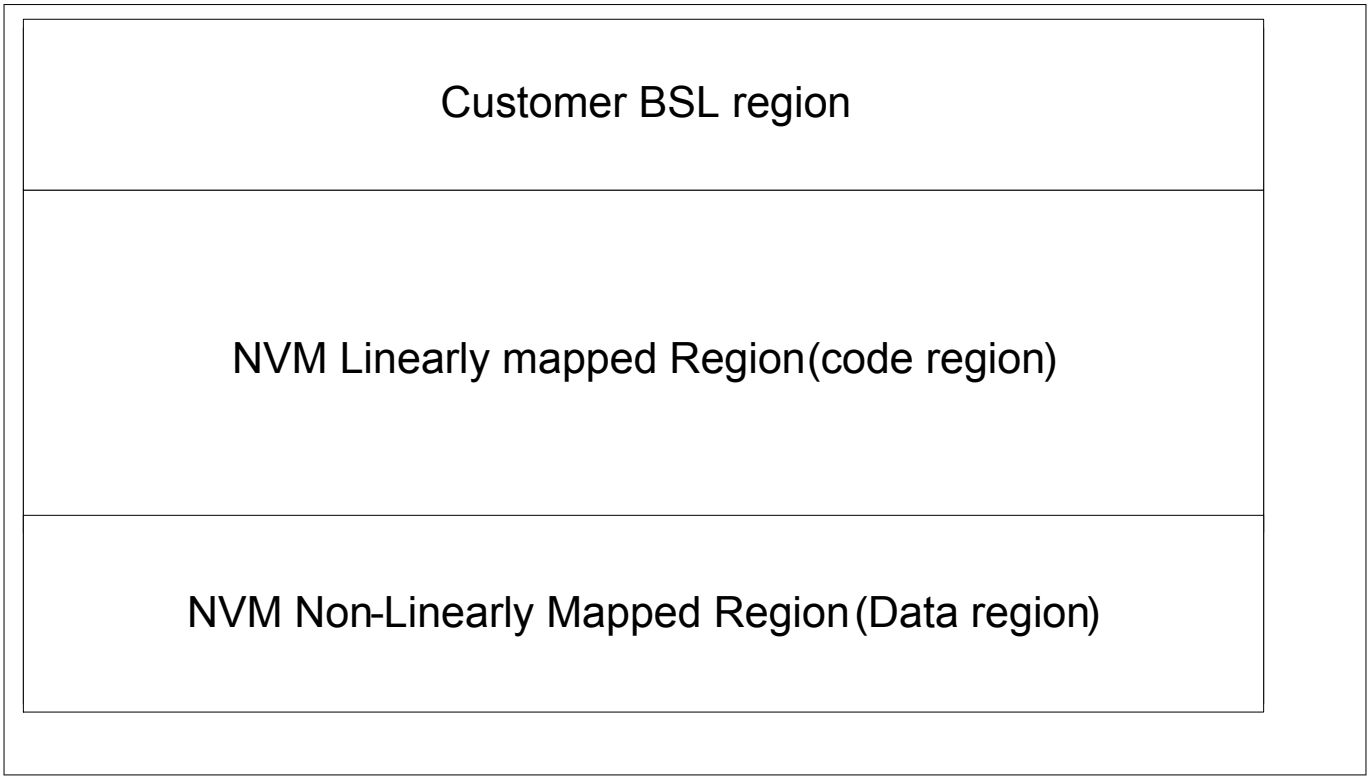


Figure 47 NVM protection regions

10.6.2 Hardware protection mode

The hardware protection mode controls the access right on each memory or memory region available. Every access to any memory is checked against the memory protection settings and accordingly executed or rejected.

For the MOTIX™ TLE984xQX BootROM protection mode is always enabled (hardware default) and it can never be disabled. The NVM protection modes can instead be enabled separately for customer BSL, linear and non-linear mapped ranges. While the BootROM protection mode is enabled, the NVM protection mode may be enabled as well to further prevent code read out.

Regardless the protection mode enabling, the following accesses are always be possible:

- Data reading instructions executed from the BootROM targeting BootROM itself or the RAM
- Data reading instructions executed from the customer BSL NVM region targeting customer BSL NVM region itself, non-linearly mapped NVM region or RAM
- Data reading instructions executed from the linearly mapped NVM region targeting linearly mapped NVM region itself, non-linearly mapped NVM region or RAM
- Data reading instructions executed from the non-linearly mapped NVM region targeting RAM
- Data reading instructions executed from the targeting RAM itself
- Instruction fetch into any region
- Data read access to the Interrupt vector table (depending on the VTOR settings)

Unauthorized data reading instructions will be detected and consequently blocked.

10.6.2.1 BootROM protection mode

The BootROM read protection modes is enabled by default and consequently the following accesses are restricted:

- Data reading instructions executed from the NVM or targeting BootROM

The following figure shows all the data reading instructions authorized when only the BootROM protection is enabled (NVM protection disabled).

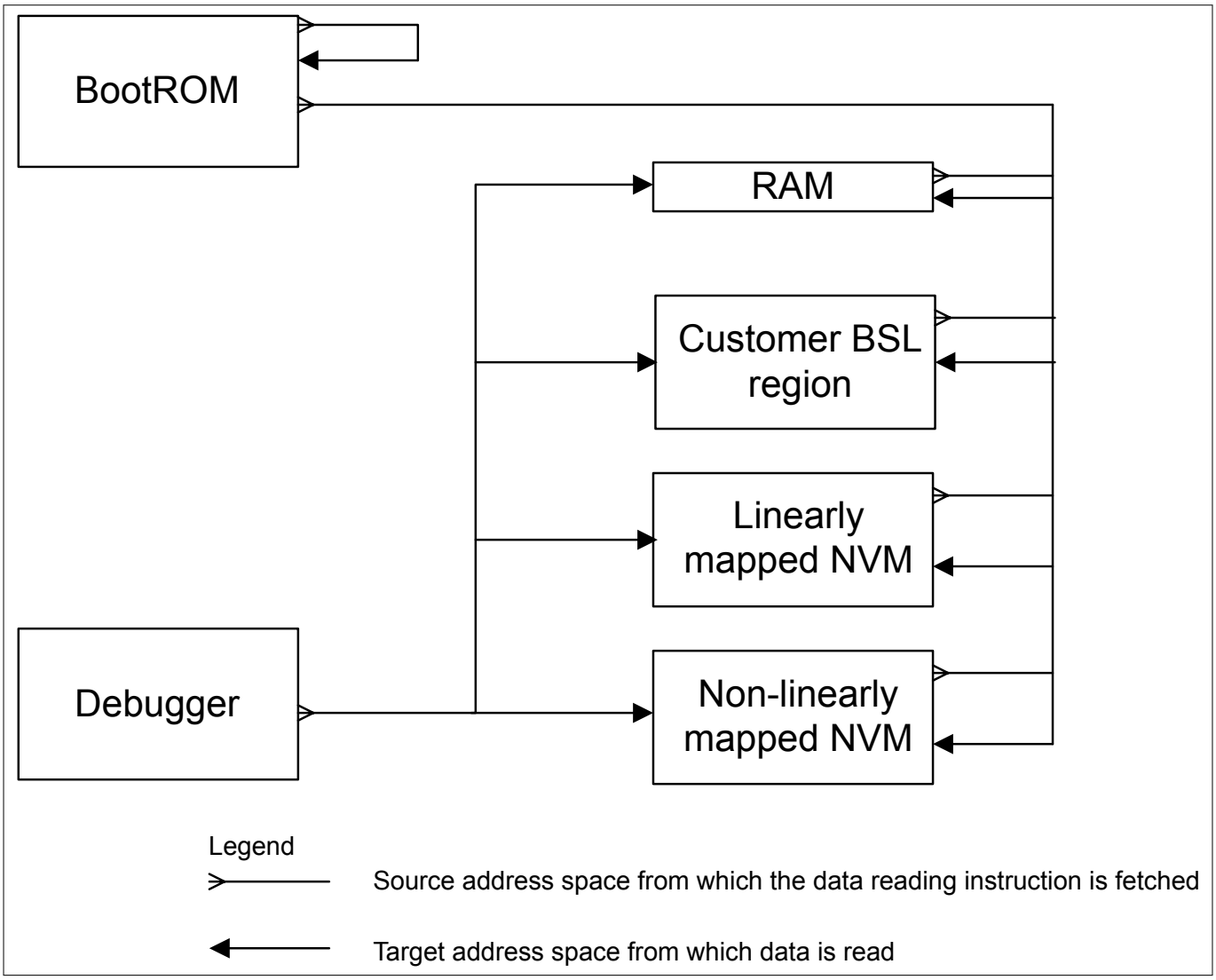


Figure 48 BootROM protection mode enabled

If the BootROM read protection mode is enabled without enabling of any NVM protection mode:

- Data reading instructions executed from NVM or RAM can target itself or one another
- Data reading instructions executed from the BootROM can target itself, NVM or RAM
- Data reading access issued by the debugger can target NVM or RAM

In addition, to avoid an indirect leak of information by hacking through the debugger, breakpoints set and step through features are disabled on the BootROM. In case debugger issues such a command, the command is suspended till the moment in which the code execution leaves the read protected region (BootROM). More information about protection against debugger activity can be found in Chapter 10.7

10.6.2.2 NVM protection modes

The NVM address space is divided into the three supported NVM regions: customer BSL, linearly mapped, non-linearly mapped region.

The customer BSL region is supposed to be used for special user code that might not be changed over device life time. Since this region is anyhow meant to host user executable code, the region is linearly mapped even if, to distinguish it from standard user code region, it is named “customer BSL”.

The linearly mapped region is supposed to be used for user standard application code while the non-linearly mapped region is meant to be used for data storage even if code execution is not prevented.

The protection on each of the region is individually controlled by the setting of the NVM_PROT_STS register bits. Further details regarding the NVM region protection enable/disable are described in the Chapter 10.6.2.2.4

Application hint regarding read-protection:

The customer-BSL region can also be used as "normal" user code area. In that case, special care must be taken regarding protection:

If all regions are read-protected, data reads between CBSL-region (0..4K) and user code region (above 4K) would be blocked. This is possible to avoid with certain compiler settings (defining different regions), but somehow painful.

10.6.2.2.1 Customer BSL region protection mode

The customer BSL region protection can be controlled via proper dedicated password as described in the Chapter 10.6.2.2.4.

When its write protection is enabled, any operation capable to change the NVM values stored in this region is blocked. For example, neither a program nor an erase can be executed.

In case the memory protection unit (MPU) and NVM control logic detect that the target address belongs to this region and that write protection is set, a proper alarm signal is forwarded to the NVM module to prevent the NVM state machine from accepting any program or erase command (including fast invalidation). This prevents inadvertent destruction of stored data when protection is set.

When Customer BSL region read protection is enabled, the following accesses are restricted:

- Data reading instructions executed from any other memory region (BootROM, RAM, linear NVM and non-linear NVM) targeting the customer BSL region
- Data reading accesses triggered by debugger targeting the customer BSL region

The following figure shows all the data reading instructions authorized when both the BootROM and customer BSL region read protections are enabled.

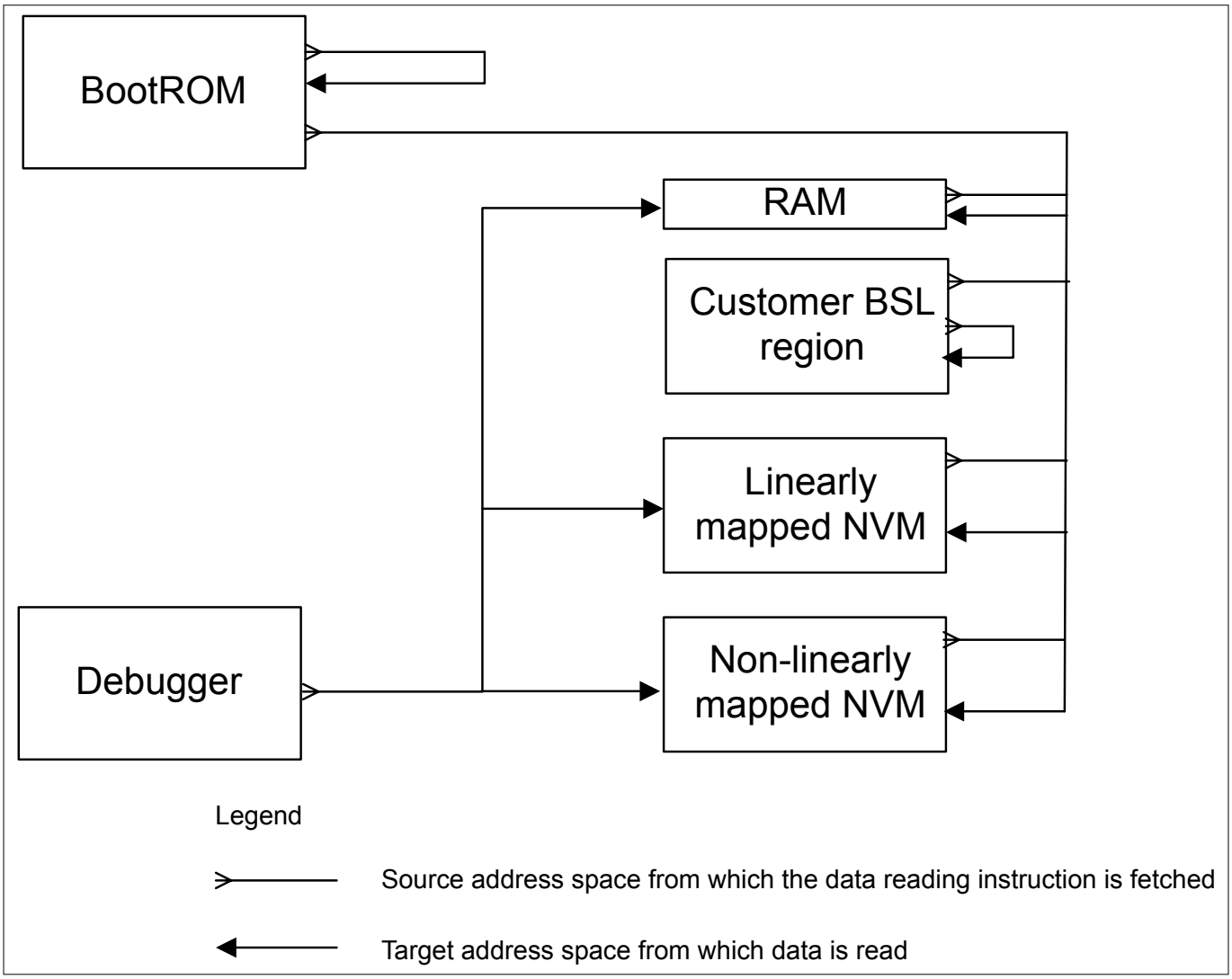


Figure 49 Customer BSL region protection mode enabled

If the BootROM and the customer BSL protection modes are enabled:

- Data reading instructions executed from the linear NVM, non-linear NVM or RAM can target itself or one another
- Data reading instructions executed from the BootROM can target itself
- Data reading instructions executed from the customer BSL NVM region can target itself, linear NVM, non-linear NVM or RAM

10.6.2.2.2 NVM linear protection mode

The NVM linear protection can be controlled through proper dedicated password or through the NVMPROT_STS register as described in the Chapter 10.6.2.2.4.

When its write protection is enabled, any operation capable to change the NVM values stored in this region is blocked. For example, neither a program nor an erase can be executed.

Regarding write protection the 100TP pages are considered to be part of the linear NVM. For this reason, in case the write protection in this region is set, even the 100TP program is blocked.

In case the MPU and NVM control logic detect that the target address belongs to this region and that write protection is set, a proper alarm signal is forwarded to the NVM module to prevent the NVM state machine from

accepting any program or erase command (including fast invalidation). This prevents inadvertent destruction of stored data while protection is set.

When NVM linear read protection is enabled, the following accesses are restricted:

- Data reading instructions executed from any other memory region (BootROM, RAM, customer BSL and non-linear NVM) targeting the NVM linear region
- Data reading accesses triggered by debugger targeting the NVM linear region

Figure 50 shows all the data reading instructions authorized when the BootROM, the customer BSL region and NVM linear read protections are enabled.

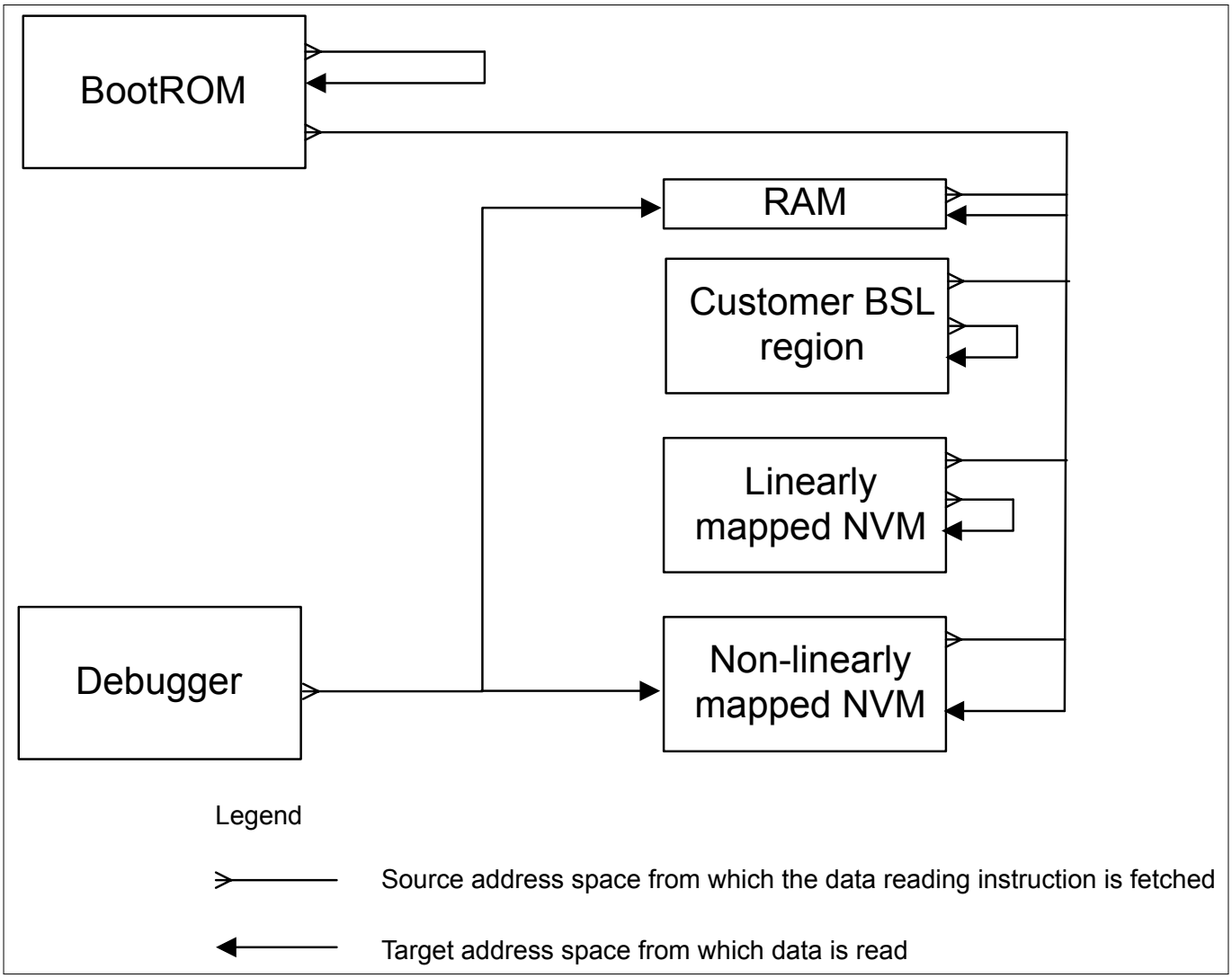


Figure 50 NVM linear protection mode enabled

If the BootROM, the customer BSL and the NVM linear protection modes are enabled:

- Data reading instructions executed from the non-linear NVM or RAM can target itself or one another
- Data reading instructions executed from the BootROM can target itself
- Data reading instructions executed from the customer BSL NVM region can target itself, non-linear NVM or RAM
- Data reading instructions executed from the NVM linear region can target itself, non-linear NVM or RAM

10.6.2.2.3 NVM non-linear protection mode

The NVM Non-Linear protection can be controlled through proper dedicated Password or through the NVMPROT_STS register as described in the Chapter 10.6.2.2.4.

When its write protection is enabled, any operation capable to change the NVM values stored in this region is blocked. For example, neither a program nor an erase can be executed.

In case the MPU and NVM control logic detect that the target address belongs to this region and that write protection is set, a proper alarm signal is forwarded to the NVM module to prevent the NVM state machine from accepting any program or erase command (including fast invalidation). This prevents inadvertent destruction of stored data while protection is set.

When NVM non-linear read protection is enabled, the following accesses are restricted:

- Data reading instructions executed from BootROM, RAM and non-linear NVM targeting the NVM non-linear region
- Data reading accesses triggered by debugger targeting the NVM non-linear region

Figure 51 shows all the data reading instructions authorized when the BootROM, the customer BSL region, NVM linear and NVM non-linear read protections are enabled.

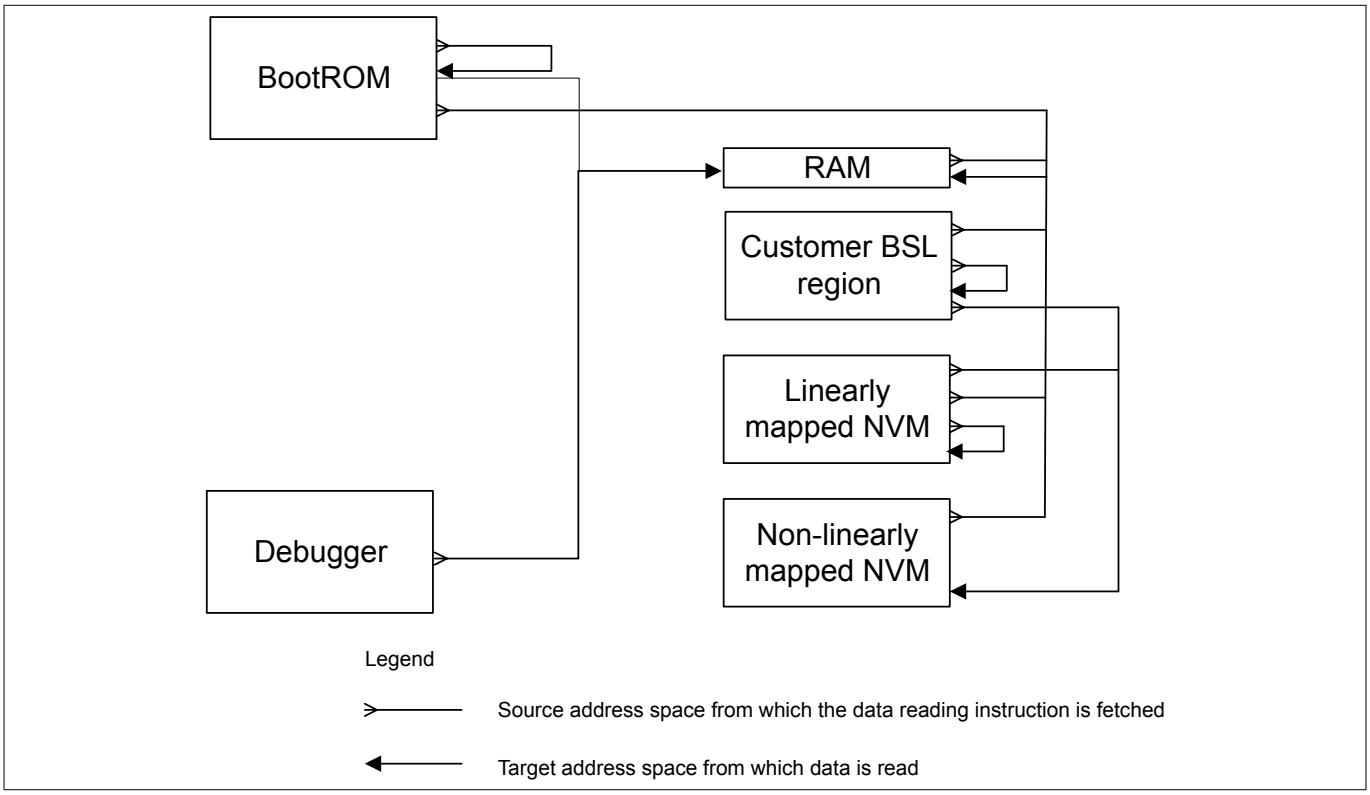


Figure 51 NVM non-linear protection mode enabled

If the BootROM, the customer BSL, the NVM linear and the NVM non-linear protection modes are enabled:

- Data reading instructions executed from the non-linear NVM can target the RAM
- Data reading instructions executed from the RAM can target itself
- Data reading instructions executed from the BootROM can target itself
- Data reading instructions executed from the customer BSL NVM region can target itself, non-linear NVM or RAM
- Data reading instructions executed from the NVM Linear region can target itself, non-linear NVM or RAM

10.6.2.2.4 NVM protection mode control

The read and write protection on the different regions are controlled via the register NVM_PROT_STS.

The value of this register can be changed in 2 different ways.

Memory region protection password

The first method is based on a region specific protection password. After the complete code has been programmed into the customer BSL and linear NVM regions, the protection scheme can be enabled by calling the BootROM password routine by means of the dedicated MOTIX™ TLE984xQX BSL mode. The BootROM password routine programs a user provided password into the reserved space register according to the information stored into the 2 most significant bits of the password. The format of the password is shown in the following figure.

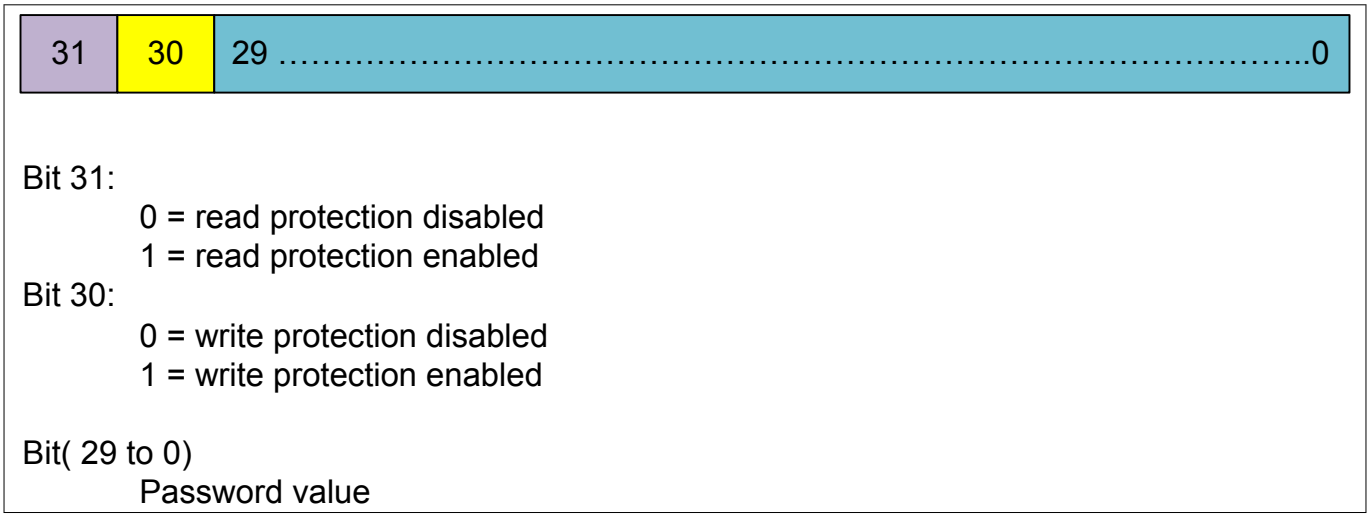


Figure 52 NVM region password format

To allow external access to the device or to reprogram a new password

When removing the password through the BootROM password routine, in case the provided password does not match the valid password currently stored.

There is a password for each region.

Memory region protection register access

The hardware memory protection mechanism is controlled by the values of the NVM_ PROT_STS register bits. When user set a protection through password, the BootROM start-up sequence enables proper protection modes by writing the related bit of the NVM_PROT_STS register.

Even if user enables protection on a defined region at start-up using the dedicated password, during the application code execution there might be the need to temporarily remove the protection to store some new code/data.

For example, user might want to set by default at start-up the write protection on the non-linear NVM region to avoid accidental data loss. Nevertheless, during application code execution, there might be the need to update some of the data stored in this region. For this reason, the MOTIX™ TLE984xQX provides the user the possibility to change the protection status writing directly the NVM_PROT_STS bits. The changes in the active protection scheme obtained through direct access to the register are anyhow temporary and the default protections controlled by password status will be automatically restored at the next reset (next BootROM startup sequence execution).

Anyhow, to safeguard against accidental access by user on this register, its access is controlled depending on boot mode, memory regions protections status and source address.

The BootROM code (firmware), provides proper APIs to individually set/clear, read and write protection on each memory protection region.

These routines:

- Freely change read/write protection as long as no valid password for the target region has been installed.
- In case a valid password for the target region is installed, the routine has to take the current valid password as input. If the provided password matches the current valid installed one, the target NVM_PROT_STS bits can be freely changed. In case, instead, the password provided as input does not match the current installed one, the NVM_PROT_STS target bits are not changed and the content of all the 3 password controlled memory region (customer BSL, linear NVM and non-linear NVM) together with the related passwords is erased to avoid hacking of the stored password by repetitive trials.

The above reported feature and routines applies in general for all the different memory protection regions. Exceptions:

- Customer BSL protection region is controllable only via password.

Note: The possibility to remove a protection even when password has been installed is provided to ensure to the user the possibility to unlock the device in case of FAR investigation. Of course, as reported above, all the provided mechanism require the knowledge of the current installed password. Without knowing the password value unlock is not possible.

10.6.3 Firmware protection mode

The firmware protection scheme is the second leg of the overall memory protection concept.

In particular, the BootROM code provides following features:

- Each BootROM routine provided by the firmware for the NVM data handling (e.g. program or erase routines) checks the address to identify which region is targeted and accordingly check the relevant bit of the NVM_PROT_STS register. In case the write protection for the target region is not set, the operation is executed. In case, instead, the write protection for the target region is set, the routine is exited reporting a proper error
- In case read protection is enabled on any of the password controllable protection regions (customer BSL, linear NVM and non-linear NVM), all provided feature to download code into the device are blocked (for example all BSL modes available to download code into the device)

The firmware protection features are provided to complete the protection scheme. The first implemented feature is implemented to ease the detection of any BootROM routine fails due to the protection setting. In fact, in case a BootROM routine is called with write protection enabled, the routine would not affect the NVM content due to the hardware protection scheme. In such a case, the BootROM based protection feature would recognize in firmware the protection settings and stop the routine providing a proper fail indication to the user code.

The second firmware based protection feature is instead needed to make the read protection mechanism provided by hardware effective. In fact, the feature for code download could be used for hacking even if the read protection is set on a region (but not the write protection). It would then be possible to read out the code/data by downloading a proper code into the same region. In fact, according to the hardware protection scheme, a code running from a selected region can always address itself. So, the firmware will block all the boot options such that it is not possible to load and execute any external code, but only to execute user code starting at address pointed by the standard reset handler routine address stored at 11000004_H.

10.7 Core protection mode

The Chapter 10.6.2 and Chapter 10.6.3 describe the protection against accidental or malicious read and write memory access implemented in hardware and software. The hardware implements a check of all direct access to the each memory region (even from debugger) granting access only when the target region is not protected. The firmware, instead, blocks any download of new code via BSL in case any NVM read protection is installed to avoid the possibility to install any malicious software that removes the protection and reads out the user code.

Without any further feature, there would still be the possibility to use the debugger to leak information about user code. In fact, even if the read out of the memory content via debugger is blocked when accessing a read protected region, it is still possible to use the other debugger features (e.g. step through, breakpoints, watchpoints, code profiling) to perform a reverse engineering of executed code.

For this reason, a further level of protection is implemented between the memory control unit and the core.

In particular, the debugger features are disabled according to the current program counter and the installed passwords.

By default, when no password is installed, the debug features are disabled while executing from the BootROM thus avoiding any code profiling.

As soon as at least one read protection is set via one of the three NVM region passwords then the protected region is actually extended to the complete MOTIX™ TLE984xQX code region. This means that any debugger command will be left pending thus resulting in a time out and a loss of connection. Consequently, once a password with most significant bit set to 1 is installed (read protection enabled), at the following reset no connection to the device is possible anymore.

11 NVM module (flash memory)

The flash memory provides an 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 on data block (double words, 64 bit).
- Interrupt and signaling of double bit error by NMI, address of double bit error readable by FW API user routine.
- Possibility of checking single bit error occurrence by ROM routines
- Program width of 128 Byte (page)
- Minimum erase width of 128 Byte (page)
- Integrated hardware support for EEPROM emulation
- 8 Byte read access
- Physical read access time: typical 75 ns
- Code read access acceleration integrated; read buffer
- Page program time: typical 3 ms
- Programming time for 64 Kbyte via debug interface: < 1800 ms (typical)
- Page erase (128 Byte) and sector erase (4 Kbyte) time: typical 4 ms
- 3 separate keys for data area, program area and BSL area
- Password protection for three configurable program flash areas, three separate keys for data, program and BSL
- Option to protect read out via debug interface in application run mode. NVM protection mode available, which can be enabled/disabled with password
- Write/erase access to 100TP (e.g. option Bytes) is possible via the debug interface

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 are provided to ease NVM, and other operations including EEPROM emulation.

The MOTIX™ TLE984xQX NVM module provides physical implementation of the memory module as well as needed complementary features and interface towards the core.

The module provides proper access to the memory through 2 AHB-Lite interfaces: a 8-bit data interface for NVM internal register access and a 32-bit data interface for code/data access both multiplexed on Arm® Cortex®-M0 system bus.

The MOTIX™ TLE984xQX NVM module consists of the memory cell array and all the control circuits and registers needed to access the array itself. The 64 Kbyte data module is mapped in the Arm® Cortex®-M0 code address range 11000000_H - $1100FFFF_H$ while the dedicated SFRs are mapped in the Arm® Cortex®-M0 system address range 58004000_H - $58007FFF_H$.

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.

11.1 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.

11.1.1 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 a ready programmed page (implicitly causing an erase of the page before writing the new data).

Data portions

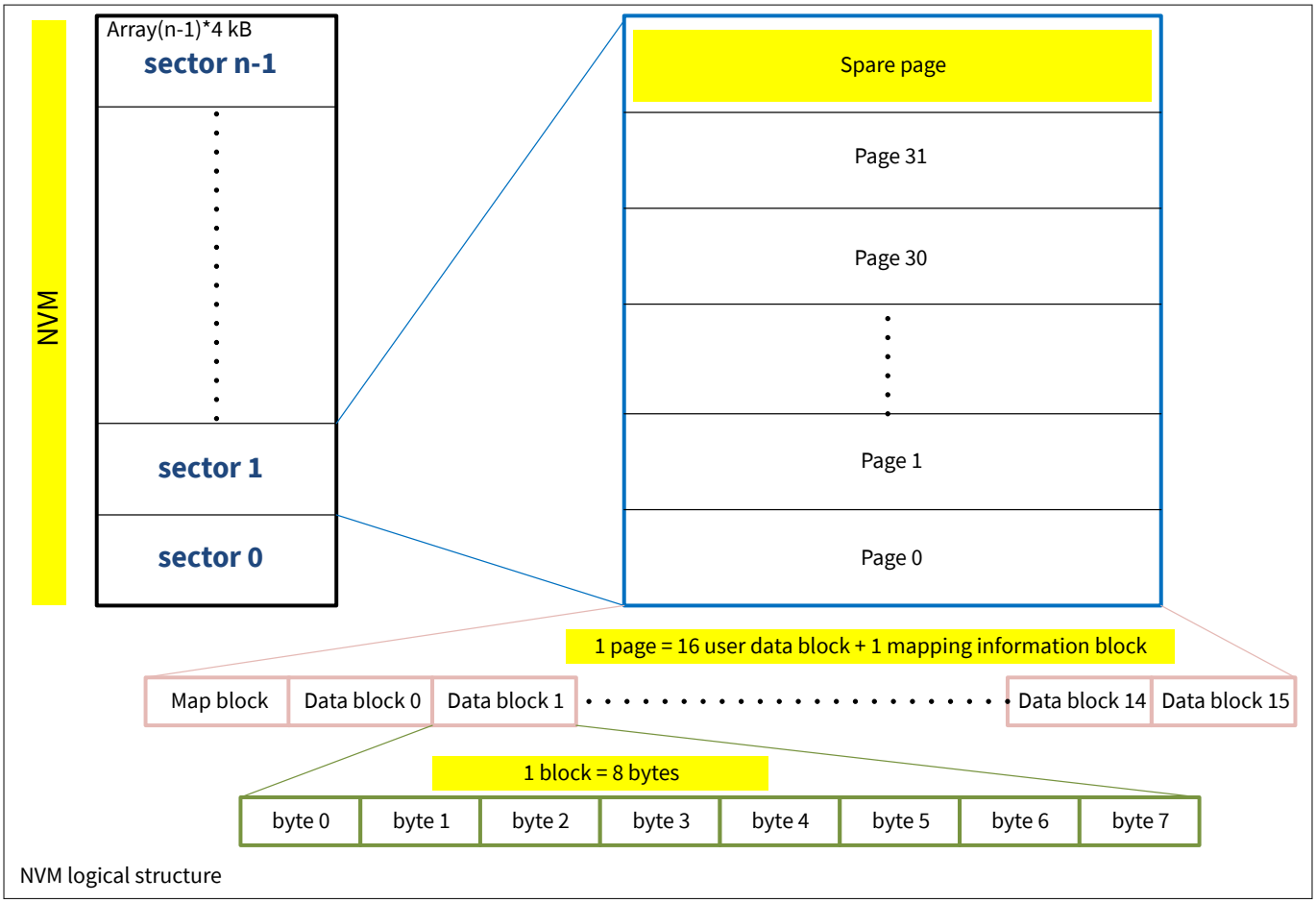


Figure 53 Logical structure of the NVM core

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 pages.

11.2 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. The main features are listed following:

- 64 KB memory size
- 3 ms write time per page
- 4 ms erase time per page
- Error correction and error detection code (ECC and EDC)
- In module memory protection logic

11.2.1 Basic block functions

Description of all major/significant blocks with sub-block diagrams.

Diagram showing the product´s internal functional composition.

The following figure shows a schematic block diagram of the NVM module:

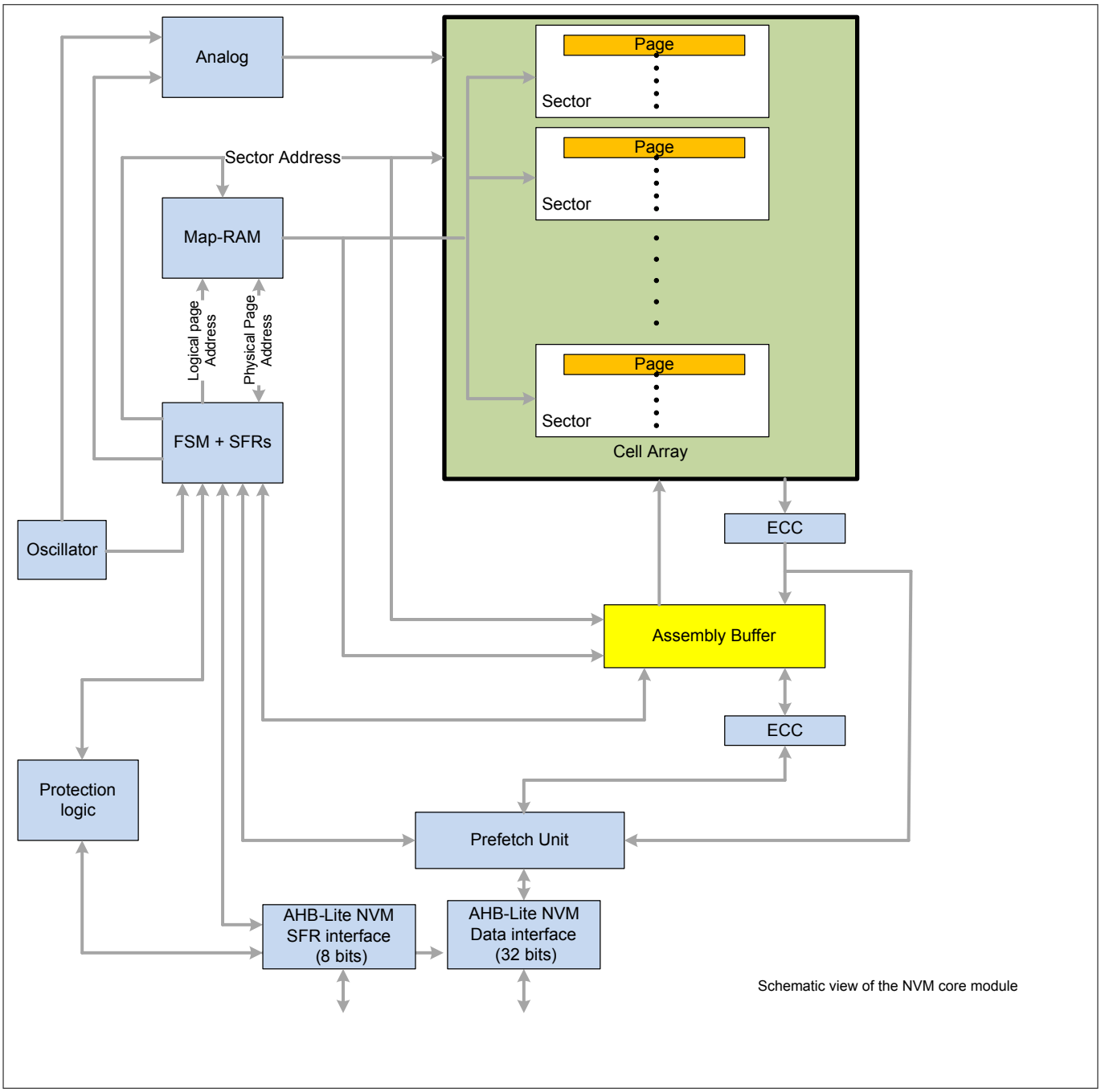


Figure 54 Schematic view of the NVM core module

11.2.2 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 mapblocK.

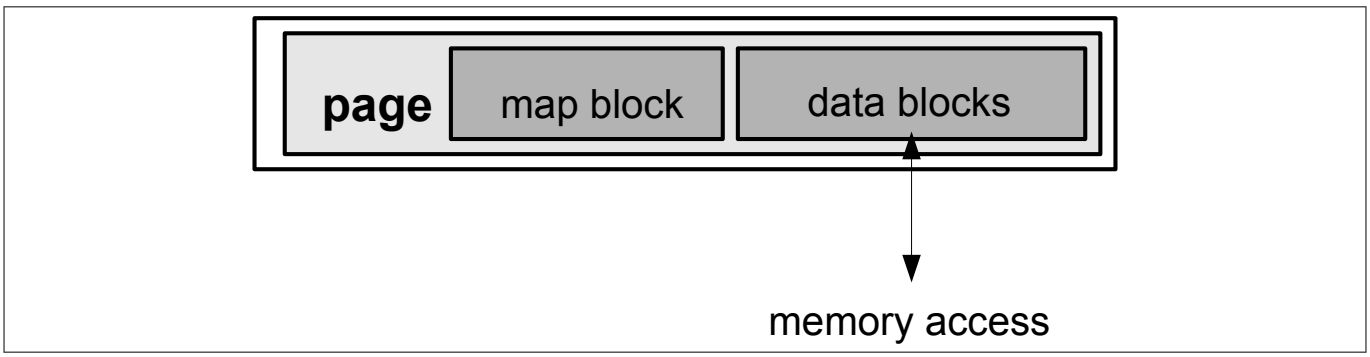


Figure 55 Structure of assembly buffer

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 mapped sector. It is completely handled by the NVM programming related BootROM routines.

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.

11.2.3 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.

11.2.4 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.

If the page accessed during a read is not mapped, an NVM_TRAP is triggered (e.g. when accessing an erased data sector).

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 wait states (3, 1 or, 0) is generated for different system frequencies selected by SYSCON0.NVMCLKFAC.

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. For execution of linear code three out of four 16-bit instructions or one out od two 32-bit instructions accesses are served without any wait states.

11.2.5 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.

11.2.6 Timing

The target timing of the hardware sequences excluding the software overhead is shown below:

- Erase: 4.0 ms per page
- Write: 3.0 ms per page
- Program (= erase + write): 7.0 ms per page

The disturb handling routine when enabled with a probability of a approximately 0.1% adds additional 7.0 ms to a page write or program operation.

11.2.7 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.

11.2.8 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.

11.2.9 Dynamic address scrambling

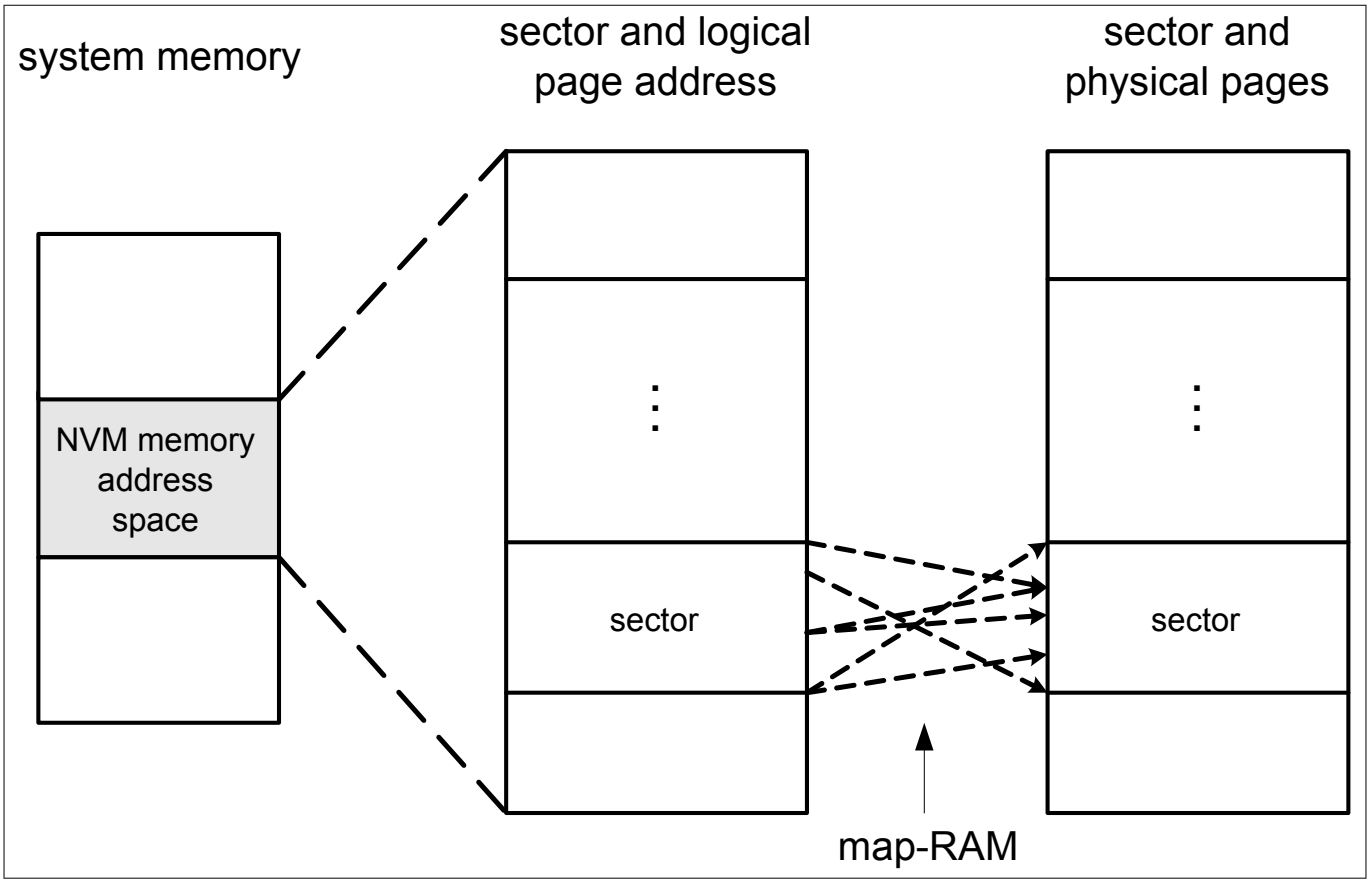


Figure 56 Dynamic address scrambling through map-RAM

Starting from the system address space of the NVM, the NVM module supports mapping of pages within each sector. As described above, this is useful for tearing-safe programming, but it also provides a dynamic address scrambling: After programming a page the new information is physically stored in a different page in the same sector. The logical page address and therefore the physical memory address stays the same. For this reason the mapping is fully transparent for the customer software.

11.2.10 Linearly mapped sectors

A number of sectors can be configured not to use the map-RAM mapping mechanism, that is for these sectors logical and physical page addresses are identical. The range of these linearly mapped sectors always starts at the lowest NVM address of the NVM module, extending upwards to higher addresses. For these sectors (intended to mainly store executable code without the need for tearing-save programming) no reconstruction of the map-RAM content after reset is necessary, which saves time during the sleep-wake-up and power-up of the chip.

11.2.11 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.

11.2.12 Hot spot distribution

In the used EEPROM technology always a whole page has to be programmed when any part of it (e.g. only a byte or doubleword) is modified. This means that cycling multiple parts of one page separately physically means cycling of the whole page every time one part is programmed.

In the following one such part (e.g. byte or doubleword) will be called a “hot spot”.

For H hot spots in a page where each hot spot is separately cycled c times, this results in a cycle stress of the page of c*H.

As the EEPROM programming is always performed by copying the modified currently mapped page to the spare page, the cycle stress is shared among two pages. Furthermore, as after some time the disturb handling described in Chapter 11.2.11 kicks in, the cycling stress eventually is shared among all 33 pages of the sector.

Therefore, the average cycle stress for a physical page in a sector is c*H/33, when H hot spots in a logical page are separately cycled c times.

On the other hand, this means that with a cycle endurance of E for a page, the number of cycles which can be performed before a page can become damaged by cycle stress can be calculated as c = 33*E/H per hot spot in the sector.

Depending on the number of hot spots in a sector the maximum allowed number of cycles c per hot spot can become unacceptably low.

If the hot spots are concentrated in one sector and other sectors have only a low number of hot spots, a hot spot distribution over several sectors is advisable. This hot spot distribution is not supported in HW but has to be done during the implementation of the software.

11.2.13 Properties of error correcting code (ECC)

The error correcting code (ECC) for the data blocks implements a one-bit error correction and a double-bit error detection for every data block of 64 bits. The correct ECC bits for every block are generated automatically when the assembly buffer is written. During every read the ECC bits are read together with the data bits. The validity of the code word is checked by hardware. Every single bit error is corrected automatically.

The described ECC mechanism results in the limitation that a block of 64 bits is the smallest data unit that can be read internally, since always a complete block has to be read to check for possible ECC errors and writing a byte automatically triggers a read of the complete block.

A data block with all bits fully erased is ECC-correct.

When a page is copied to the assembly buffer, the ECC correction of data blocks with a one-bit error is done automatically, whereas data with an uncorrectable error is passed on unchanged. No ECC interrupt is generated for ECC errors that are detected during the copying of a page to the assembly buffer.

11.2.14 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 sleep mode. 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 SCU_MEMTEST register in order to perform appropriate corrective actions if needed. Furthermore the SCU_SYS_STRTUP_STS register 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 SCU_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.

11.2.15 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.

12 Interrupt system

12.1 Features

- Up to 24 interrupt nodes for on-chip peripherals
- Up to 8 NMI nodes for critical system events
- Maximum flexibility for all 24 interrupt nodes

12.2 Introduction

12.2.1 Overview

The MOTIX™ TLE984xQX supports 24 interrupt vectors with 4 priority levels. 22 of these interrupt vectors are assigned to the on-chip peripherals: GPT12, SSC1, SSC2, CCU6 low-side switch, high-side switch and A/D converter are each assigned to one dedicated interrupt vector; while UART1 and Timer2 or 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.
- OT prewarning
- 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

Chapter 12.3.1 gives a general overview of the interrupt sources and nodes, and their corresponding control and status flags. Chapter 12.3.1.14 gives the corresponding overview for the NMI sources. The table below shows the available interrupt vectors.

Table 66 Interrupt vector table

Service request	Node ID	Description
GPT1	0	GPT1 interrupt
GPT2	1	GPT2 interrupt
MU	2	MU interrupt / ADC2, VBG interrupt
ADC1	3	ADC10 bit interrupt
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 66 (continued) Interrupt vector table

Service request	Node ID	Description
UART1	10	UART1 (ASC-LIN) interrupt (receive, transmit), t2, linsync1, LIN
UART2	11	UART2 interrupt (receive, transmit), t21, external interrupt (EINT2)
EXINT0	12	External interrupt (EINT0)
EXINT1	13	External interrupt (EINT1)
WAKEUP	14	Wake-up interrupt (generated by a wake-up event)
rfu	15	Reserved for future use
rfu	16	Reserved for future use
LS1	17	Low-side 1 interrupt
LS2	18	Low-side 2 interrupt
HS1	19	High-side 1 interrupt
HS2	20	High-side 2 interrupt (product variant dependent)
DU	21	Differential unit - DPP1 (product variant dependent, only TLE9845QX)
MONx	22	MONx interrupt
Port 2.x	23	Port 2.x - DPP1

Table 67 NMI interrupt table

Service request	Node	Description
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 and 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

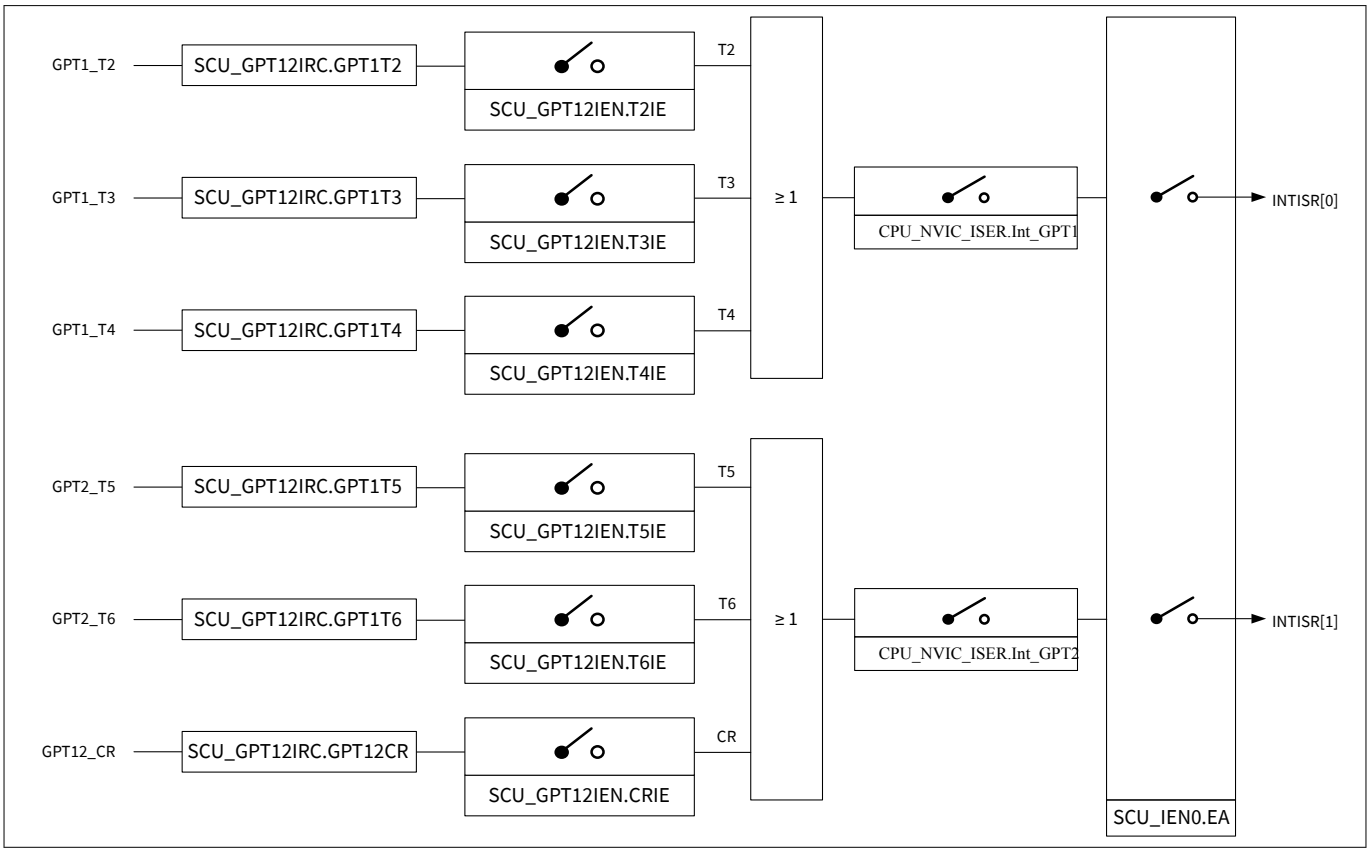


Figure 57 Interrupt request sources 0 and 1 (GPT12)

12.3.1.2 Interrupt node 2 – measurement unit

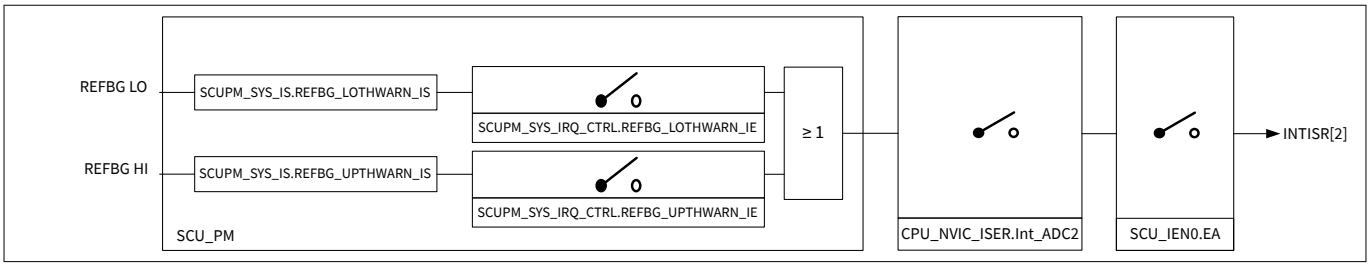


Figure 58 Interrupt request sources 2 (MU)

12.3.1.3 Interrupt node 3 – ADC10

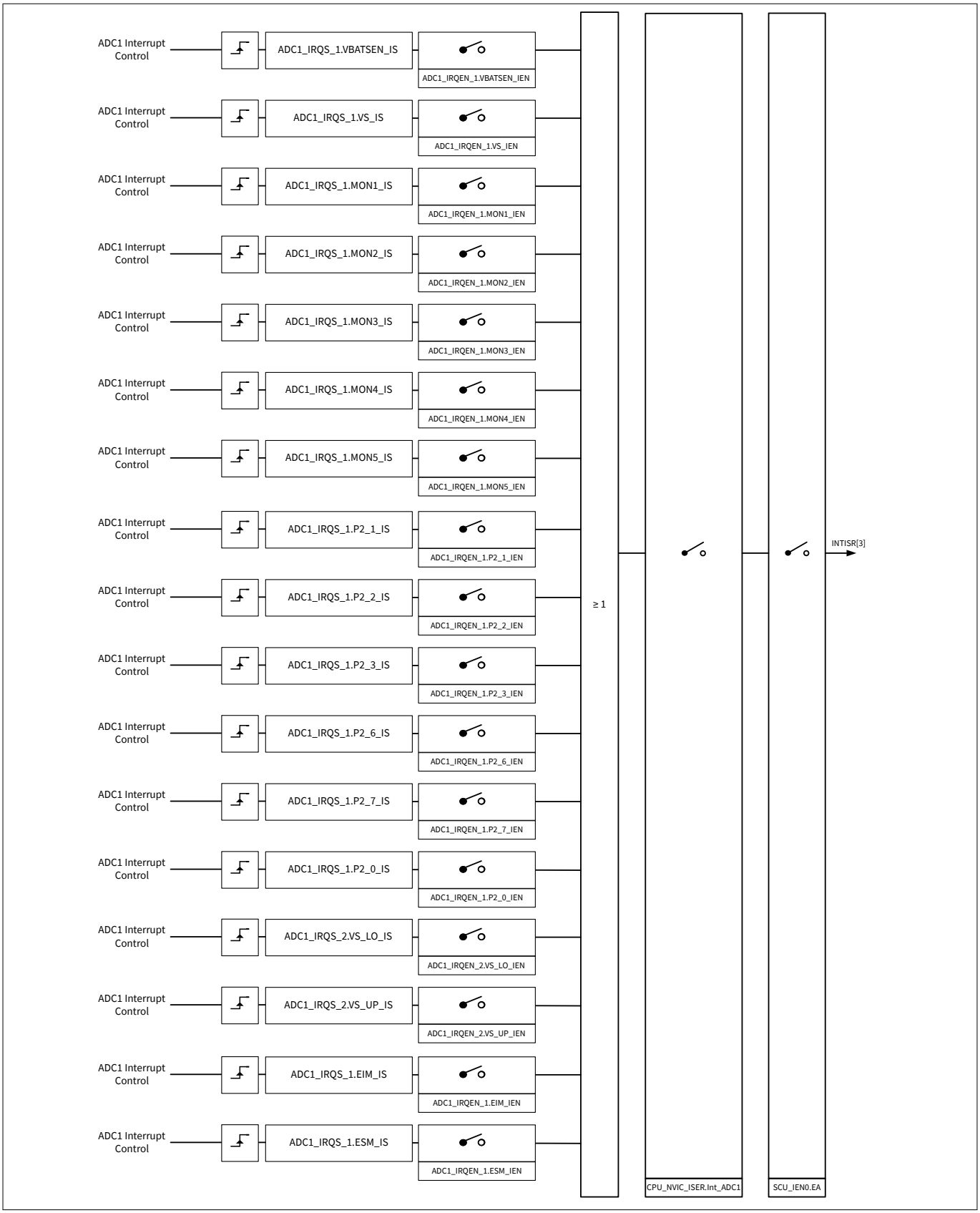


Figure 59 Interrupt request sources 3 (ADC10)

12.3.1.4 Interrupt node 4, 5, 6, 7 – CCU6

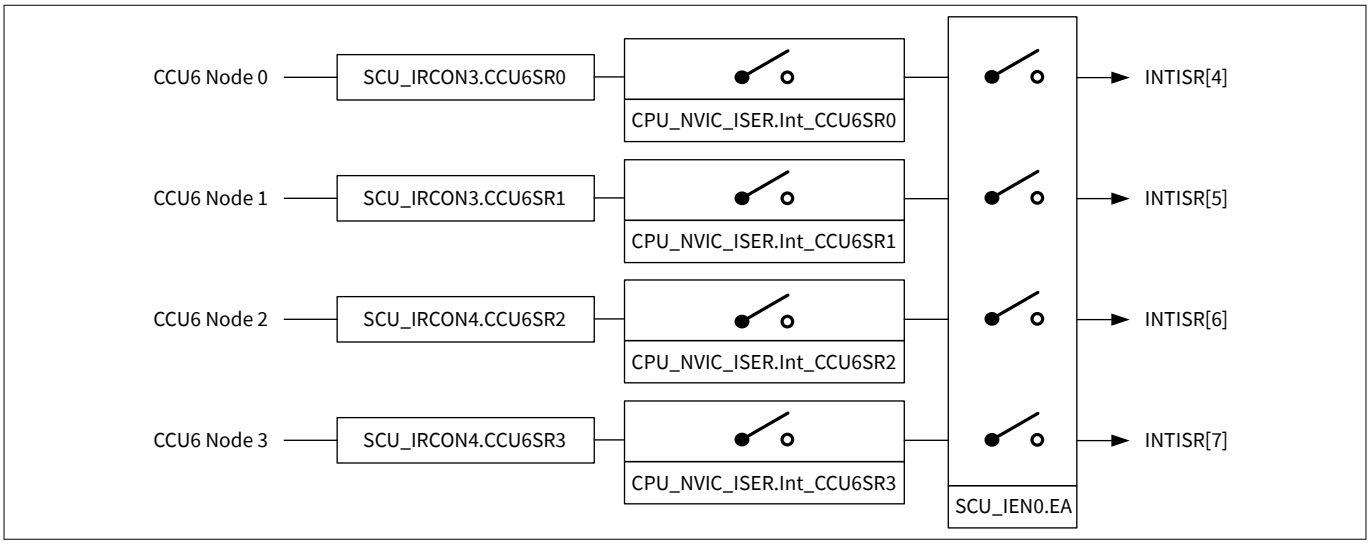


Figure 60 Interrupt request sources 4, 5, 6, 7 (CCU6)

12.3.1.5 Interrupt node 8 and 9 – SSC

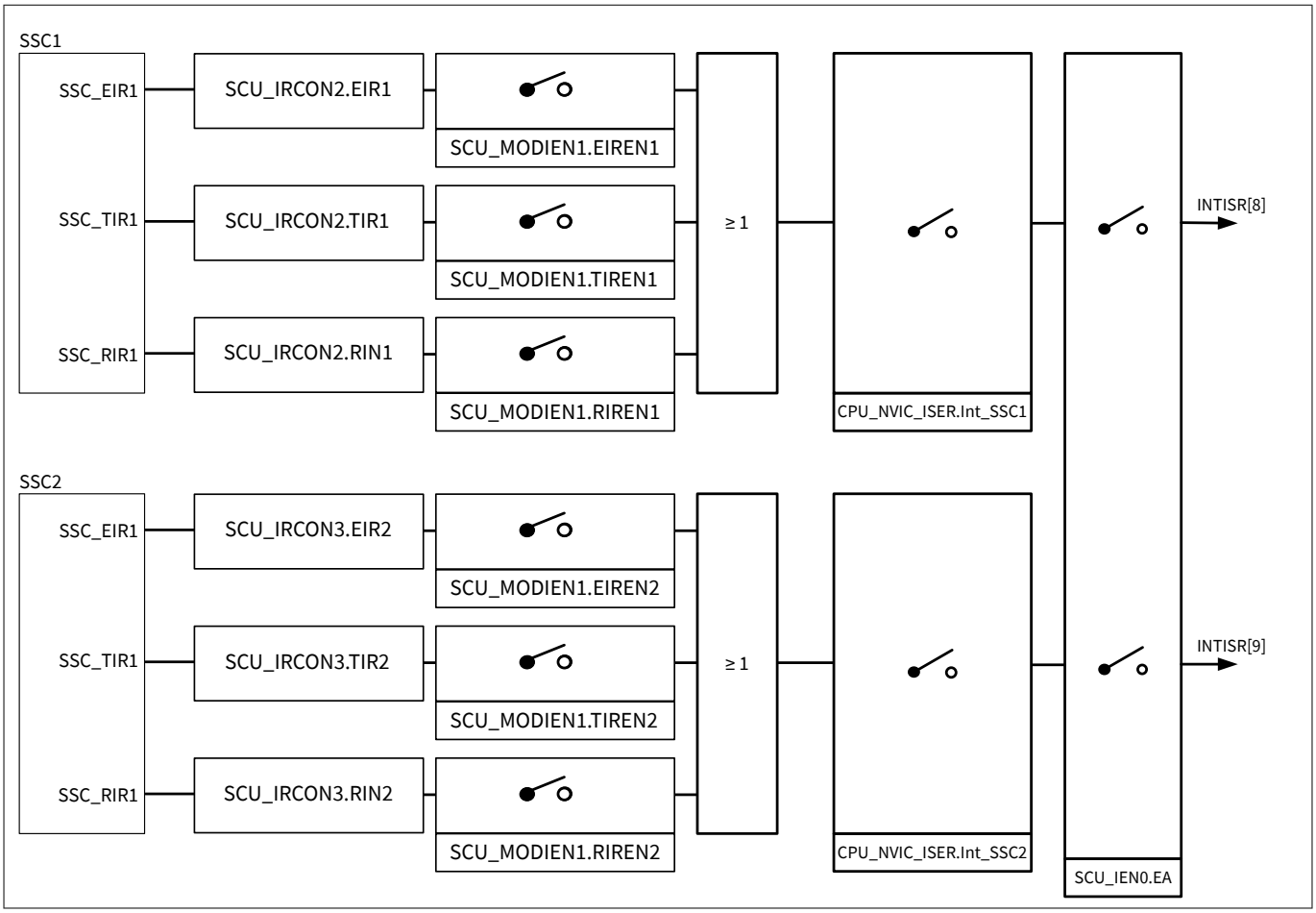


Figure 61 Interrupt request sources 8 and 9 (SSC)

12.3.1.6 Interrupt node 10 – UART1

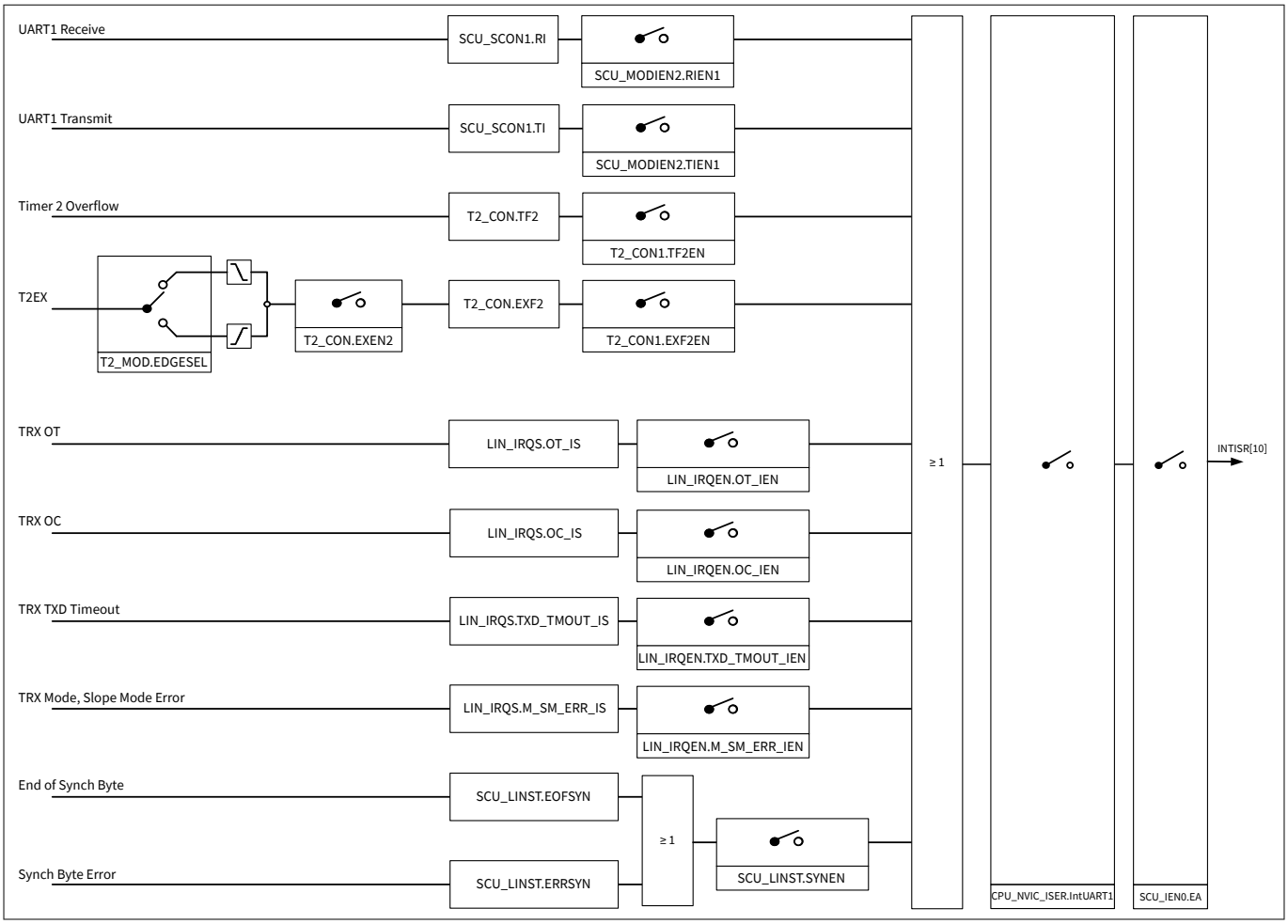


Figure 62 Interrupt request source 10 (UART1)

12.3.1.7 Interrupt node 11 – UART2

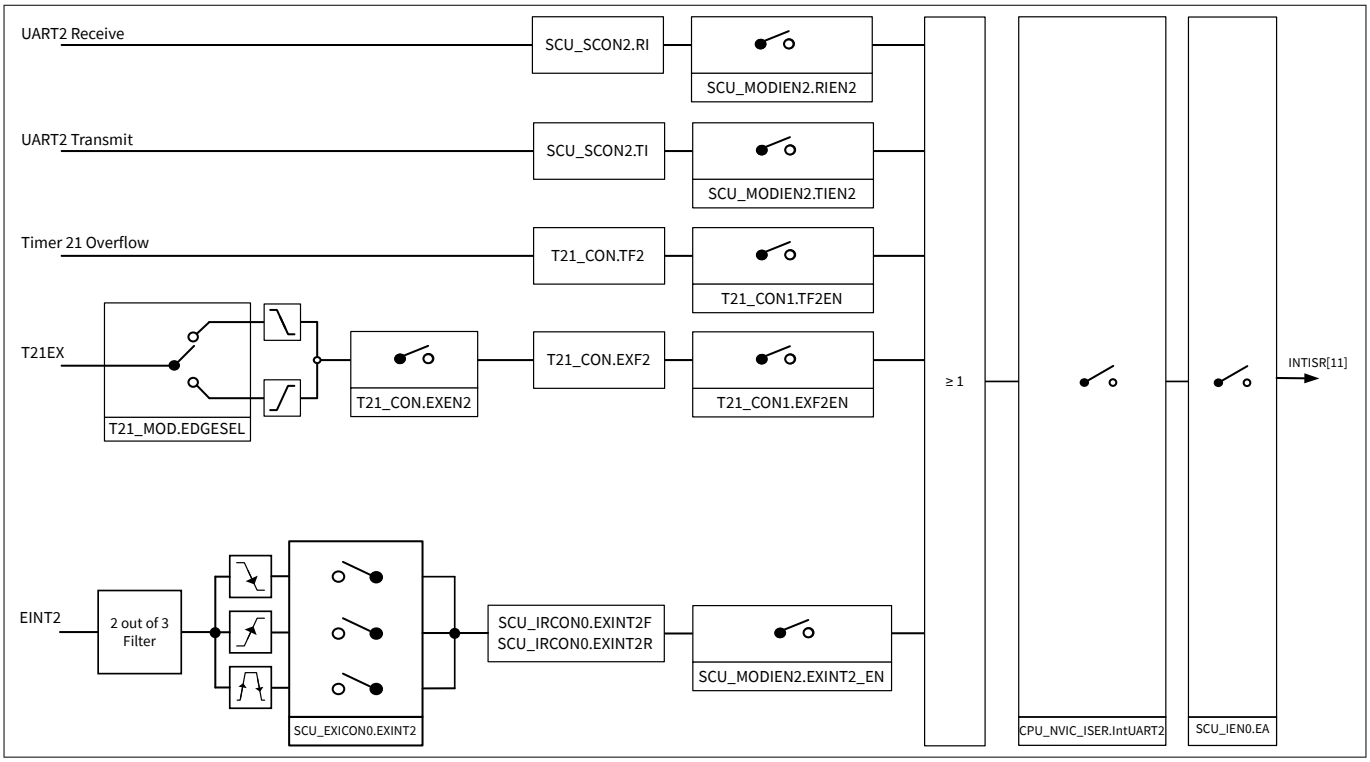


Figure 63 Interrupt request source 11 (UART2)

12.3.1.8 Interrupt node 12 and 13 – interrupt

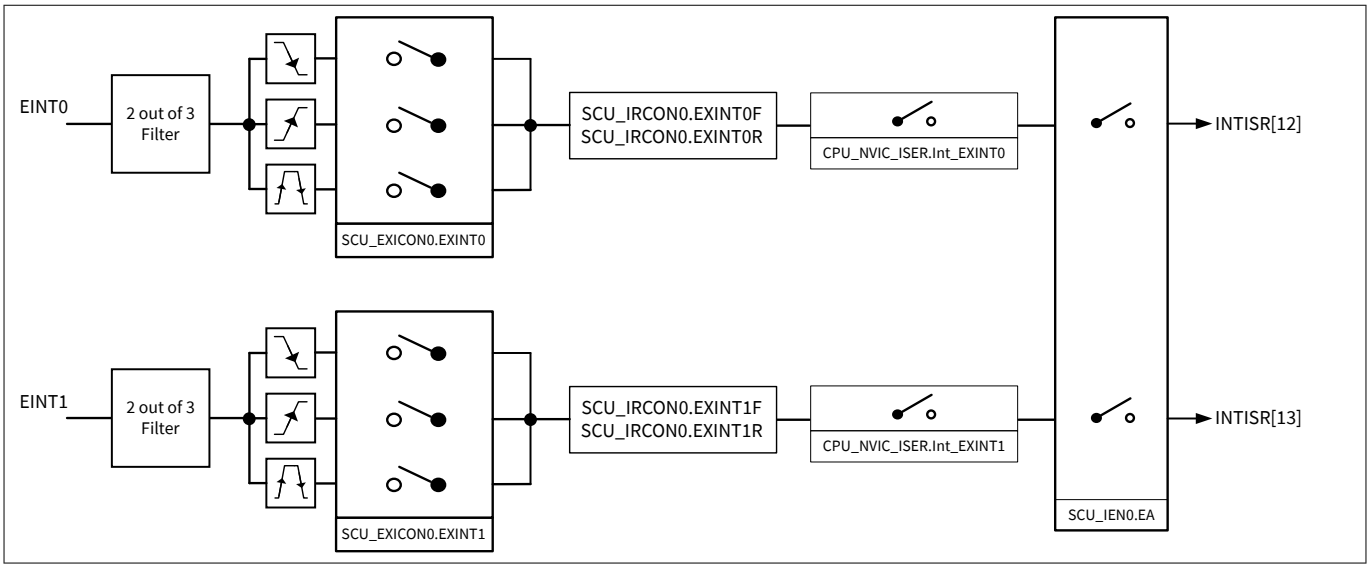


Figure 64 Interrupt request sources 12 and 13 (external interrupt)

12.3.1.9 Interrupt node 17 and 18 – LS1, LS2

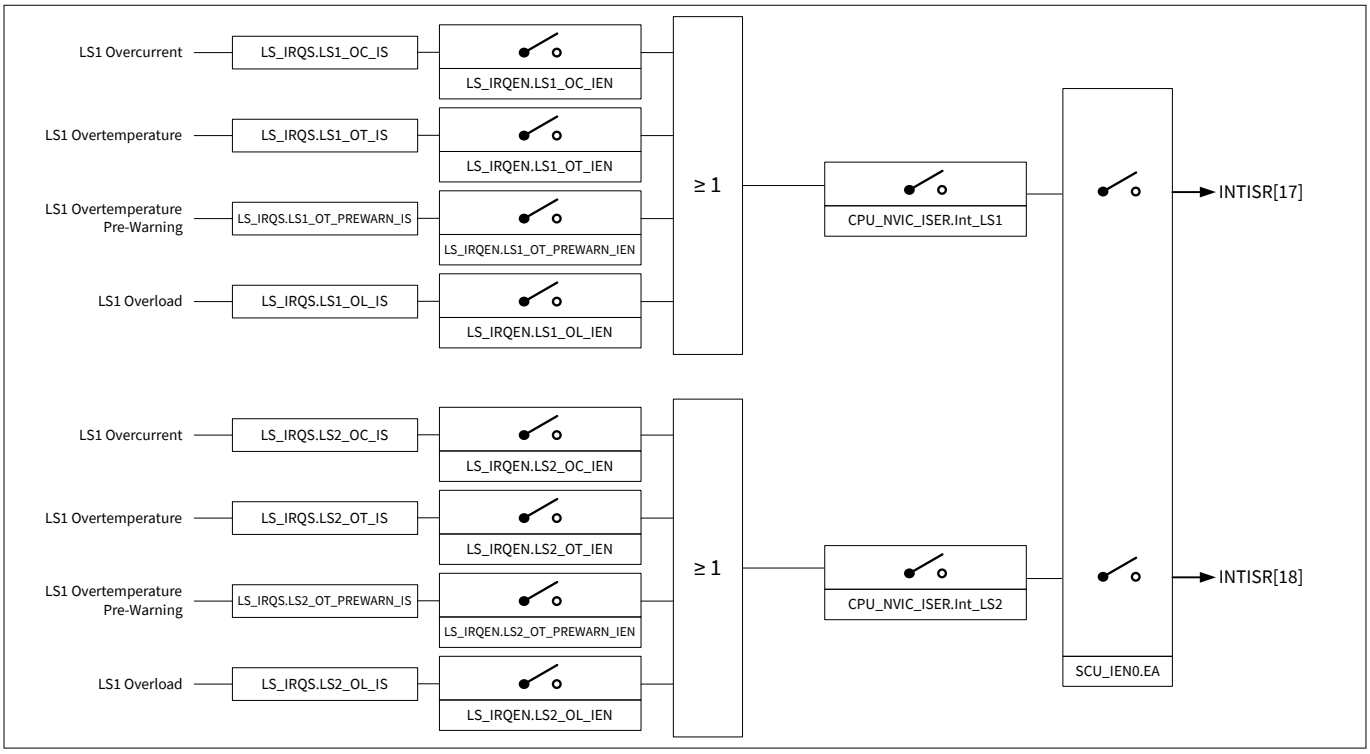


Figure 65 Interrupt request sources 17,18 (LS1, LS2)

12.3.1.10 Interrupt node 19 and 20 – HS1, HS2

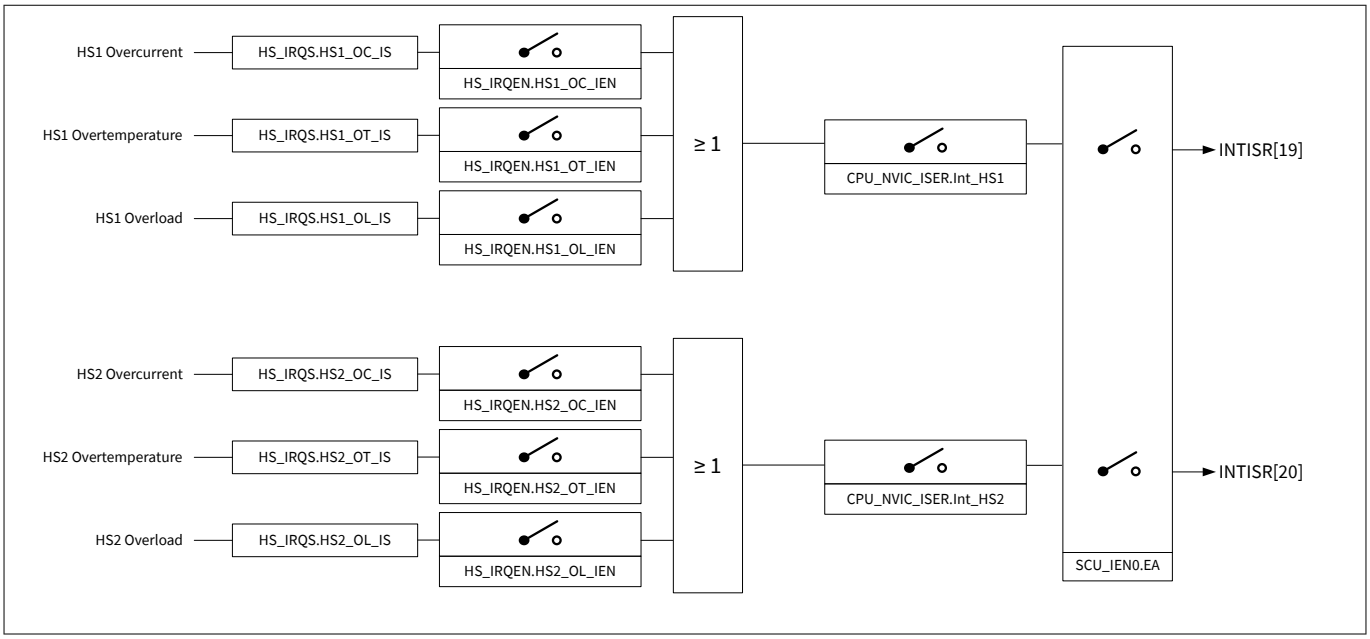


Figure 66 Interrupt request sources 19, 20 (HS1, HS2[5])

5 HS2 is device variant specific.

12.3.1.11 Interrupt node 21 – DPP1

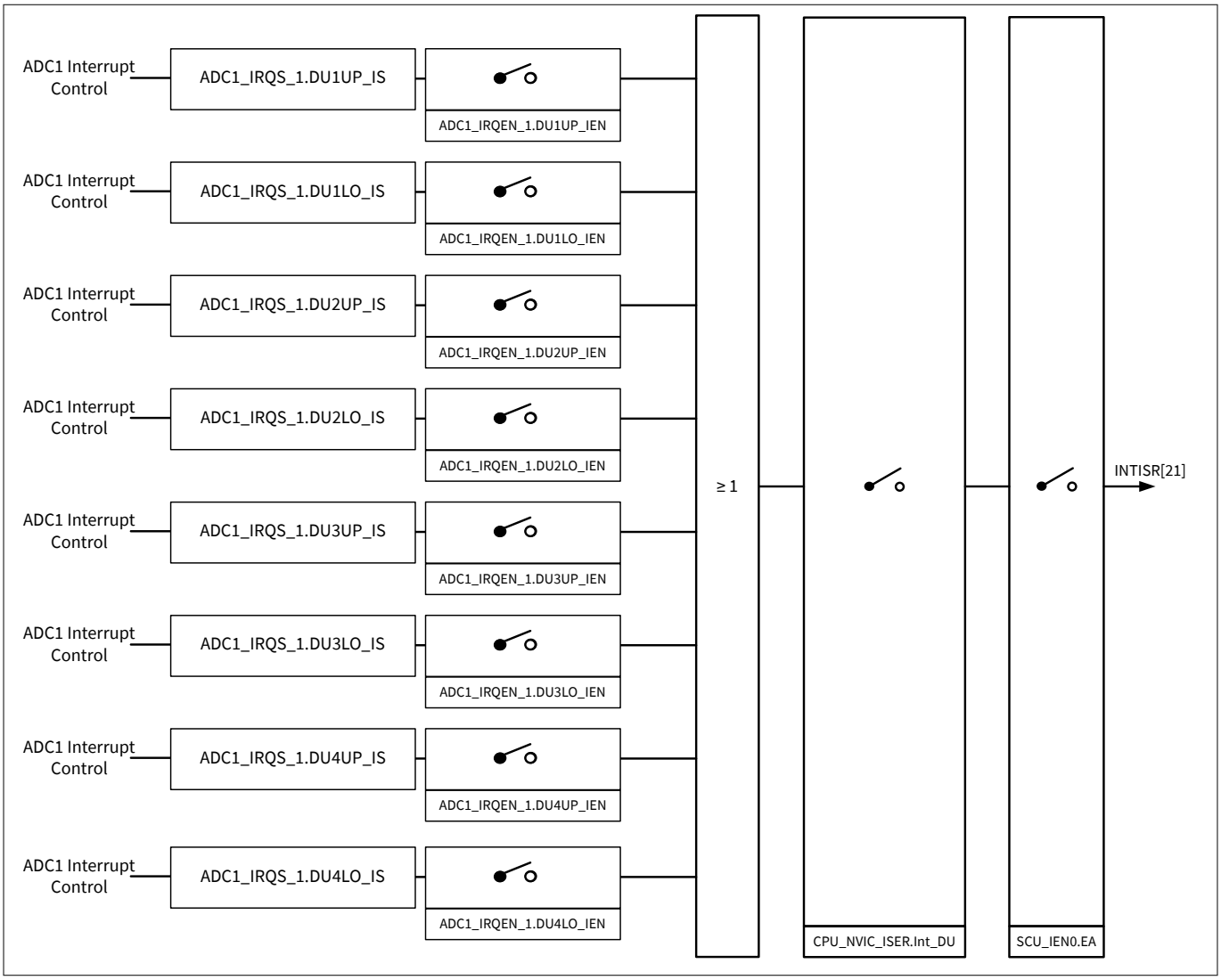


Figure 67 Interrupt request sources 21 (DPP1 – Diff Unit)[6)]

6 DU is device variant specific.

12.3.1.12 Interrupt node 22 – MON1...5

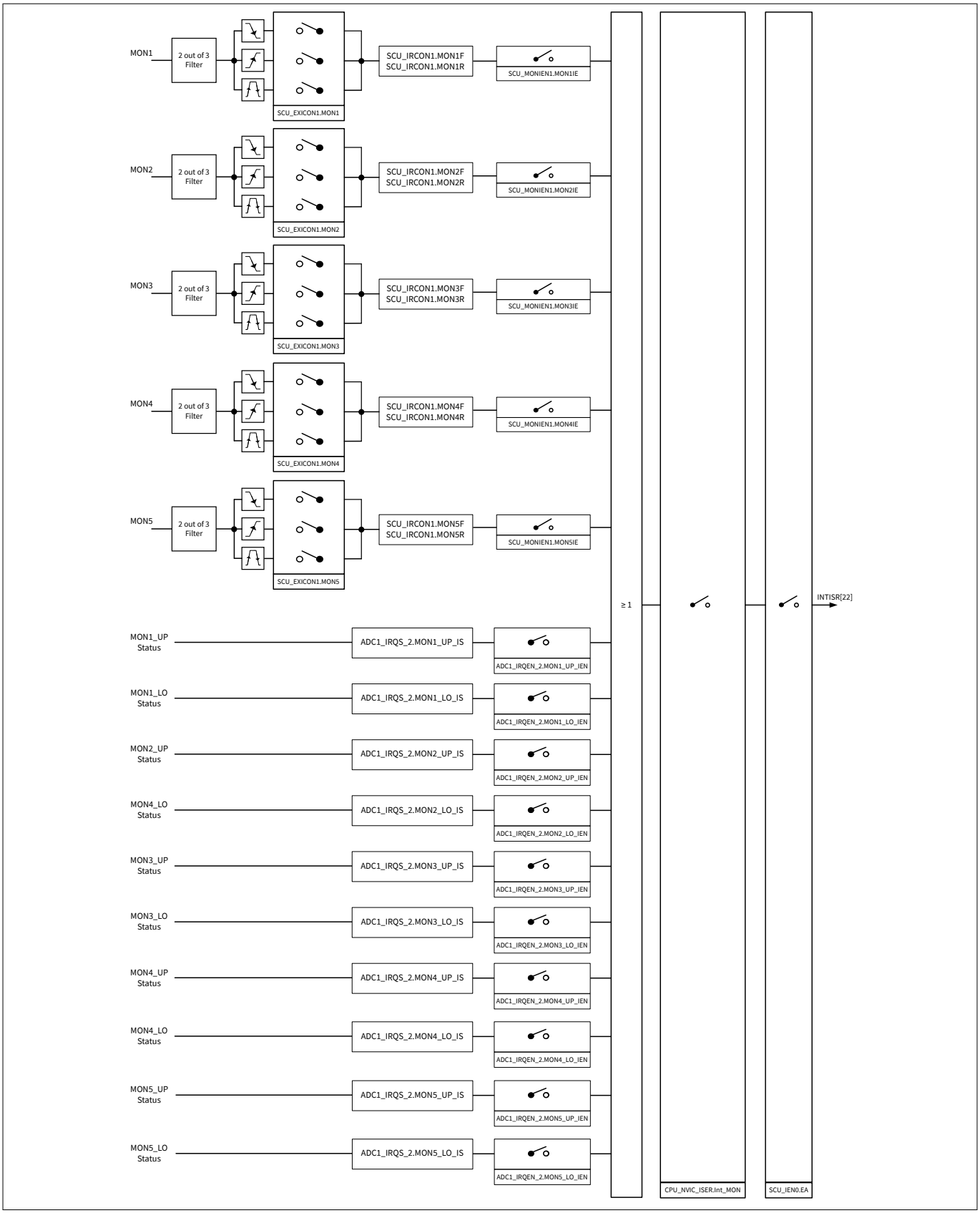


Figure 68 Interrupt request sources 22 (MON1...5)[7)]

12.3.1.13 Interrupt node 23 – Port2.x

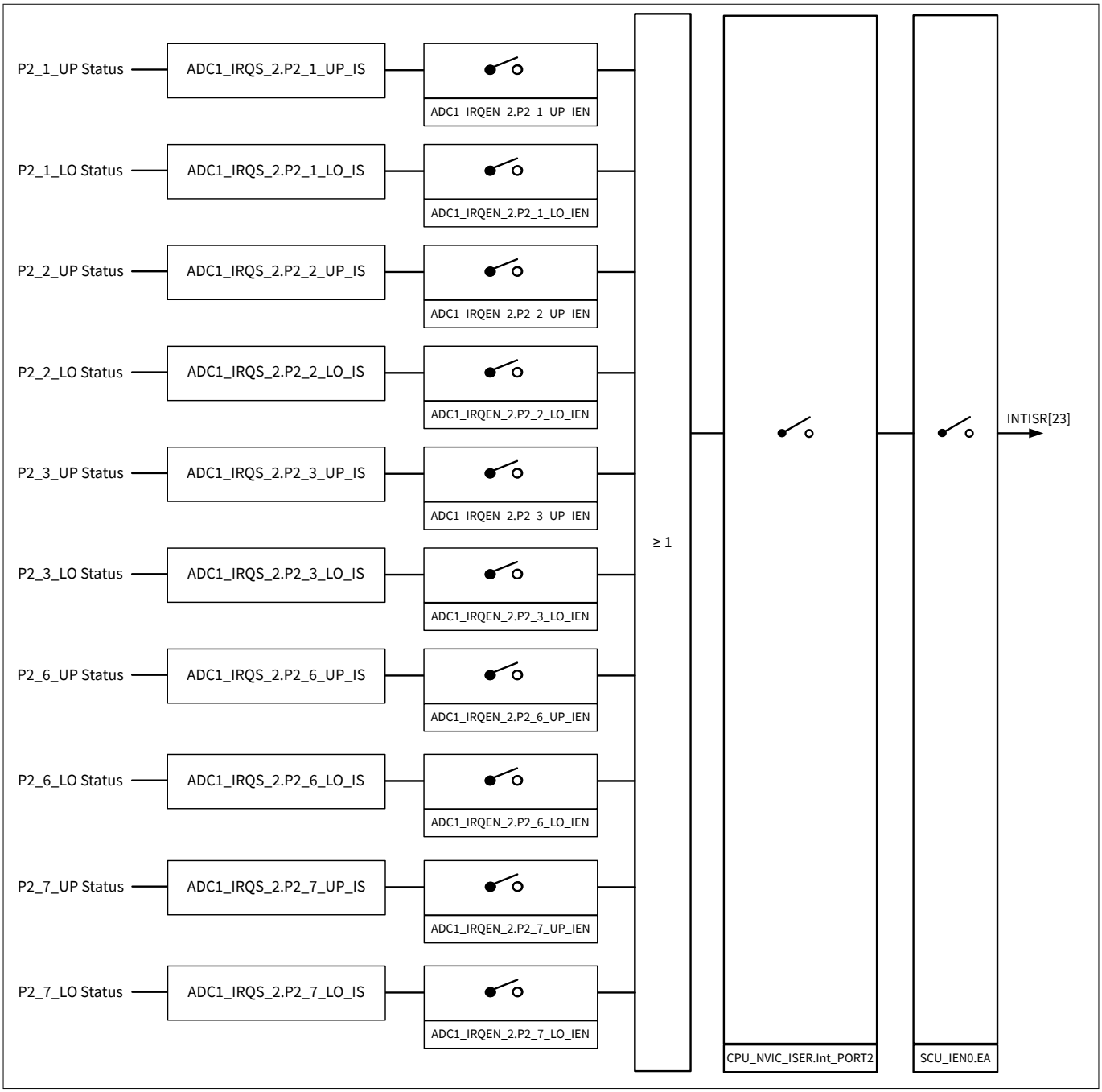


Figure 69 Interrupt request sources 23 (Port 2.x)

7 MON5 is device variant specific.

12.3.1.14 Non-maskable interrupt request source (NMI)

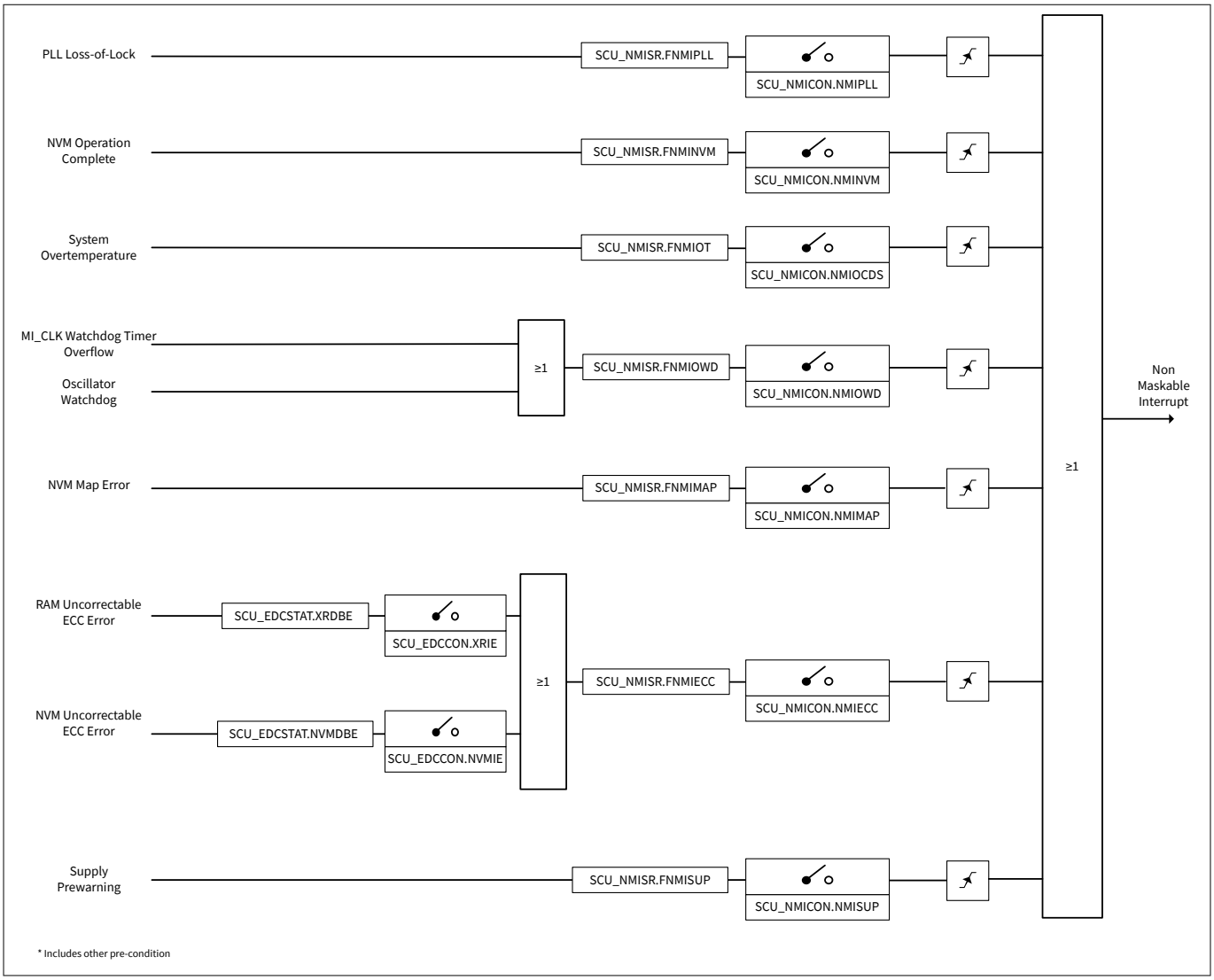


Figure 70 Non-maskable interrupt request source

12.3.1.15 Interrupt flags overview

Table 68 All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
PLL					
Loss of lock	NMI	level		SCU_APCLK_STS	Always enabled
NVM					
NVM operation complete	NMI	level			Always enabled
Overtemperature					
OT prewarning	NMI	edge		SCUPM_SYS_IS.SYS_OTWARN_IS	SCUPM_SYS_IRQ_CTRL.SYS_OTWARN_IE
OT shutdown	NMI	edge		SCUPM_SYS_IS.SYS_OT_IS	SCUPM_SYS_IRQ_CTRL.SYS_OT_IE
Oscillator watchdog					
Analog peripheral clock	NMI	level			SCU_APCLK_CTRL.CLKWDT_IE
Oscillator	NMI	level		SCU_OSC_CON.OSC2L	Always enabled
NVM					
NVM map error	NMI	level			Always enabled
ECC error					
RAM uncorrectable error	NMI	level		SCU_EDCSTAT.RDBE	SCU_EDCCON.RIE
NVM uncorrectable error	NMI	level		SCU_EDCSTAT.NVMDBE	SCU_EDCCON.NVMIE
Supply status					
VBAT undervoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VBAT_UV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VBAT_UV_IE
VBAT overvoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VBAT_OV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VBAT_OV_IE
VS undervoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VS_UV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VS_UV_IE
VS overvoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VS_OV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VS_OV_IE
VDDP undervoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VDD5V_UV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VDD5V_UV_IE
VDDP overvoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VDD5V_OV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VDD5V_OV_IE

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
VDDC undervoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VDD1V5_UV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VDD1V5_UV_IE
VDDC overvoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VDD1V5_OV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VDD1V5_OV_IE
VDDEXT undervoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VDDEXT_UV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VDDEXT_UV_IE
VDDEXT overvoltage	NMI	edge		SCUPM_SYS_SUPPLY_IRQ_STS.VDDEXT_OV_IS	SCUPM_SYS_SUPPLY_IRQ_CTRL.VDDEXT_OV_IE
PMU VDDP overvoltage	NMI	level		PMU_SUPPLY_STS.PMU_5V_OVERVOLT	PMU_SUPPLY_STS.PMU_5V_FAIL_EN
PMU VDDP overload	NMI	level		PMU_SUPPLY_STS.PMU_5V_OVERLOAD	PMU_SUPPLY_STS.PMU_5V_FAIL_EN
PMU VDDC overvoltage	NMI	level		PMU_SUPPLY_STS.PMU_1V5_OVERVOLT	PMU_SUPPLY_STS.PMU_1V5_FAIL_EN
PMU VDDC overload	NMI	level		PMU_SUPPLY_STS.PMU_1V5_OVERLOAD	PMU_SUPPLY_STS.PMU_1V5_FAIL_EN
PMU undervoltage	NMI	level		PMU_VDDEXT_CTRL.VDDEXT_UV_IS	PMU_VDDEXT_CTRL.VDDEXT_FAIL_EN
PMU overtemperature	NMI	level		PMU_VDDEXT_CTRL.VDDEXT_OT_IS	PMU_VDDEXT_CTRL.VDDEXT_FAIL_EN
INTISR<0,1> → GPT12					
GPT12-T2	0	level	2 per_clk cycles	SCU_GPT12IRC.GPT1T2	SCU_GPT12IEN.T2IE
GPT12-T3	0	level	2 per_clk cycles	SCU_GPT12IRC.GPT1T3	SCU_GPT12IEN.T3IE
GPT12-T4	0	level	2 per_clk cycles	SCU_GPT12IRC.GPT1T4	SCU_GPT12IEN.T4IE
GPT12-T5	1	level	2 per_clk cycles	SCU_GPT12IRC.GPT2T5	SCU_GPT12IEN.T5IE
GPT12-T6	1	level	2 per_clk cycles	SCU_GPT12IRC.GPT2T6	SCU_GPT12IEN.T6IE
GPT12-CR	1	level	2 per_clk cycles	SCU_GPT12IRC.GPT2CR	SCU_GPT12IEN.CRIE
INTISR<2> → MU					
REF_BG_LO	2	level	set until cleared by software	SYS_IS.REFBG_LOTHWARN_IS	SCUPM_SYS_IRQ_CTRL.REFBG_LOTHWARN_IE
REF_BG_HI	2	level		SYS_IS.REFBG_UPTHWARN_IS	SCUPM_SYS_IRQ_CTRL.REFBG_UPTHWARN_IE

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
			set until cleared by software		
INTISR<3> → ADC 10-bit					
ADC10-CH0	3	level	set until cleared by software	ADC1_IRQS_1.VBATSEN_IS	ADC1_IRQEN_1.VBATSEN_IEN
ADC10-CH1	3	level	set until cleared by software	ADC1_IRQS_1.VS_IS	ADC1_IRQEN_1.VS_IEN
ADC10-CH2	3	level	set until cleared by software	ADC1_IRQS_1.MON1_IS	ADC1_IRQN_1.MON1_IEN
ADC10-CH3	3	level	set until cleared by software	ADC1_IRQS_1.MON2_IS	ADC1_IRQN_1.MON2_IEN
ADC10-CH4	3	level	set until cleared by software	ADC1_IRQS_1.MON3_IS	ADC1_IRQN_1.MON3_IEN
ADC10-CH5	3	level	set until cleared by software	ADC1_IRQS_1.MON4_IS	ADC1_IRQN_1.MON4_IEN
ADC10-CH6	3	level	set until cleared by software	ADC1_IRQS_1.MON5_IS	ADC1_IRQN_1.MON5_IEN
ADC10-CH7	3	level	set until cleared by software	ADC1_IRQS_1.P2_1_IS	ADC1_IRQEN_1.P2_1_IEN
ADC10-CH8	3	level	set until cleared by software	ADC1_IRQS_1.P2_2_IS	ADC1_IRQEN_1.P2_2_IEN
ADC10-CH9	3	level	set until cleared by software	ADC1_IRQS_1.P2_3_IS	ADC1_IRQEN_1.P2_3_IEN
ADC10-CH10	3	level	set until cleared by software	ADC1_IRQS_1.P2_6_IS	ADC1_IRQEN_1.P2_6_IEN
ADC10-CH11	3	level	set until cleared by software	ADC1_IRQS_1.P2_7_IS	ADC1_IRQEN_1.P2_7_IEN
ADC10-CH12	3	level		ADC1_IRQS_1.P2_0_IS	ADC1_IRQEN_1.P2_0_IEN

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
			set until cleared by software		
ADC10-ESM	3	level	set until cleared by software	ADC1_IRQS_1.ESM_IS	ADC1_IRQEN_1.ESM_IEN
ADC10-EIM	3	level	set until cleared by software	ADC1_IRQS_1.EIM_IS	ADC1_IRQEN_1.EIM_IEN
ADC10-VS_LO	3	level	set until cleared by software	ADC1_IRQS_2.VS_LO_IS	ADC1_IRQEN_2.VS_LO_IEN
ADC10-VS_UP	3	level	set until cleared by software	ADC1_IRQS_2.VS_UP_IS	ADC1_IRQEN_2.VS_UP_IEN
INTISR<4,5,6,7> → CCU6					
CCU0[1)]	4	level	2 per_clk cycles	SCU_IRCON4.CCU6SR0	CPU_NVIC_ISER.Int_CCU6SR0
CCU1[1)]	5	level	2 per_clk cycles	SCU_IRCON4.CCU6SR1	CPU_NVIC_ISER.Int_CCU6SR1
CCU2[1)]	6	level	2 per_clk cycles	SCU_IRCON4.CCU6SR2	CPU_NVIC_ISER.Int_CCU6SR2
CCU3[1)]	7	level	2 per_clk cycles	SCU_IRCON4.CCU6SR3	CPU_NVIC_ISER.Int_CCU6SR3
INTISR<8,9> → SSC1/SSC2					
SSC1	8	level	2 per_clk cycles	SCU_IRCON2.EIR1	MODIEN1.EIREN1
SSC1	8	level	2 per_clk cycles	SCU_IRCON2.TIR1	MODIEN1.TIREN1
SSC1	8	level	2 per_clk cycles	SCU_IRCON2.RIR1	MODIEN1.RIREN1
SSC2	9	level	2 per_clk cycles	SCU_IRCON3.EIR2	MODIEN1.EIREN2
SSC2	9	level	2 per_clk cycles	SCU_IRCON3.TIR2	MODIEN1.TIREN2
SSC2	9	level	2 per_clk cycles	SCU_IRCON3.RIR2	MODIEN1.RIREN2
INTISR<10,11> → UART1/UART2					
UART1 receive	10	level		SCU_SCON1.RI	SCU_MODIEN2.RIEN1

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
			copy of RI bit, set until cleared by software		
UART1 transmit	10	level	copy of TI bit, set until cleared by software	SCU_SCON1.TI	SCU_MODIEN2.TIEN1
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
Timer 2	10	level		T2_CON.TF2	T2_CON1.TF2EN
T2EX	10			T2_CON.EXF2	T2_CON1.EXF2EN
LIN OT	10	edge		LIN_IRQS.OT_IS	LIN_IRQEN_OT_IEN
LIN OC	10	level		LIN_IRQS.OC_IS	LIN_IRQEN.OC_IEN
TXD_TMOUT	10			LIN_IRQS.TXD_TMOUT_IS	LIN_IRQEN.TXD_TMOUT_IEN
M_SM_ERR_IS	10			LIN_IRQS.M_SM_ERR_IS	LIN_IRQEN.M_SM_ERR_IS
UART2 receive	11	level	copy of RI bit, set until cleared by software	SCU_SCON2.RI	SCU_MODIEN2.RIEN2
UART2 transmit	11	level	copy of TI bit, set until cleared by software	SCU_SCON2.TI	SCU_MODIEN2.TIEN2
exint2	11	level		SCU_EXICON0.EXINT2	SCU_MODIEN2.EXINT2_EN
Timer21	11	level		T2_CON.TF2	T2_CON1.TF2EN
T21EX	11			T2_CON.EXF2	T2_CON1.EXF2EN
INTISR<12,13> → EXTINT0/EXTINT1					
exint0	12	level		SCU_EXICON0.EXINT0	SCU_MODIEN3.IE0
exint1	13	level		SCU_EXICON0.EXINT1	SCU_MODIEN4.IE1
INTISR<14> → Wake-up					
wake-up	14	edge			

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
				Wake: SCU_IRCON5.WAKEUP	SCU_WAKECON.WAKEUPEN
INTISR<17,18> → LS1/LS2					
LS1	17			LS_IRQS.LS1_OT_PREWARN_IS	LS_IRQEN.LS1_OT_PREWARN_IEN
LS1 OC	17	level	set until cleared by software	LS_IRQS.LS1_OC_IS	LS_IRQEN.LS1_OC_IEN
LS1 OT	17	edge	set until cleared by software	LS_IRQS.LS1_OT_IS	LS_IRQEN.LS1_OT_IEN
LS1 OL	17	edge	set until cleared by software	LS_IRQS.LS1_OL_IS	LS_IRQEN.LS1_OL_IEN
LS2	18			LS_IRQS.LS2_OT_PREWARN_IS	LS_IRQEN.LS2_OT_PREWARN_IEN
LS2 OC	18	level	set until cleared by software	LS_IRQS.LS2_OC_IS	LS_IRQEN.LS2_OC_IEN
LS2 OL	18	edge	set until cleared by software	LS_IRQS.LS2_OT_IS	LS_IRQEN.LS2_OT_IEN
LS2 OT	18	edge	set until cleared by software	LS_IRQS.LS2_OL_IS	LS_IRQEN.LS2_OL_IEN
INTISR<19,20**> → HS1**/HS2[2)]					
HS1 OC	19	level	set until cleared by software	HS_1_IS.OC_IS	HS_1_IEN.OC_IEN
HS1 OT	19	edge	set until cleared by software	HS_1_IS.OT_IS	HS_1_IEN.OT_IEN
HS1 OL	19	edge	set until cleared by software	HS_1_IS.OL_IS	HS_1_IEN.OL_IEN
HS2 OC	20	level	set until cleared by software	HS_2_IS.OC_IS	HS_2_IEN.OC_IEN
HS2 OT	20	edge	set until cleared by software	HS_2_IS.OT_IS	HS_2_IEN.OT_IEN

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
HS2 OL	20	edge	set until cleared by software	HS_2_IS.OL_IS	HS_2_IEN.OL_IEN
INTISR<21> → DU[3]					
DU1	21			ADC1_IRQS_1.DU1UP_IS ADC1_IRQS_1.DU1LO_IS	ADC1_IRQEN_1.DU1UP_IEN ADC1_IRQEN_1.DU1LO_IEN
DU2	21			ADC1_IRQS_1.DU2UP_IS ADC1_IRQS_1.DU2LO_IS	ADC1_IRQEN_1.DU2UP_IEN ADC1_IRQEN_1.DUZ2LO_IEN
DU3	21			ADC1_IRQS_1.DU3UP_IS ADC1_IRQS_1.DU3LO_IS	ADC1_IRQEN_1.DU3UP_IEN ADC1_IRQEN_1.DU3LO_IEN
DU4	21			ADC1_IRQS_1.DU4UP_IS ADC1_IRQS_1.DU4LO_IS	ADC1_IRQEN_1.DU4UP_IEN ADC1_IRQEN_1.DU4LO_IEN
INTISR<22> → Wake-up					
WAKEUP	22		set until cleared by software	SCU_EXICON1.MON1	SCU_MONIEN.MON1IE
WAKEUP	22		set until cleared by software	SCU_EXICON1.MON2	SCU_MONIEN.MON2IE
WAKEUP	22		set until cleared by software	SCU_EXICON1.MON3	SCU_MONIEN.MON3IE
WAKEUP	22		set until cleared by software	SCU_EXICON1.MON4	SCU_MONIEN.MON4IE
WAKEUP[4]			set until cleared by software	SCU_EXICON1.MON5	SCU_MONIEN.MON5IE
MON1	22			ADC1_IRQS_2.MON1_UP_IS ADC1_IRQS_2.MON1_LO_IS	ADC1_IRQEN_2.MON1_UP_IEN ADC1_IRQEN_2.MON1_LO_IEN
MON2	22				

(table continues...)

Table 68 (continued) All interrupt flags and enable

Service request	Node ID	Level/edge sensitive	Duration	SFR flag	Interrupt enable
				ADC1_IRQS_2.MON2_UP_IS ADC1_IRQS_2.MON2_LO_IS	ADC1_IRQEN_2.MON2_UP_IEN ADC1_IRQEN_2.MON2_LO_IEN
MON3	22			ADC1_IRQS_2.MON3_UP_IS ADC1_IRQS_2.MON3_LO_IS	ADC1_IRQEN_2.MON3_UP_IEN ADC1_IRQEN_2.MON3_LO_IEN
MON4	22			ADC1_IRQS_2.MON4_UP_IS ADC1_IRQS_2.MON4_LO_IS	ADC1_IRQEN_2.MON4_UP_IEN ADC1_IRQEN_2.MON4_LO_IEN
MON5[4)]	22			ADC1_IRQS_2.MON5_UP_IS ADC1_IRQS_2.MON5_LO_IS	ADC1_IRQEN_2.MON5_UP_IEN ADC1_IRQEN_2.MON5_LO_IEN
INTISR<23> → Port2					
P2_1	23			ADC1_IRQS_2.P2_1_UP_IS ADC1_IRQS_2.P2_1_LO_IS	ADC1_IRQEN_2.P2_1_UP_IEN ADC1_IRQEN_2.P2_1_LO_IEN
P2_2	23			ADC1_IRQS_2.P2_2_UP_IS ADC1_IRQS_2.P2_2_LO_IS	ADC1_IRQEN_2.P2_2_UP_IEN ADC1_IRQEN_2.P2_2_LO_IEN
P2_3	23			ADC1_IRQS_2.P2_3_UP_IS ADC1_IRQS_2.P2_3_LO_IS	ADC1_IRQEN_2.P2_3_UP_IEN ADC1_IRQEN_2.P2_3_LO_IEN
P2_6	23			ADC1_IRQS_2.P2_6_UP_IS ADC1_IRQS_2.P2_6_LO_IS	ADC1_IRQEN_2.P2_6_UP_IEN ADC1_IRQEN_2.P2_6_LO_IEN
P2_7	23			ADC1_IRQS_2.P2_7_UP_IS ADC1_IRQS_2.P2_7_LO_IS	ADC1_IRQEN_2.P2_7_UP_IEN ADC1_IRQEN_2.P2_7_LO_IEN

1) Each CCU6 interrupt can be assigned to any of the CCU6 interrupt nodes [3:0] via CCU6 registers INPL/INPH.
2) HS2 is device variant specific.
3) DU is device variant specific.

4) MON5 is device variant specific.

12.4 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.

The interrupt masking bit, EA, 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.

12.4.1 Interrupt structure 1

For interrupt structure 1 (see Figure 71), 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).

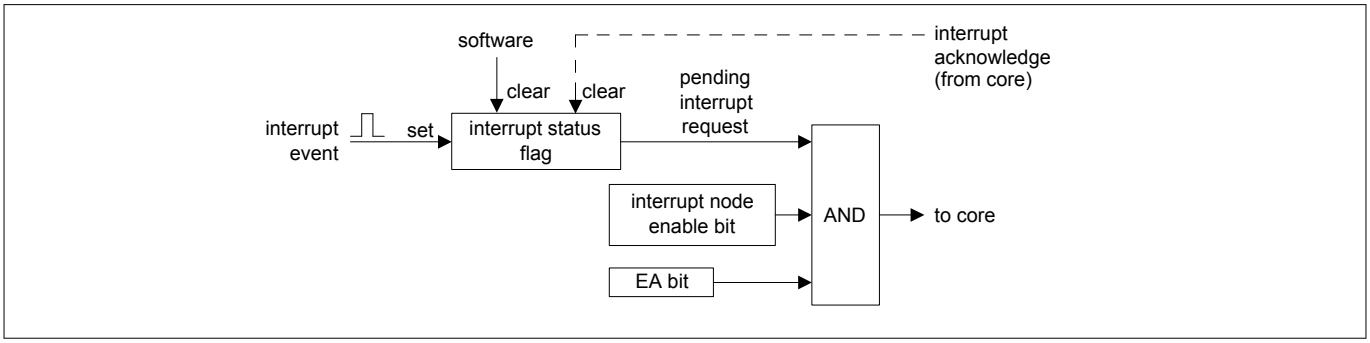


Figure 71 Interrupt structure 1

For the MOTIX™ TLE984xQX, interrupt sources like ADC10 and MU(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 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.

For interrupts related to edge-detection the behaviour is slightly different, when interrupts are disabled and re-enabled. This behavior occurs at MON (node 22) and EXTINT (node 12, 13, shared node 11). The event (detected edge) is stored in register bits (in IRCON0 or IRCON1) as for other interrupts as well. But the signaling to the core is done directly from the event, not from the register bit (see also the figures in Chapter 12.3.1). While the interrupt is disabled (e.g. via EA-bit), incoming edges are stored in the register bits, and not signaled to the core. When re-enabling the interrupt, the missed event is visible in the bit, but no re-signaled to the core.

12.5 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 MOTIX™ TLE984xQX interrupt sources to the interrupt vector address and the corresponding interrupt node enable bits are summarized in Table 66.

Table 69 Interrupt vector address

Interrupt node	Assignment for MOTIX™ TLE984xQX	Enable bit	SFR
NMI	PLL	NMIPLL	SCU_NMICON
	NVM operation complete	NMINVM	
	Overtemperature	NMIOT	
	Oscillator watchdog	NMIOWD	
	NVM map error	NMIMAP	
	ECC error	NMIECC	
	Supply status	NMISUP	
INTISR[0]	GPT1_T2, GPT1_T3, GPT1_T4	Int_GPT1	CPU_NVIC_ISER
INTISR[1]	GPT2_T5, GPT2_T6, GPT2_CR	Int_GPT2	CPU_NVIC_ISER
INTISR[2]	MU	Int_ADC2	CPU_NVIC_ISER
INTISR[3]	ADC10	Int_ADC1	CPU_NVIC_ISER
INTISR[4]	CCU6 node 0	Int_CCU6SR0	CPU_NVIC_ISER
INTISR[5]	CCU6 node 1	Int_CCU6SR1	CPU_NVIC_ISER
INTISR[6]	CCU6 node 2	Int_CCU6SR2	CPU_NVIC_ISER
INTISR[7]	CCU6 node 3	Int_CCU6SR3	CPU_NVIC_ISER
INTISR[8]	SSC1	Int_SSC1	CPU_NVIC_ISER
INTISR[9]	SSC2	Int_SSC2	CPU_NVIC_ISER
INTISR[10]	UART1	Int_UART1	CPU_NVIC_ISER
INTISR[11]	UART2	Int_UART2	CPU_NVIC_ISER
INTISR[12]	EINT0	Int_EXINT0	CPU_NVIC_ISER
INTISR[13]	EINT1	Int_EXINT1	CPU_NVIC_ISER
INTISR[14]	Wake	Int_WAKEUP	CPU_NVIC_ISER
INTISR[17]	LS1	Int_LS1	CPU_NVIC_ISER
INTISR[18]	LS2	Int_LS2	CPU_NVIC_ISER
INTISR[19]	HS1	Int_HS1	CPU_NVIC_ISER
INTISR[20]	HS2[1)]	Int_HS2	CPU_NVIC_ISER
INTISR[21]	DU[2)]	Int_DU	CPU_NVIC_ISER

(table continues...)

Table 69 (continued) Interrupt vector address

Interrupt node	Assignment for MOTIX™ TLE984xQX	Enable bit	SFR
INTISR[22]	MON	Int_MON	CPU_NVIC_ISER
INTISR[23]	Port2	Int_PORT2	CPU_NVIC_ISER

1) HS2 is device variant specific.
2) DU is device variant specific.

12.6 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 68.

Table 70 Interrupt node table

Service request	Node ID	Description
GPT1	0	GPT1 interrupt (T2-T4)
GPT2	1	GPT2 interrupt (T5-T6, CR)
MU	2	Measurement unit/ADC2, VBG
ADC1	3	ADC 10 bit interrupt
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), t2, linsync1, LIN
uart2	11	UART2 interrupt (receive, transmit), t21, linsync2, external interrupt (EINT2)
exint0	12	External interrupt (EINT0)
exint1	13	External interrupt (EINT1)
wakeup	14	Wake-up interrupt
LS1	17	Low-side driver 1
LS2	18	Low-side driver 2
HS1	19	High-side driver 1
HS2	20	High-side driver 2 (HS2 is device variant specific)
MONx	22	MONx interrupt – DPP1

(table continues...)

Table 70 (continued) Interrupt node table

Service request	Node ID	Description
Port 2.x	23	Port 2.x interrupt – DPP1

For further description see ARM_Architecture_v7n_Reference_Manual.

12.6.1 Interrupt priority registers

The interrupt priority is configured in the corresponding NVIC control register (CPU_NVIC_IPRx), located in the Arm® Cortex®-M0 core module.

Each interrupt node can be individually programmed to one of the 4 priority levels available.

12.7 Interrupt handling

See also ARM_Architecture_v7n_Reference_Manual. The most important interrupt registers are CPU_NVIC_ISER, CPU_NVIC_ICER, CPU_NVIC_ISPR and CPU_NVIC_ICPR, located in the Arm® Cortex®-M0 core module. 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 - digital modules (SCU-DM) module or the System control unit - power modules (SCU-PM) module.

12.8 Interrupt (SCU) registers

Interrupt registers are used for interrupt node enable, external interrupt control, interrupt flags and interrupt priority setting.

The registers are addressed wordwise.

12.8.1 Register overview - Interrupt management registers (ascending offset address)

Table 71 Register overview - Interrupt management registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCU_NMISRCLR	NMI status clear register	0000_H	173
SCU_IRCON0	Interrupt request 0 register	0004_H	175
SCU_IRCON1	Interrupt request 1 register	0008_H	176
SCU_IRCON2	Interrupt request 2 register	$000C_H$	178
SCU_IRCON3	Interrupt request 3 register	0010_H	179
SCU_IRCON4	Interrupt request 4 register	0014_H	180
SCU_NMISR	NMI status register	0018_H	181
SCU_IEN0	Interrupt enable 0 register	$001C_H$	183
SCU_VTOR	Vector table reallocation register	0020_H	184
SCU_NMICON	NMI control register	0024_H	185
SCU_EXICON0	External interrupt control 0 register	0028_H	186
SCU_EXICON1	External interrupt control 1 register	$002C_H$	187
SCU_MODIEN1	Peripheral interrupt enable 1 register	0030_H	188
SCU_MODIEN2	Peripheral interrupt enable 2 register	0034_H	189
SCU_MODIEN3	Peripheral interrupt enable 3 register	0038_H	190
SCU_MODIEN4	Peripheral interrupt enable 4 register	$003C_H$	191
SCU_WAKECON	Wake-up interrupt control register	0078_H	192
SCU_IRCON5	Interrupt request 5 register	$007C_H$	193
SCU_GPT12IEN	General purpose timer 12 interrupt enable register	$015C_H$	194
SCU_GPT12IRC	Timer and counter control/status register	0160_H	195
SCU_IRCON0CLR	Interrupt request 0 clear register	0178_H	196
SCU_IRCON1CLR	Interrupt request 1 clear register	$017C_H$	197
SCU_GPT12ICLR	Timer and counter control/status clear register	0180_H	199
SCU_MONIEN	Monitoring input interrupt enable register	$018C_H$	200
SCU_IRCON2CLR	Interrupt request 2 clear register	0190_H	201
SCU_IRCON3CLR	Interrupt request 3 clear register	0194_H	202
SCU_IRCON4CLR	Interrupt request 4 clear register	0198_H	203
SCU_IRCON5CLR	Interrupt request 5 clear register	$019C_H$	204

12.8.2 Interrupt node enable registers

Register SCU_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 SCU_NMICON.

After reset, the enable bits in IEN0, IEN1 and NMICON are cleared to 0. This implies that all interrupt nodes are disabled by default.

12.8.3 External interrupt control registers

The external interrupts are driven into the MOTIX™ TLE984xQX from the ports. External interrupts can be positive, negative or double edge triggered. Register SCU_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 interrupt 2 share the interrupt node with other interrupt sources. Therefore in addition to the corresponding interrupt node enable, external interrupt 2 may be disabled individually, and is 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.

12.8.4 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. For a complete listing of the interrupt flags and their assignment to SFRs, refer to Table 68.

In case of software and hardware access to a flag bit at the same time, hardware will have higher priority.

13 Watchdog timer (WDT1)

13.1 Features

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 debug mode the WDT1 is disabled.

Functional features

- Watchdog timer is operating with a from the system clock (f_{SYS}) independent clock source (f_{LP_CLK})
- Windowed watchdog timer with programmable timing (16, 32, 48, …, 1008 ms period) in active mode
- Long open window (200 ms) after power-up, reset, wake-up
- Short open window (30 ms) to facilitate flash programming
- System safety shutdown to sleep mode after 5 missed WDT1 services (see Watchdog (WDT1) fail safe)
- Watchdog is disabled in debug mode
- Watchdog cannot be deactivated in normal mode
- Watchdog reset is stored in reset status register PMU_RESET_STS

13.2 Introduction

The behavior of the watchdog timer in active mode is depicted in Figure 72.

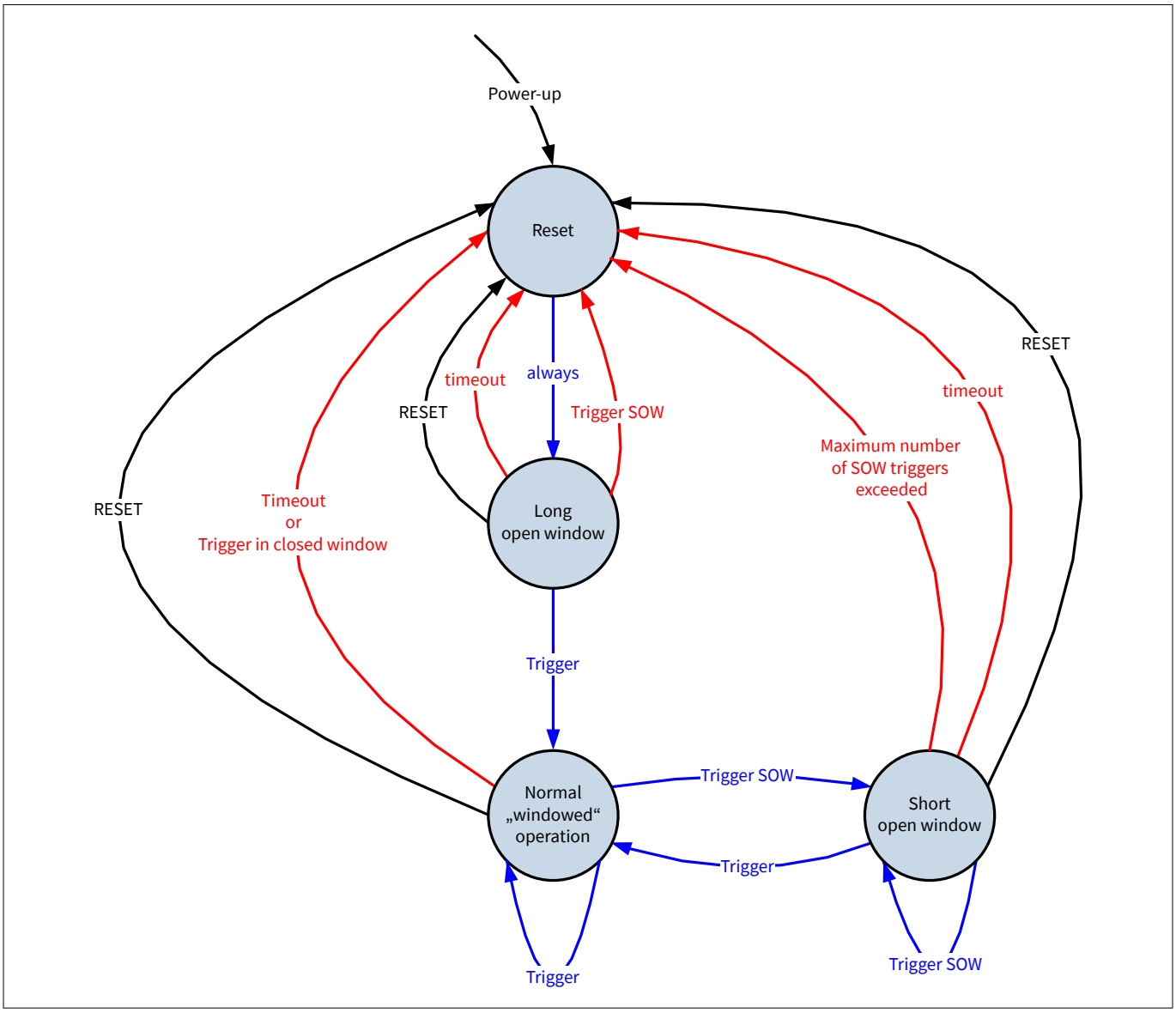


Figure 72 Watchdog timer behavior

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

The 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. If the watchdog is not served properly consecutively 5 times, the system will enter sleep mode. After an initial successful trigger the WDT1 operates in a window watchdog mode. Configuring of a short open window inside the long open window is not allowed and will also cause a WDT1 reset.

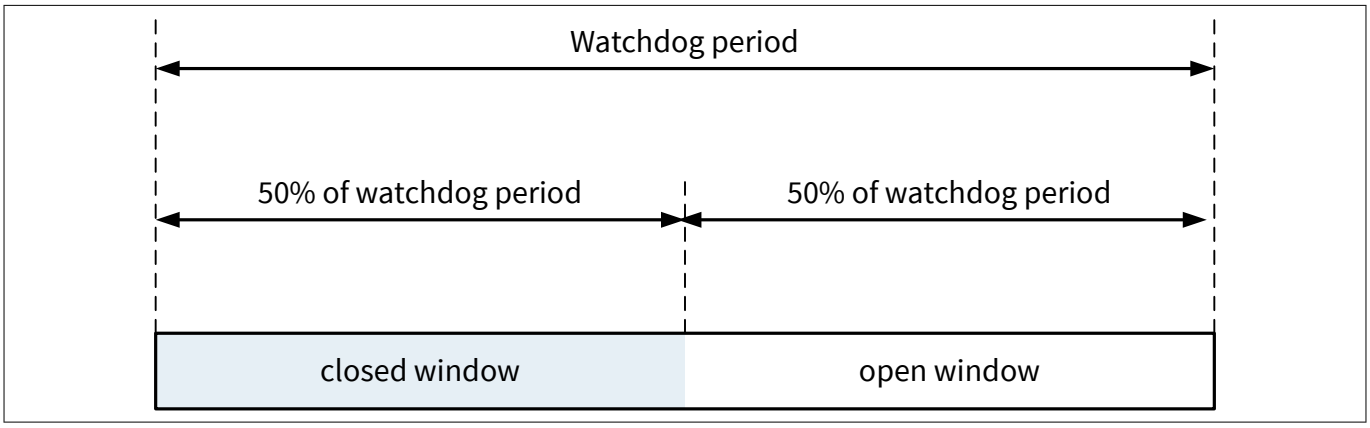


Figure 73 Windowed watchdog

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 in the reset status register PMU_RESET_STS 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 the following figure.

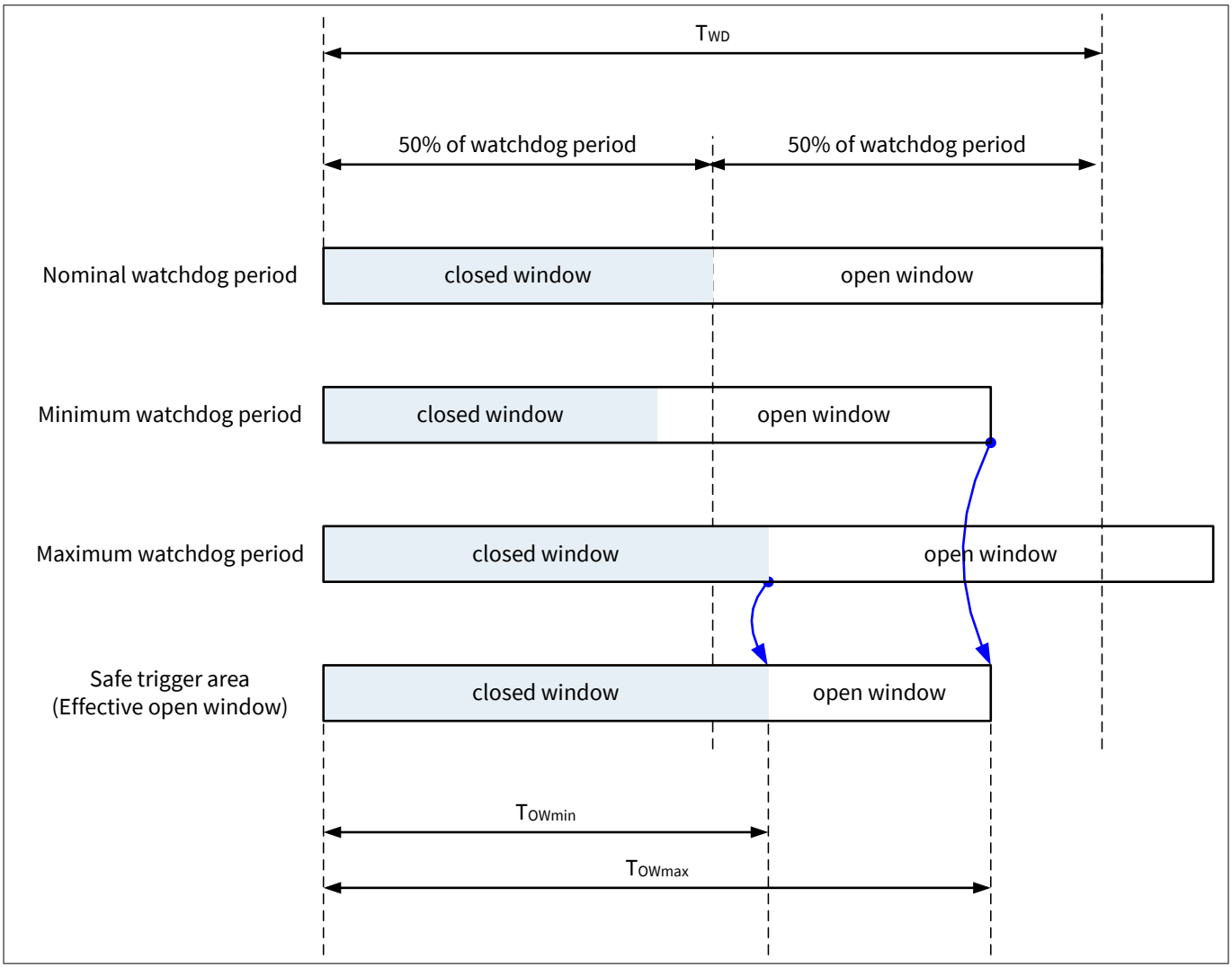


Figure 74 Effective open window

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 (SOW)

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 75) 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 independent of the settings of the WDP_SEL bits.

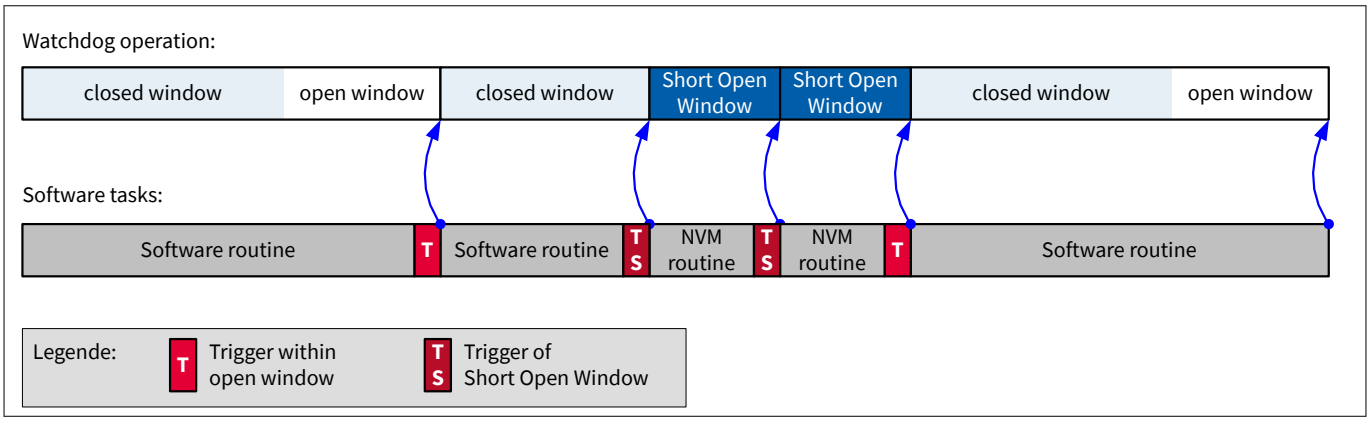


Figure 75 Short open window

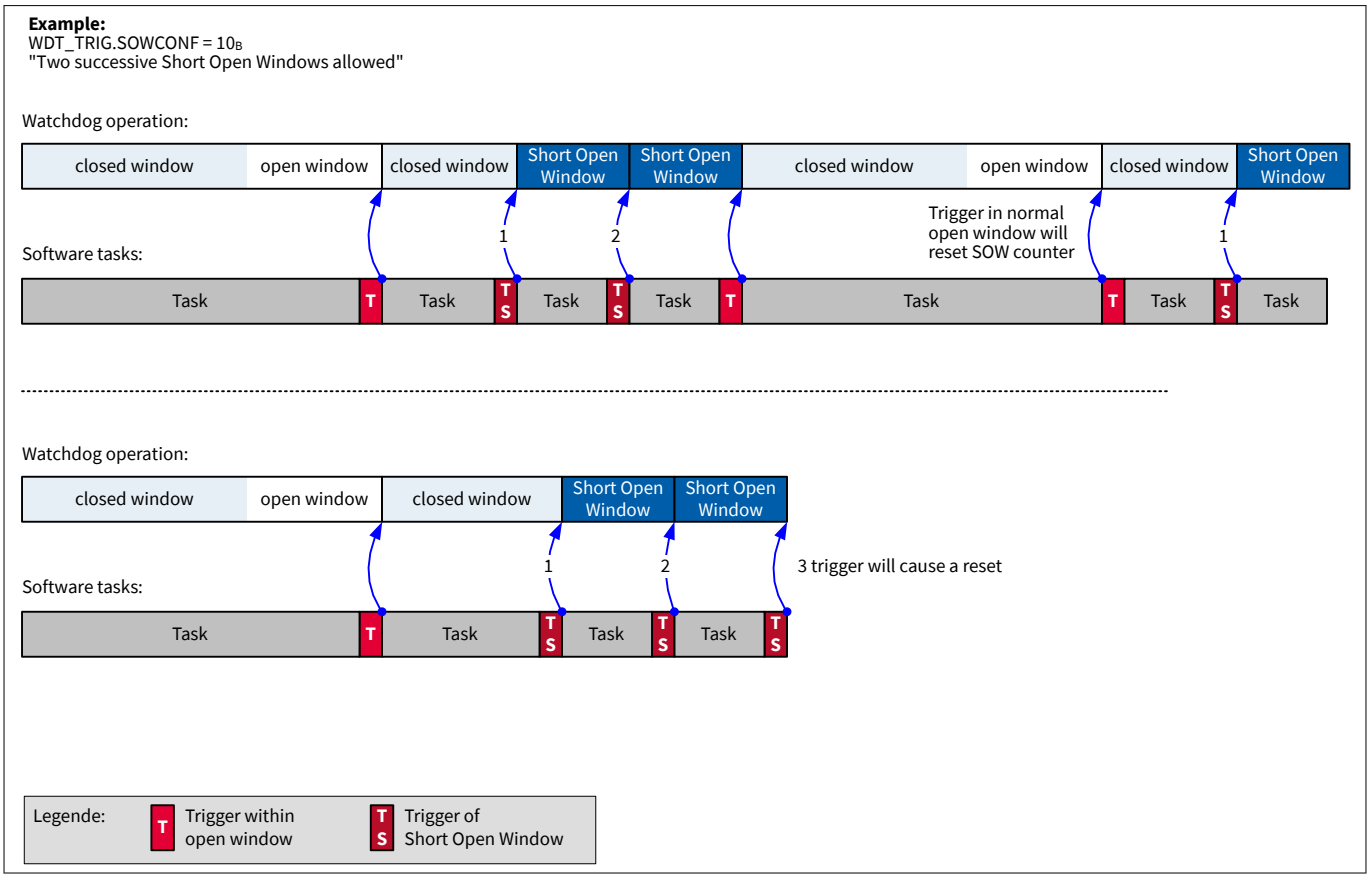


Figure 76 SOW counter

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.

13.3.3 Watchdog register

The watchdog control register SCUPM_WDT1_TRIG is located in the System control unit - power modules (SCU-PM) module.

13.3.3.1 Register overview - External Watchdog registers (ascending offset address)

Table 72 Register overview - External Watchdog registers (ascending offset address)

Short name	Long name	Offset address	Page number
SCUPM_WDT1_TRIG	WDT1 watchdog control register	0034_H	269

14 GPIO ports and peripheral I/O

This chapter describes the GPIO of the MOTIX™ TLE984xQX. It contains the following sections:

- Functional description of the GPIO Ports (see Chapter 14.2)
- MOTIX™ TLE984xQX implementation specific details and registers of the GPIO module (see Chapter 14.3)

The MOTIX™ TLE984xQX has 18 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. When configured as an output, the open drain mode can be selected. On port 2 (P2) analog inputs are shared with general purpose input.

14.1 Features

- 10 GPIOs (P0.x & P1.x), 6 analog inputs (P2.x) and two additional analog inputs shared with an XTAL feature (P2.4, P2.5).
- Strong pull-up at reset pin and Hall inputs (except P2.x)

Bidirectional port features (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

Analog port features (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 77 shows the block diagram of an MOTIX™ TLE984xQX 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 internal pull-up/pull-down device.

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 mode at any one time.

In input mode (default after reset), the output driver is switched off (high-impedance). 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 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 function is defined in registers Px_ALTSEL0 and Px_ALTSEL1. When a port pin is used as 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 the 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_DM.

The temperature compensation signals TC[1:0] of all output drivers are connected to all outputs and are controlled by register SCU_TCCR.TC[1:0] in SCU_DM.

Note: For the definition of Px_POCONy and TCCR registers, refer to Port output control registers *of SCU_DM chapter.*

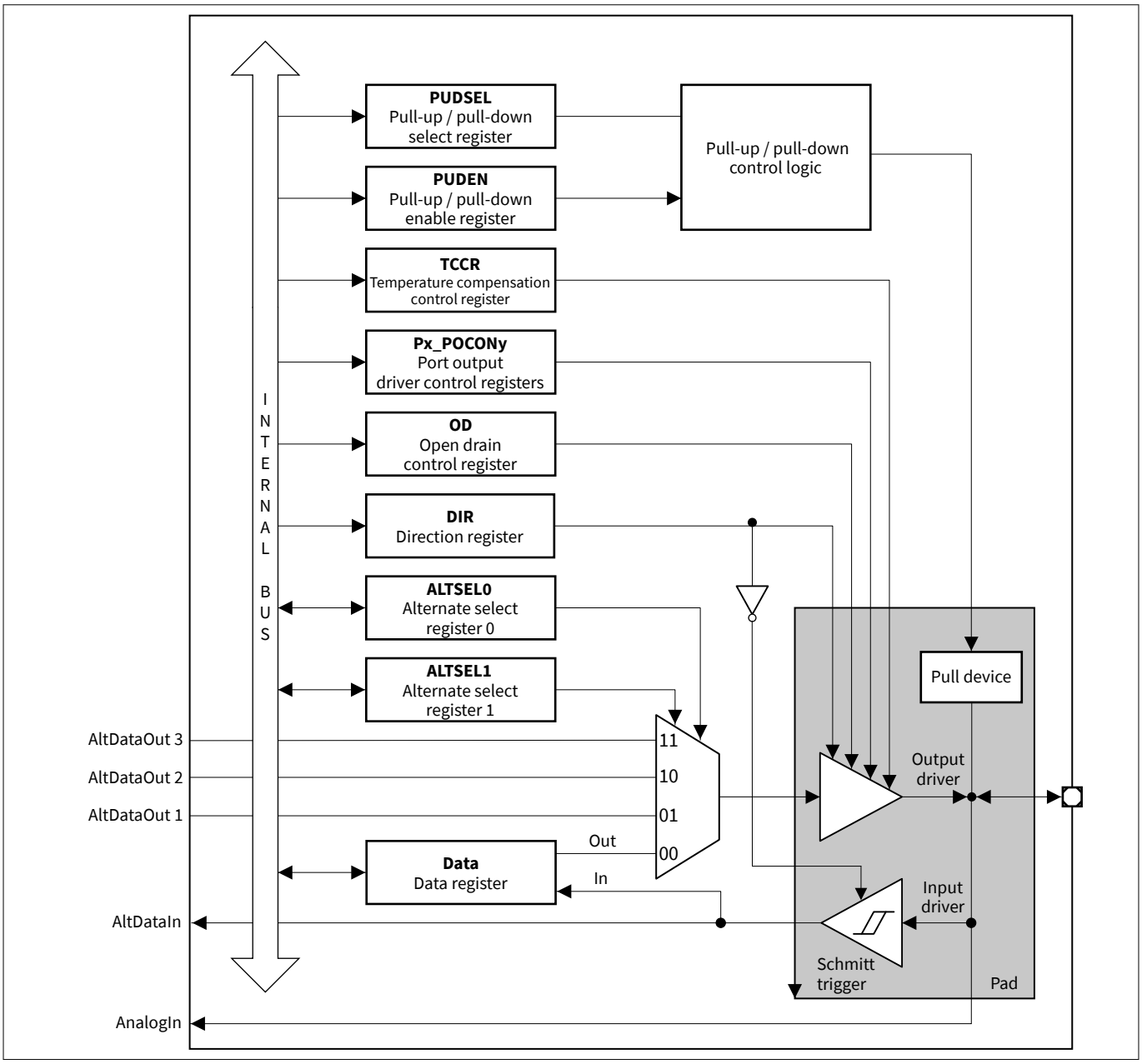


Figure 77 General structure of bidirectional port

14.2.2 Port 2

Figure 78 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 the 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 channel.

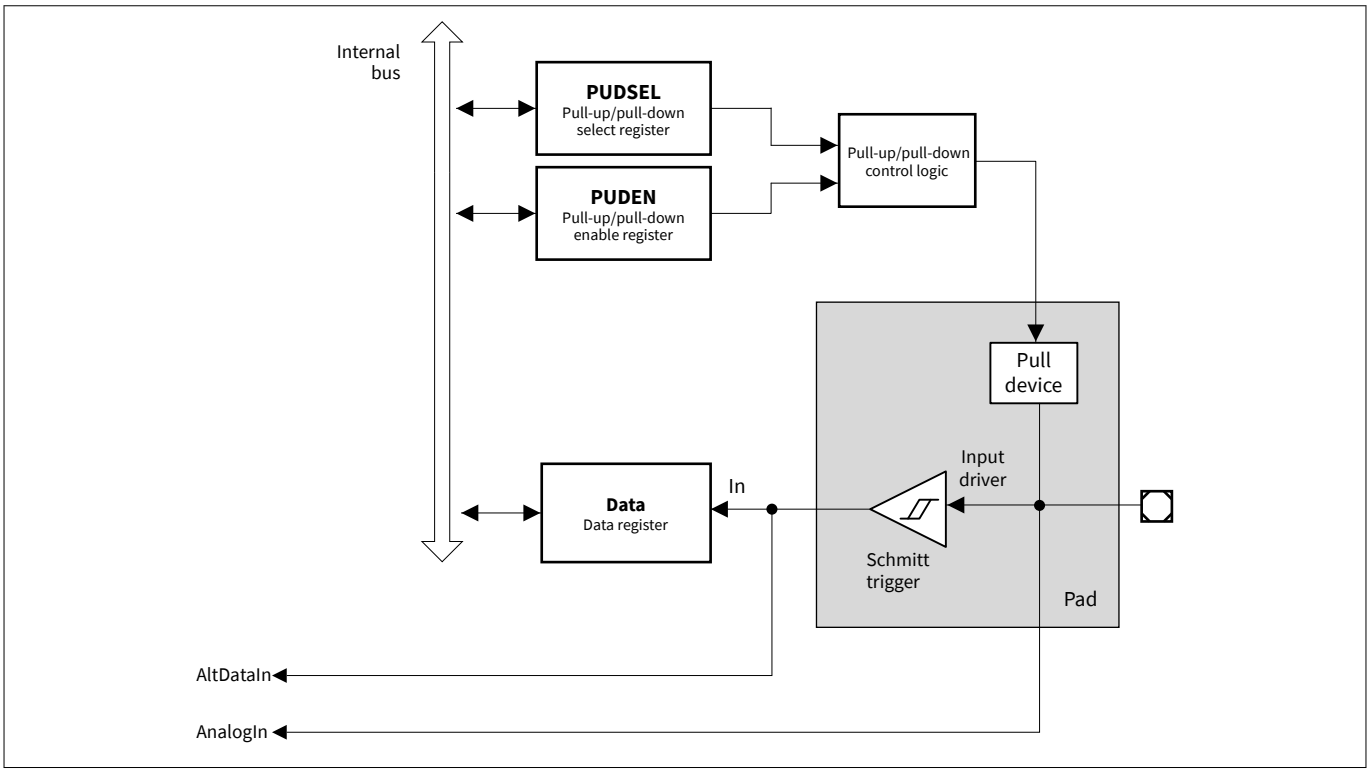


Figure 78 General structure of input port

14.3 Port implementation details

14.3.1 Port 0

14.3.1.1 Port 0 functions

Port 0 alternate function mapping according Table 73

Table 73 Port 0 input/output functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
P0.0	Input	GPI	P0_DATA.P0	
		INP1	T12HR_0	CCU6
		INP2	T4INA	GPT12
		INP3	T2_0	Timer 2
		INP4	SWD_CLK	SWD
		INP5	EXINT2_3	SCU
	Output	GPO	P0_DATA.P0	
		ALT1	T3OUT_0	GPT12
		ALT2	EXF21_0	Timer 21
		ALT3	UART2_RXDO	UART2
P0.1	Input	GPI	P0_DATA.P1	
		INP1	T13HR_0	CCU6
		INP2	UART1_RXD	UART1
		INP3	T2EX_1	Timer 2
		INP4	T21_0	Timer 21
		INP5	EXINT0_3	SCU
		INP6	T4INC	GPT12
		INP7	CAPINA	GPT12
		INP8	SSC12_S_SCK	SSC1/2
		INP9	CC62_0	CCU6
	Output	GPO	P0_DATA.P1	
		ALT1	T6OUT_0	GPT12
		ALT2	CC62_0	CCU6
		ALT3	SSC12_M_SCK	SSC1/2
P0.2	Input	GPI	P0_DATA.P2	
		INP1	T2EUDA	GPT12
		INP2	CTRAP_0	CCU6
		INP3	SSC12_M_MRST	SSC1/2

(table continues...)

Table 73 (continued) Port 0 input/output functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
		INP4	T21EX_0	Timer 21
		INP5	EXINT1_3	SCU
	Output	GPO	P0_DATA.P2	
		ALT1	SSC12_S_MRST	SSC1/2
		ALT2	UART1_TXD	UART1
		ALT3	EXF2_0	Timer 2
P0.3	Input	GPI	P0_DATA.P3	
		INP1	SSC1_S_SCK	SSC1
		INP2	T4EUDA	GPT12
		INP3	CAPINB	GPT12
		INP4	EXINT1_2	SCU
		INP5	T3EUDD	GPT12
		INP6	CCPOS0_1	CCU6
	Output	GPO	P0_DATA.P3	
		ALT1	SSC1_M_SCK	SSC1
		ALT2	T6OFL	GPT12
		ALT3	T6OUT_1	GPT12
P0.4	Input	GPI	P0_DATA.P4	
		INP1	SSC1_S_MTSR	SSC1
		INP2	CC60_0	CCU6
		INP3	T21_2	Timer 21
		INP4	EXINT2_2	SCU
		INP5	T3EUDA	GPT12
		INP6	CCPOS1_1	CCU6
	Output	GPO	P0_DATA.P4	
		ALT1	SSC1_M_MTSR	SSC1
		ALT2	CC60_0	CCU6
		ALT3	CLKOUT_0	SCU
P0.5	Input	GPI	P0_DATA.P5	
		INP1	SSC1_M_MRST	SSC1
		INP2	EXINT0_0	SCU
		INP3	T21EX_2	Timer 21
		INP4	T5INA	GPT12
		INP5	CCPOS2_1	CCU6

(table continues...)

Table 73 (continued) Port 0 input/output functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
	Output	GPO	P0_DATA.P5	
		ALT1	SSC1_S_MRST	SSC1
		ALT2	COUT60_0	CCU6
		ALT3	LIN_RXD	TRX

14.3.1.2 Overview - Port 0 registers

Port 0 is a general purpose bidirectional port. The port registers of port 0 are shown in Figure 79.

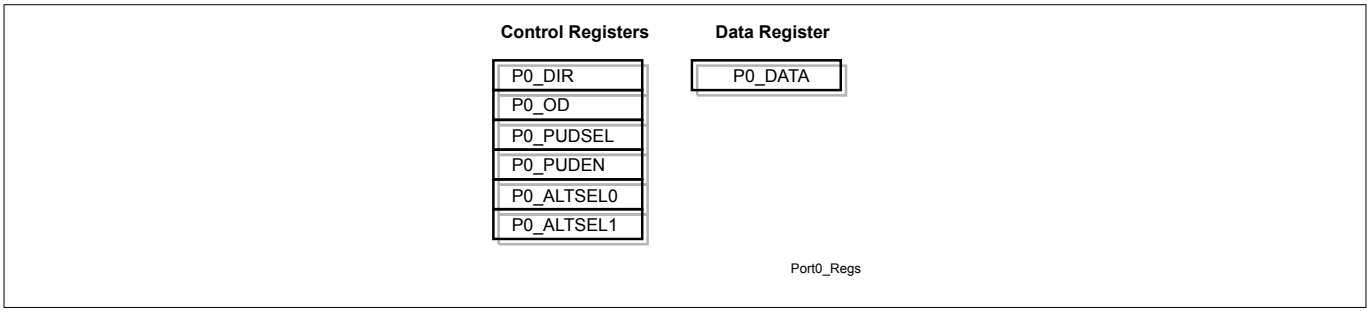


Figure 79 Port 0 registers

The registers are addressed wordwise.

14.3.1.2.1 Register overview - Port 0 registers (ascending offset address)

Table 74 Register overview - Port 0 registers (ascending offset address)

Short name	Long name	Offset address	Page number
P0_DATA	Port 0 data register	0000_H	386
P0_DIR	Port 0 direction register	0004_H	388
P0_OD	Port 0 open drain control register	0008_H	390
P0_PUDSEL	Port 0 pull-up/pull-down select register	$000C_H$	391
P0_PUDEN	Port 0 pull-up/pull-down enable register	0010_H	392
P0_ALTSEL0	Port 0 alternate select 0 register	0014_H	393
P0_ALTSEL1	Port 0 alternate select 1 register	0018_H	395

14.3.2 Port 1

14.3.2.1 Port 1 functions

Port 1 alternate function mapping according Table 75

Table 75 Port 1 input/output functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
P1.0	Input	GPI	P1_DATA.P0	
		INP1	T3INC	GPT12
		INP2	CC61_0	CCU6
		INP3	SSC2_S_SCK	SSC2
		INP4	T4EUDB	GPT12
	Output	GPO	P1_DATA.P0	
		ALT1	SSC2_M_SCK	SSC2
		ALT2	CC61_0	CCU6
		ALT3	UART2_TXD	UART2
P1.1	Input	GPI	P1_DATA.P1	
		INP1	T6EUDA	GPT12
		INP2	T5INB	GPT12
		INP3	T3EUDC	GPT12
		INP4	SSC2_S_MTSR	SSC2
		INP5	T21EX_3	Timer 21
		INP6	UART2_RXD	UART2
	Output	GPO	P1_DATA.P1	
		ALT1	SSC2_M_MTSR	SSC2
		ALT2	COUT61_0	CCU6
		ALT3	EXF21_1	Timer 21
P1.2	Input	GPI	P1_DATA.P2	
		INP1	EXINT0_1	SCU
		INP2	T21_1	Timer 21
		INP3	T2INA	GPT12
		INP4	SSC2_M_MRST	SSC2
		INP5	CCPOS2_2	CCU6
	Output	GPO	P1_DATA.P2	
		ALT1	SSC2_S_MRST	SSC2
		ALT2	COUT63_0	CCU6
		ALT3	T3OUT_1	GPT12

(table continues...)

Table 75 (continued) Port 1 input/output functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
P1.4	Input	GPI	P1_DATA.P4	
		INP1	EXINT2_1	SCU
		INP2	T21EX_1	Timer 21
		INP3	T2INB	GPT12
		INP4	T5EUDA	GPT12
		INP5	SSC12_S_MTSR	SSC1/2
		INP6	CCPOS1_2	CCU6
	Output	GPO	P1_DATA.P4	
		ALT1	CLKOUT_1	SCU
		ALT2	COUT62_0	CCU6
		ALT3	SSC12_M_MTSR	SSC1/2

14.3.2.2 Overview - Port 1 registers

Port 1 is a general purpose bidirectional port. The port registers of Port 1 are shown in Figure 80.

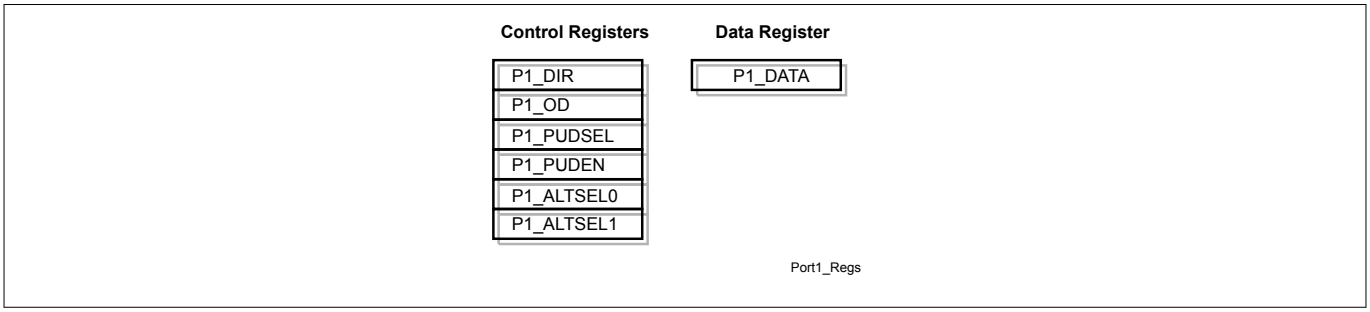


Figure 80 Port 1 registers

The registers are addressed wordwise.

14.3.2.2.1 Register overview - Port 1 registers (ascending offset address)

Table 76 Register overview - Port 1 registers (ascending offset address)

Short name	Long name	Offset address	Page number
P1_DATA	Port 1 data register	0020_H	397
P1_DIR	Port 1 direction register	0024_H	399
P1_OD	Port 1 open drain control register	0028_H	401
P1_PUDSEL	Port 1 pull-up/pull-down select register	$002C_H$	402
P1_PUDEN	Port 1 pull-up/pull-down enable register	0030_H	403
P1_ALTSEL0	Port 1 alternate select 0 register	0034_H	404
P1_ALTSEL1	Port 1 alternate select 1 register	0038_H	405

14.3.3 Port 2

14.3.3.1 Port 2 functions

Port 2 alternate function mapping according Table 77

Table 77 Port 2 input functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
P2.0	Input	GPI	P2_DATA.P0	
		INP1	EXINT1_1	SCU
		INP2	CCPOS0_2	CCU6
		INP3	T5EUDB	GPT12
		ANALOG	AN0	ADC
P2.1	Input	GPI	P2_DATA.P1	
		INP1	CCPOS0_0	CCU6
		INP2	EXINT1_0	SCU
		INP3	T12HR_1	CCU6
		INP4	CC61_1	CCU6
		ANALOG	AN1	ADC
P2.2	Input	GPI	P2_DATA.P2	
		INP1	T6EUDB	GPT12
		INP2	T2EX_0	Timer 2
		INP3	T12HR_2	CCU6
		ANALOG	AN2	ADC
P2.3	Input	GPI	P2_DATA.P3	
		INP1	CCPOS1_0	CCU6
		INP2	EXINT0_2	SCU
		INP3	CTRAP_1	CCU6
		INP4	T3IND	GPT12
		INP5	CC60_1	CCU6
		ANALOG	AN3	ADC
P2.4	Input	GPI	P2_DATA.P4	
		INP1	T2EUDB	GPT12
		INP2	T2_2	Timer 2
		INP3	T2EX_2	Timer 2
		INP4	CCPOS0_3	CCU6
		INP5	CTRAP_2	CCU6
		IN	XTAL (in)[1)]	XTAL

(table continues...)

Table 77 (continued) Port 2 input functions

Port pin	Input/output	Select	Connected signal(s)	From/to module
P2.5	Input / Output	GPI	P2_DATA.P5	
		INP1	T3EUDB	GPT12
		INP2	T4EUDC	GPT12
		INP3	T2_1	Timer 2
		INP4	LIN_TXD	TRX
		INP5	CCPOS1_3	CCU6
		OUT	XTAL (out)[1)]	XTAL
P2.6	Input	GPI	P2_DATA.P6	
		INP1	T4EUDD	GPT12
		INP2	T2EX_3	Timer 2
		INP3	CCPOS2_3	CCU6
		INP4	T13HR_2	CCU6
		ANALOG	AN6	ADC
P2.7	Input	GPI	P2_DATA.P7	
		INP1	CCPOS2_0	CCU6
		INP2	EXINT2_0	SCU
		INP3	T13HR_1	CCU6
		INP4	CC62_1	CCU6
		ANALOG	AN7	ADC

1) Configurable by user.

14.3.3.2 Overview - Port 2 registers

Port 2 is a general purpose input-only port. The port registers of port 2 are shown in Figure 81.

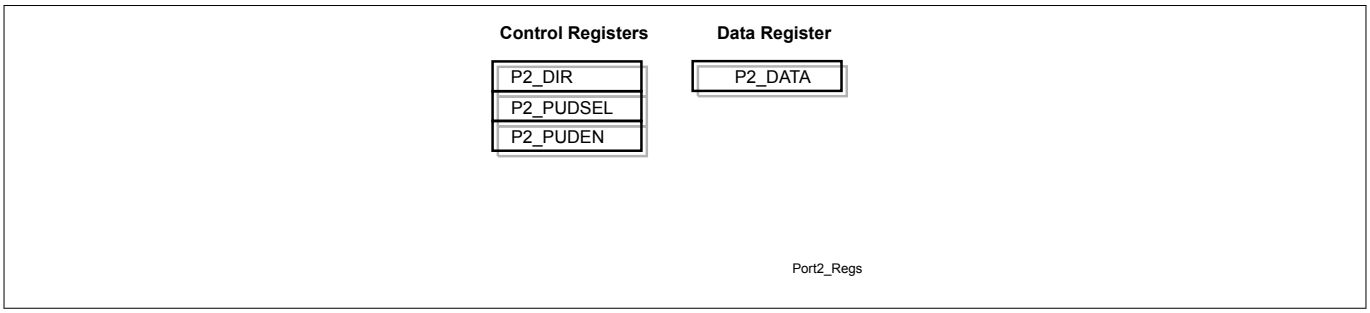


Figure 81 Port 2 registers

The registers are addressed wordwise.

14.3.3.2.1 Register overview - Port 2 registers (ascending offset address)

Table 78 Register overview - Port 2 registers (ascending offset address)

Short name	Long name	Offset address	Page number
P2_DATA	Port 2 data register	0040_H	406
P2_DIR	Port 2 direction register	0044_H	407
P2_PUDSEL	Port 2 pull-up/pull-down select register	$004C_H$	408
P2_PUDEN	Port 2 pull-up/pull-down enable register	0050_H	409

14.3.4 GPIO ports and peripheral I/O (PORT) register definition

14.3.4.1 Register address space - PORT

Table 79 Registers address space - PORT

Module	Base address	End address	Note
PORT	48028000_H	$48029FFF_H$	Ports registers

14.3.4.2 Register overview - PORT (ascending offset address)

Table 80 Register overview - PORT (ascending offset address)

Short name	Long name	Offset address	Page number
P0_DATA	Port 0 data register	0000_H	386
P0_DIR	Port 0 direction register	0004_H	388
P0_OD	Port 0 open drain control register	0008_H	390
P0_PUDSEL	Port 0 pull-up/pull-down select register	$000C_H$	391
P0_PUDEN	Port 0 pull-up/pull-down enable register	0010_H	392
P0_ALTSEL0	Port 0 alternate select 0 register	0014_H	393
P0_ALTSEL1	Port 0 alternate select 1 register	0018_H	395
P1_DATA	Port 1 data register	0020_H	397
P1_DIR	Port 1 direction register	0024_H	399
P1_OD	Port 1 open drain control register	0028_H	401
P1_PUDSEL	Port 1 pull-up/pull-down select register	$002C_H$	402
P1_PUDEN	Port 1 pull-up/pull-down enable register	0030_H	403
P1_ALTSEL0	Port 1 alternate select 0 register	0034_H	404
P1_ALTSEL1	Port 1 alternate select 1 register	0038_H	405
P2_DATA	Port 2 data register	0040_H	406
P2_DIR	Port 2 direction register	0044_H	407
P2_PUDSEL	Port 2 pull-up/pull-down select register	$004C_H$	408
P2_PUDEN	Port 2 pull-up/pull-down enable register	0050_H	409

14.3.4.3 Port 0 data register

P0_DATA Offset address: 0000_H

Port 0 data register RESET_TYPE_3 value: $0000\ 00XX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES										PP5_STS	PP4_STS	PP3_STS	PP2_STS	PP1_STS	PP0_STS
r										r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES										PP5	PP4	PP3	PP2	PP1	PP0
r										rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
PP0	0	rwh	**Port 0 pin 0 data value** 0_B **0**: Port 0 pin 0 data value = 0 1_B **1**: Port 0 pin 0 data value = 1
PP1	1	rwh	**Port 0 pin 1 data value** 0_B **0**: Port 0 pin 1 data value = 0 1_B **1**: Port 0 pin 1 data value = 1
PP2	2	rwh	**Port 0 pin 2 data value** 0_B **0**: Port 0 pin 2 data value = 0 1_B **1**: Port 0 pin 2 data value = 1
PP3	3	rwh	**Port 0 pin 3 data value** 0_B **0**: Port 0 pin 3 data value = 0 1_B **1**: Port 0 pin 3 data value = 1
PP4	4	rwh	**Port 0 pin 4 data value** 0_B **0**: Port 0 pin 4 data value = 0 1_B **1**: Port 0 pin 4 data value = 1
PP5	5	rwh	**Port 0 pin 5 data value** 0_B **0**: Port 0 pin 5 data value = 0 1_B **1**: Port 0 pin 5 data value = 1
RES	15:6, 31:22	r	**Reserved** Always read as 0.
PP0_STS	16	r	**Port 0 pin 0 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 0 pin 0 data value = 0 1_B **1**: Port 0 pin 0 data value = 1
PP1_STS	17	r	**Port 0 pin 1 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 0 pin 1 data value = 0 1_B **1**: Port 0 pin 1 data value = 1

(table continues...)

(continued)

Field	Bits	Type	Description
PP2_STS	18	r	**Port 0 pin 2 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 0 pin 2 data value = 0 1_B **1**: Port 0 pin 2 data value = 1
PP3_STS	19	r	**Port 0 pin 3 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 0 pin 3 data value = 0 1_B **1**: Port 0 pin 3 data value = 1
PP4_STS	20	r	**Port 0 pin 4 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 0 pin 4 data value = 0 1_B **1**: Port 0 pin 4 data value = 1
PP5_STS	21	r	**Port 0 pin 5 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 0 pin 5 data value = 0 1_B **1**: Port 0 pin 5 data value = 1

14.3.4.4 Port 0 direction register

P0_DIR Offset address: 0004_H

Port 0 direction register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES										PP5_INEN	PP4_INEN	PP3_INEN	PP2_INEN	PP1_INEN	PP0_INEN
r										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES										PP5	PP4	PP3	PP2	PP1	PP0
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Port 0 pin 0 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP1	1	rw	**Port 0 pin 1 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP2	2	rw	**Port 0 pin 2 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP3	3	rw	**Port 0 pin 3 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP4	4	rw	**Port 0 pin 4 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP5	5	rw	**Port 0 pin 5 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
RES	15:6, 31:22	r	**Reserved** Always read as 0.
PP0_INEN	16	rw	**Port 0 pin 0 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP1_INEN	17	rw	**Port 0 pin 1 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled

(table continues...)

(continued)

Field	Bits	Type	Description
PP2_INEN	18	rw	**Port 0 pin 2 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP3_INEN	19	rw	**Port 0 pin 3 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP4_INEN	20	rw	**Port 0 pin 4 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP5_INEN	21	rw	**Port 0 pin 5 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled

14.3.4.5 Port 0 open drain control register

P0_OD Offset address: 0008_H

Port 0 open drain control register RESET_TYPE_3 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										**PP5**	**PP4**	**PP3**	**PP2**	**PP1**	**PP0**
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Port 0 pin 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
PP1	1	rw	**Port 0 pin 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
PP2	2	rw	**Port 0 pin 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
PP3	3	rw	**Port 0 pin 3 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
PP4	4	rw	**Port 0 pin 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
PP5	5	rw	**Port 0 pin 5 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	31:6	r	**Reserved** Always read as 0.

14.3.4.6 Port 0 pull-up/pull-down select register

P0_PUDSEL

Port 0 pull-up/pull-down select register

Offset address: $000C_H$

RESET_TYPE_3 value: $0000\ 003B_H$

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

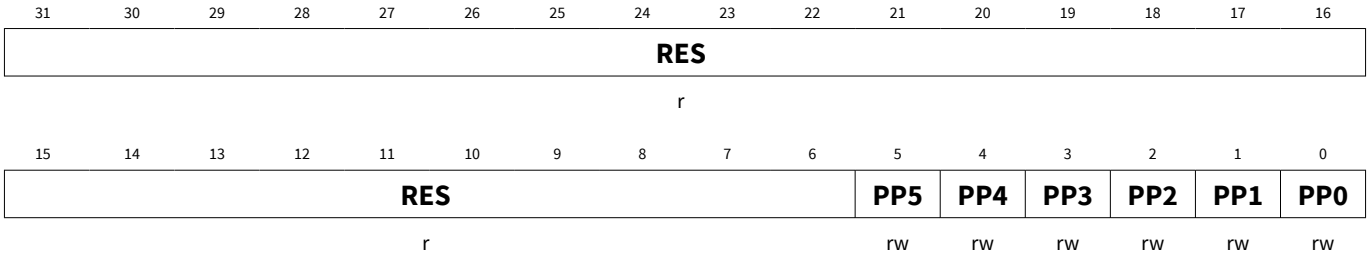
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES										**PP5**	**PP4**	**PP3**	**PP2**	**PP1**	**PP0**
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Pull-up/pull-down select port 0 bit 0** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP1	1	rw	**Pull-up/pull-down select port 0 bit 1** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP2	2	rw	**Pull-up/pull-down select port 0 bit 2** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP3	3	rw	**Pull-up/pull-down select port 0 bit 3** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP4	4	rw	**Pull-up/pull-down select port 0 bit 4** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP5	5	rw	**Pull-up/pull-down select port 0 bit 5** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
RES	31:6	r	**Reserved** Always read as 0.

14.3.4.7 Port 0 pull-up/pull-down enable register

P0_PUDEN Offset address: 0010_H

Port 0 pull-up/pull-down enable register RESET_TYPE_3 value: $0000\ 003F_H$



Field	Bits	Type	Description
PP0	0	rw	**Pull-up/pull-down enable at port 0 bit 0** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP1	1	rw	**Pull-up/pull-down enable at port 0 bit 1** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP2	2	rw	**Pull-up/pull-down enable at port 0 bit 2** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP3	3	rw	**Pull-up/pull-down enable at port 0 bit 3** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP4	4	rw	**Pull-up/pull-down enable at port 0 bit 4** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP5	5	rw	**Pull-up/pull-down enable at port 0 bit 5** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
RES	31:6	r	**Reserved** Always read as 0.

14.3.4.8 Port 0 alternate select 0 register

P0_ALTSEL0 Offset address: 0014_H

Port 0 alternate select 0 register RESET_TYPE_3 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										PP5	PP4	PP3	PP2	PP1	PP0
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL1.PPx = 0; Alternate select 2 if P0_ALTSEL1.PPx = 1 1_B Alternate select 1 if P0_ALTSEL1.PPx = 0; Alternate select 3 if P0_ALTSEL1.PPx = 1
PP1	1	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL1.PPx = 0; Alternate select 2 if P0_ALTSEL1.PPx = 1 1_B Alternate select 1 if P0_ALTSEL1.PPx = 0; Alternate select 3 if P0_ALTSEL1.PPx = 1
PP2	2	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL1.PPx = 0; Alternate select 2 if P0_ALTSEL1.PPx = 1 1_B Alternate select 1 if P0_ALTSEL1.PPx = 0; Alternate select 3 if P0_ALTSEL1.PPx = 1
PP3	3	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL1.PPx = 0; Alternate select 2 if P0_ALTSEL1.PPx = 1 1_B Alternate select 1 if P0_ALTSEL1.PPx = 0; Alternate select 3 if P0_ALTSEL1.PPx = 1
PP4	4	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL1.PPx = 0; Alternate select 2 if P0_ALTSEL1.PPx = 1 1_B Alternate select 1 if P0_ALTSEL1.PPx = 0; Alternate select 3 if P0_ALTSEL1.PPx = 1
PP5	5	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B Normal GPIO if P0_ALTSEL1.PPx = 0; Alternate select 2 if P0_ALTSEL1.PPx = 1 1_B Alternate select 1 if P0_ALTSEL1.PPx = 0; Alternate select 3 if P0_ALTSEL1.PPx = 1
RES	31:6	r	**Reserved** Always read as 0.

14.3.4.9 Port 0 alternate select 1 register

P0_ALTSEL1 Offset address: 0018_H

Port 0 alternate select 1 register RESET_TYPE_3 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										PP5	PP4	PP3	PP2	PP1	PP0
r										rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL0.PPx = 0; Alternate select 2 if P0_ALTSEL0.PPx = 1 1_B Alternate select 1 if P0_ALTSEL0.PPx = 0; Alternate select 3 if P0_ALTSEL0.PPx = 1
PP1	1	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL0.PPx = 0; Alternate select 2 if P0_ALTSEL0.PPx = 1 1_B Alternate select 1 if P0_ALTSEL0.PPx = 0; Alternate select 3 if P0_ALTSEL0.PPx = 1
PP2	2	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL0.PPx = 0; Alternate select 2 if P0_ALTSEL0.PPx = 1 1_B Alternate select 1 if P0_ALTSEL0.PPx = 0; Alternate select 3 if P0_ALTSEL0.PPx = 1
PP3	3	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL0.PPx = 0; Alternate select 2 if P0_ALTSEL0.PPx = 1 1_B Alternate select 1 if P0_ALTSEL0.PPx = 0; Alternate select 3 if P0_ALTSEL0.PPx = 1
PP4	4	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)** 0_B Normal GPIO if P0_ALTSEL0.PPx = 0; Alternate select 2 if P0_ALTSEL0.PPx = 1 1_B Alternate select 1 if P0_ALTSEL0.PPx = 0; Alternate select 3 if P0_ALTSEL0.PPx = 1
PP5	5	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P0_ALTSEL0.PPx and P0_ALTSEL1.PPx)**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B Normal GPIO if P0_ALTSEL0.PPx = 0; Alternate select 2 if P0_ALTSEL0.PPx = 1 1_B Alternate select 1 if P0_ALTSEL0.PPx = 0; Alternate select 3 if P0_ALTSEL0.PPx = 1
RES	31:6	r	**Reserved** Always read as 0.

14.3.4.10 Port 1 data register

P1_DATA Offset address: 0020_H

Port 1 data register RESET_TYPE_3 value: $0000\ 00XX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											PP4_STS	RES	PP2_STS	PP1_STS	PP0_STS
r											rwh	r	rwh	rwh	rwh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											PP4	RES	PP2	PP1	PP0
r											rwh	r	rwh	rwh	rwh

Field	Bits	Type	Description
PP0	0	rwh	**Port 1 pin 0 data value** 0_B **0**: Port 1 pin 0 data value = 0 1_B **1**: Port 1 pin 0 data value = 1
PP1	1	rwh	**Port 1 pin 1 data value** 0_B **0**: Port 1 pin 1 data value = 0 1_B **1**: Port 1 pin 1 data value = 1
PP2	2	rwh	**Port 1 pin 2 data value** 0_B **0**: Port 1 pin 2 data value = 0 1_B **1**: Port 1 pin 2 data value = 1
RES	3, 15:5, 19, 31:21	r	**Reserved** Always read as 0.
PP4	4	rwh	**Port 1 pin 4 data value** 0_B **0**: Port 1 pin 4 data value = 0 1_B **1**: Port 1 pin 4 data value = 1
PP0_STS	16	rwh	**Port 1 pin 0 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 1 pin 0 data value = 0 1_B **1**: Port 1 pin 0 data value = 1
PP1_STS	17	rwh	**Port 1 pin 1 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 1 pin 1 data value = 0 1_B **1**: Port 1 pin 1 data value = 1
PP2_STS	18	rwh	**Port 1 pin 2 data value (read back of port data when IO is configured as output)** 0_B **0**: Port 1 pin 2 data value = 0 1_B **1**: Port 1 pin 2 data value = 1
PP4_STS	20	rwh	

(table continues...)

(continued)

Field	Bits	Type	Description
			Port 1 pin 4 data value (read back of port data when IO is configured as output) 0_B **0**: Port 1 pin 4 data value = 0 1_B **1**: Port 1 pin 4 data value = 1

14.3.4.11 Port 1 direction register

P1_DIR Offset address: 0024_H

Port 1 direction register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES											PP3_INEN	RES	PP2_INEN	PP1_INEN	PP0_INEN
r											rw	r	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											PP4	RES	PP2	PP1	PP0
r											rw	r	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Port 1 pin 0 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP1	1	rw	**Port 1 pin 1 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP2	2	rw	**Port 1 pin 2 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
RES	3, 15:5, 19, 31:21	r	**Reserved** Always read as 0.
PP4	4	rw	**Port 1 pin 4 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP0_INEN	16	rw	**Port 1 pin 0 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP1_INEN	17	rw	**Port 1 pin 1 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP2_INEN	18	rw	**Port 1 pin 2 input Schmitt trigger enable (only valid if IO is configured as output)** 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled
PP3_INEN	20	rw	

(table continues...)

(continued)

Field	Bits	Type	Description
			Port 1 pin 4 input Schmitt trigger enable (only valid if IO is configured as output) 0_B **0**: Schmitt trigger is disabled (default) 1_B **1**: Schmitt trigger is enabled

14.3.4.12 Port 1 open drain control register

P1_OD Offset address: 0028_H

Port 1 open drain control register RESET_TYPE_3 value: $0000\ 0000_H$

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

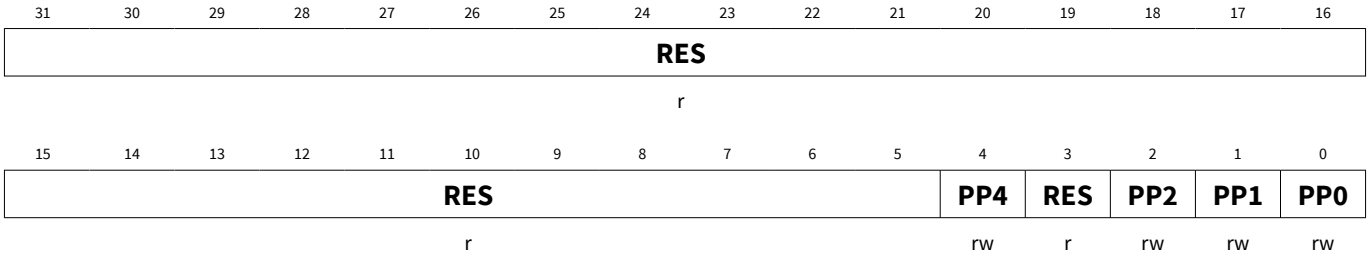
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES											PP4	RES	PP2	PP1	PP0
r											rw	r	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Port 1 pin 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
PP1	1	rw	**Port 1 pin 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
PP2	2	rw	**Port 1 pin 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
RES	3, 31:5	r	**Reserved** Always read as 0.
PP4	4	rw	**Port 1 pin 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

14.3.4.13 Port 1 pull-up/pull-down select register

P1_PUDSEL Offset address: $002C_H$

Port 1 pull-up/pull-down select register RESET_TYPE_3 value: $0000\ 0017_H$

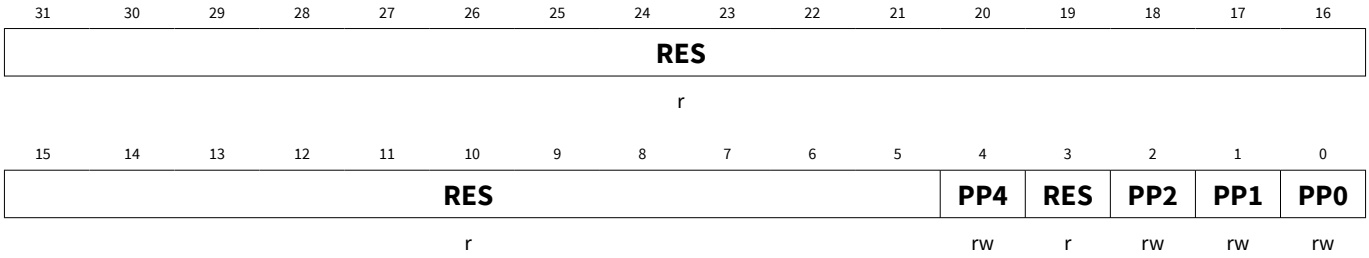


Field	Bits	Type	Description
PP0	0	rw	**Pull-up/pull-down select port 1 bit 0** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP1	1	rw	**Pull-up/pull-down select port 1 bit 1** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP2	2	rw	**Pull-up/pull-down select port 1 bit 2** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
RES	3, 31:5	r	**Reserved** Always read as 0.
PP4	4	rw	**Pull-up/pull-down select port 1 bit 4** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)

14.3.4.14 Port 1 pull-up/pull-down enable register

P1_PUDEN Offset address: 0030_H

Port 1 pull-up/pull-down enable register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
PP0	0	rw	**Pull-up/pull-down enable at port 1 bit 0** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP1	1	rw	**Pull-up/pull-down enable at port 1 bit 1** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP2	2	rw	**Pull-up/pull-down enable at port 1 bit 2** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
RES	3, 31:5	r	**Reserved** Always read as 0.
PP4	4	rw	**Pull-up/pull-down enable at port 1 bit 4** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)

14.3.4.15 Port 1 alternate select 0 register

P1_ALTSEL0 Offset address: 0034_H

Port 1 alternate select 0 register RESET_TYPE_3 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											PP4	RES	PP2	PP1	PP0
r											rw	r	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P1_ALTSEL0.PPx and P1_ALTSEL1.PPx)** 0_B Normal GPIO if P1_ALTSEL1.PPx = 0; Alternate select 2 if P1_ALTSEL1.PPx = 1 1_B Alternate select 1 if P1_ALTSEL1.PPx = 0; Alternate select 3 if P1_ALTSEL1.PPx = 1
PP1	1	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P1_ALTSEL0.PPx and P1_ALTSEL1.PPx)** 0_B Normal GPIO if P1_ALTSEL1.PPx = 0; Alternate select 2 if P1_ALTSEL1.PPx = 1 1_B Alternate select 1 if P1_ALTSEL1.PPx = 0; Alternate select 3 if P1_ALTSEL1.PPx = 1
PP2	2	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P1_ALTSEL0.PPx and P1_ALTSEL1.PPx)** 0_B Normal GPIO if P1_ALTSEL1.PPx = 0; Alternate select 2 if P1_ALTSEL1.PPx = 1 1_B Alternate select 1 if P1_ALTSEL1.PPx = 0; Alternate select 3 if P1_ALTSEL1.PPx = 1
RES	3, 31:5	r	**Reserved** Always read as 0.
PP4	4	rw	**Normal GPIO or alternate select 1, 2 or 3 (depends on bits P1_ALTSEL0.PPx and P1_ALTSEL1.PPx)** 0_B Normal GPIO if P1_ALTSEL1.PPx = 0; Alternate select 2 if P1_ALTSEL1.PPx = 1 1_B Alternate select 1 if P1_ALTSEL1.PPx = 0; Alternate select 3 if P1_ALTSEL1.PPx = 1

14.3.4.16 Port 1 alternate select 1 register

P1_ALTSEL1 Offset address: 0038_H

Port 1 alternate select 1 register RESET_TYPE_3 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											PP4	RES	PP2	PP1	PP0
r											rw	r	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	0_B Normal GPIO if P1_ALTSEL0.PPx = 0; Alternate select 2 if P1_ALTSEL0.PPx = 1 1_B Alternate select 1 if P1_ALTSEL0.PPx = 0; Alternate select 3 if P1_ALTSEL0.PPx = 1
PP1	1	rw	0_B Normal GPIO if P1_ALTSEL0.PPx = 0; Alternate select 2 if P1_ALTSEL0.PPx = 1 1_B Alternate select 1 if P1_ALTSEL0.PPx = 0; Alternate select 3 if P1_ALTSEL0.PPx = 1
PP2	2	rw	0_B Normal GPIO if P1_ALTSEL0.PPx = 0; Alternate select 2 if P1_ALTSEL0.PPx = 1 1_B Alternate select 1 if P1_ALTSEL0.PPx = 0; Alternate select 3 if P1_ALTSEL0.PPx = 1
RES	3, 31:5	r	**Reserved** Always read as 0.
PP4	4	rw	0_B Normal GPIO if P1_ALTSEL0.PPx = 0; Alternate select 2 if P1_ALTSEL0.PPx = 1 1_B Alternate select 1 if P1_ALTSEL0.PPx = 0; Alternate select 3 if P1_ALTSEL0.PPx = 1

14.3.4.17 Port 2 data register

P2_DATA Offset address: 0040_H

Port 2 data register RESET_TYPE_3 value: $0000\ 00XX_H$

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

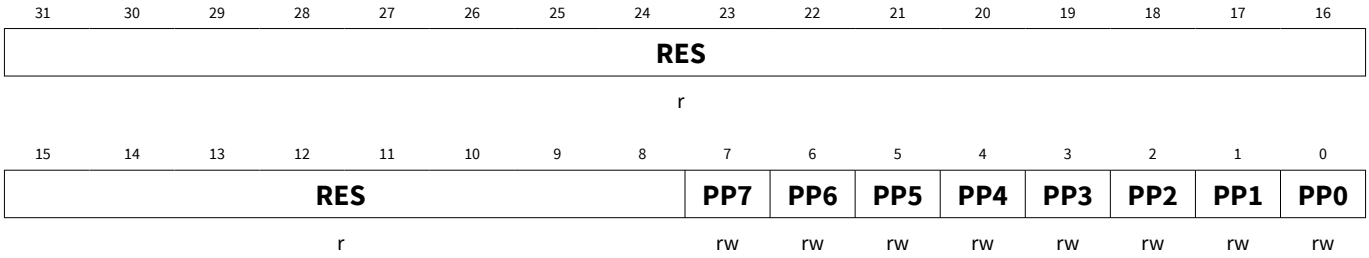
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
r								rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh

Field	Bits	Type	Description
PP0	0	rwh	**Port 2 pin 0 data value** 0_B **0**: Port 2 pin 0 data value = 0 1_B **1**: Port 2 pin 0 data value = 1
PP1	1	rwh	**Port 2 pin 1 data value** 0_B **0**: Port 2 pin 1 data value = 0 1_B **1**: Port 2 pin 1 data value = 1
PP2	2	rwh	**Port 2 pin 2 data value** 0_B **0**: Port 2 pin 2 data value = 0 1_B **1**: Port 2 pin 2 data value = 1
PP3	3	rwh	**Port 2 pin 3 data value** 0_B **0**: Port 2 pin 3 data value = 0 1_B **1**: Port 2 pin 3 data value = 1
PP4	4	rwh	**Port 2 pin 4 data value** 0_B **0**: Port 2 pin 4 data value = 0 1_B **1**: Port 2 pin 4 data value = 1
PP5	5	rwh	**Port 2 pin 5 data value** 0_B **0**: Port 2 pin 5 data value = 0 1_B **1**: Port 2 pin 5 data value = 1
PP6	6	rwh	**Port 2 pin 6 data value** 0_B **0**: Port 2 pin 6 data value = 0 1_B **1**: Port 2 pin 6 data value = 1
PP7	7	rwh	**Port 2 pin 7 data value** 0_B **0**: Port 2 pin 7 data value = 0 1_B **1**: Port 2 pin 7 data value = 1
RES	31:8	r	**Reserved** Always read as 0.

14.3.4.18 Port 2 direction register

P2_DIR Offset address: 0044_H

Port 2 direction register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
PP0	0	rw	**Port 2 pin 0 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP1	1	rw	**Port 2 pin 1 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP2	2	rw	**Port 2 pin 2 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP3	3	rw	**Port 2 pin 3 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP4	4	rw	**Port 2 pin 4 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP5	5	rw	**Port 2 pin 5 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP6	6	rw	**Port 2 pin 6 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
PP7	7	rw	**Port 2 pin 7 direction control** 0_B **0**: Direction is set to input (default) 1_B **1**: Direction is set to output
RES	31:8	r	**Reserved** Always read as 0.

14.3.4.19 Port 2 pull-up/pull-down select register

P2_PUDSEL Offset address: $004C_H$

Port 2 pull-up/pull-down select register RESET_TYPE_3 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								**PP7**	**PP6**	**PP5**	**PP4**	**PP3**	**PP2**	**PP1**	**PP0**
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Pull-up/pull-down select port 2 bit 0** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP1	1	rw	**Pull-up/pull-down select port 2 bit 1** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP2	2	rw	**Pull-up/pull-down select port 2 bit 2** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP3	3	rw	**Pull-up/pull-down select port 2 bit 3** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP4	4	rw	**Pull-up/pull-down select port 2 bit 4** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP5	5	rw	**Pull-up/pull-down select port 2 bit 5** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP6	6	rw	**Pull-up/pull-down select port 2 bit 6** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
PP7	7	rw	**Pull-up/pull-down select port 2 bit 7** 0_B **Pull_down**: Pull-down device is selected 1_B **Pull_up**: Pull-up device is selected (default)
RES	31:8	r	**Reserved** Always read as 0.

14.3.4.20 Port 2 pull-up/pull-down enable register

P2_PUDEN

Port 2 pull-up/pull-down enable register

Offset address: 0050_H

RESET_TYPE_3 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								PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
PP0	0	rw	**Pull-up/pull-down enable at port 2 bit 0** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP1	1	rw	**Pull-up/pull-down enable at port 2 bit 1** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP2	2	rw	**Pull-up/pull-down enable at port 2 bit 2** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP3	3	rw	**Pull-up/pull-down enable at port 2 bit 3** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP4	4	rw	**Pull-up/pull-down enable at port 2 bit 4** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP5	5	rw	**Pull-up/pull-down enable at port 2 bit 5** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP6	6	rw	**Pull-up/pull-down enable at port 2 bit 6** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
PP7	7	rw	**Pull-up/pull-down enable at port 2 bit 7** 0_B **Disabled**: Pull-up or pull-down device is disabled 1_B **Enabled**: Pull-up or pull-down device is enabled (default)
RES	31:8	r	**Reserved** Always read as 0.

15 General purpose timer units (GPT12)

15.1 Features

15.1.1 Features block GPT1

The following list summarizes the supported features:

- 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 block GPT2

The following list summarizes the supported features:

- 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 by register PISEL.

The GPT module is clocked with clock f_{GPT}. f_{GPT} is a clock derived from f_{SYS}.

15.2.1 Block diagram GPT1

Block GPT1 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 GPT1 timer registers.

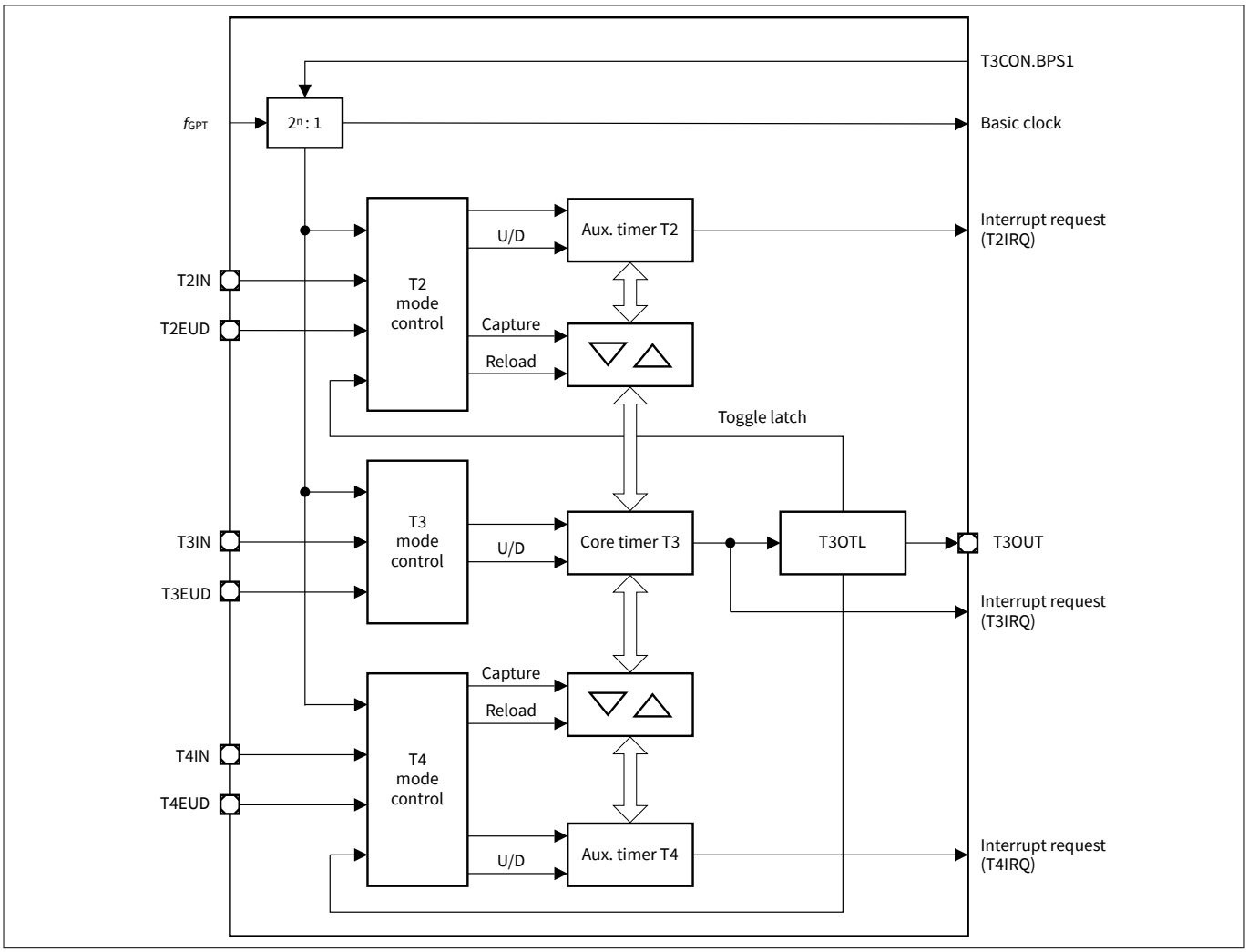


Figure 82 GPT1 block diagram (n = 2 … 5)

15.2.2 Block diagram GPT2

Block GPT2 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 GPT2 timer registers.

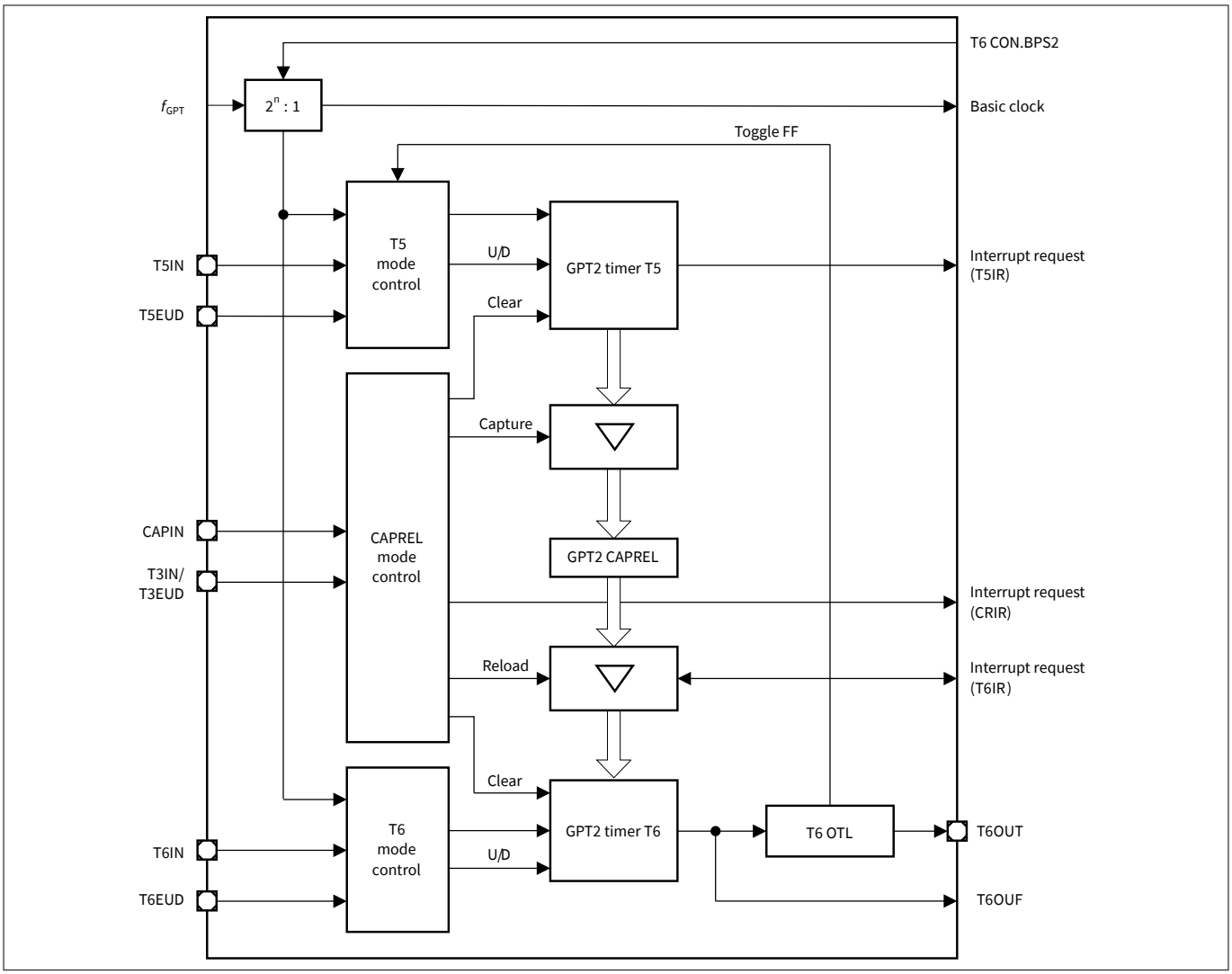


Figure 83 GPT2 block diagram (n = 1 … 4)

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.

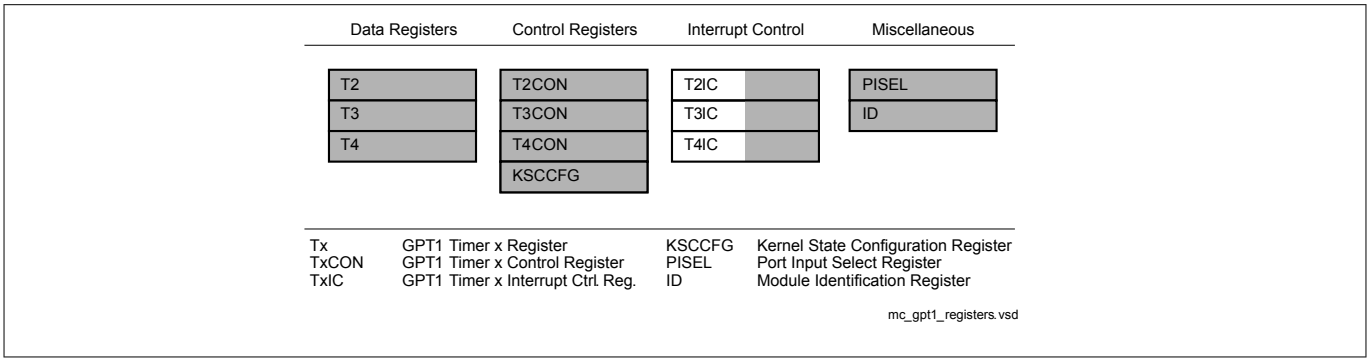


Figure 84 SFRs associated with timer block GPT1

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 bit-addressable SFR space (see GPT1 timer registers). 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 GPTM1IEN and GPTM1IRC. 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 Chapter 15.3.5, Chapter 15.6.1 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 86. 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 85 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 85, 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.

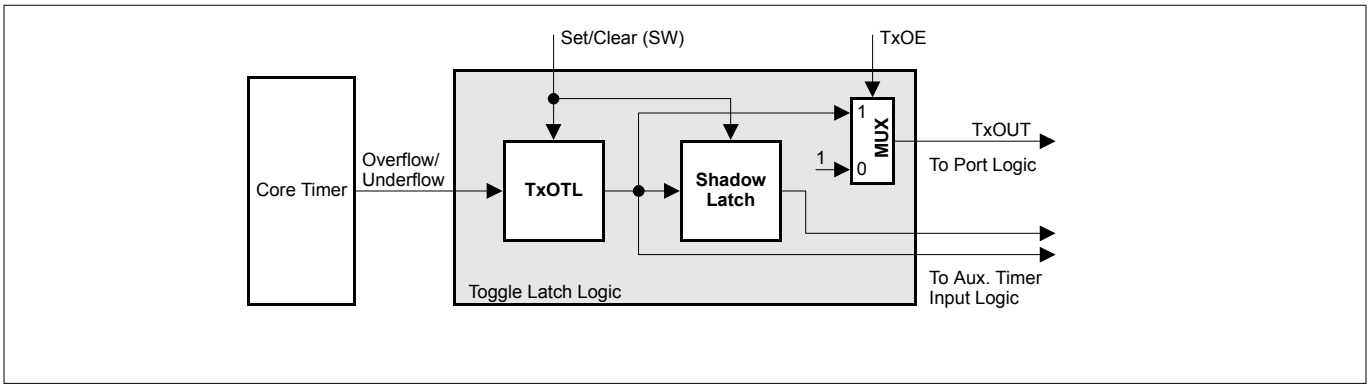


Figure 85 Block diagram of the toggle latch logic of core timer T3 (x = 3)

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 bit field 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 bit fields BPS1 and T3I in register T3CON. Please see Chapter 15.3.5 for details on the input clock options.

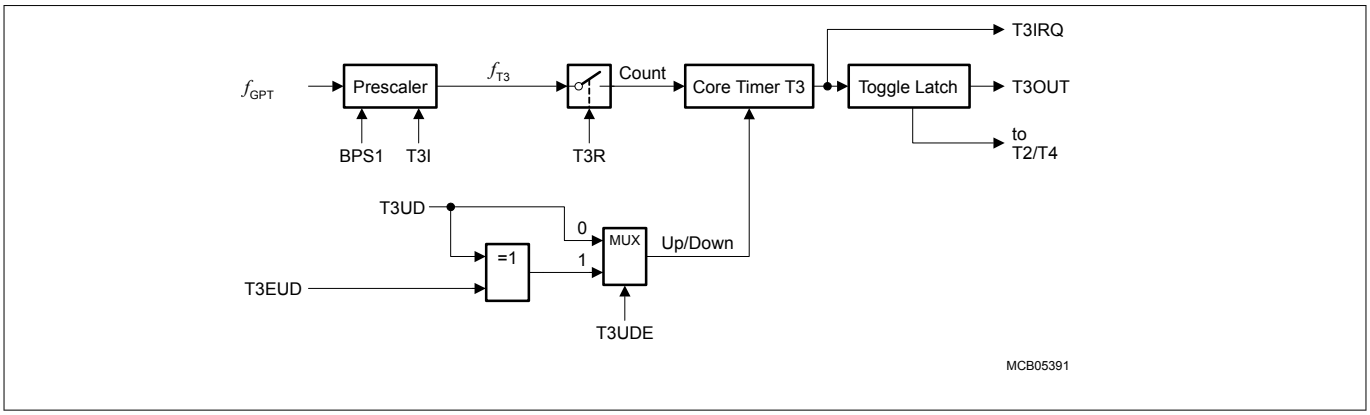


Figure 86 Block diagram of core timer T3 in mimer Mode

Timer T3 in gated timer mode

Gated timer mode for the core timer T3 is selected by setting bit field 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 Chapter 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.

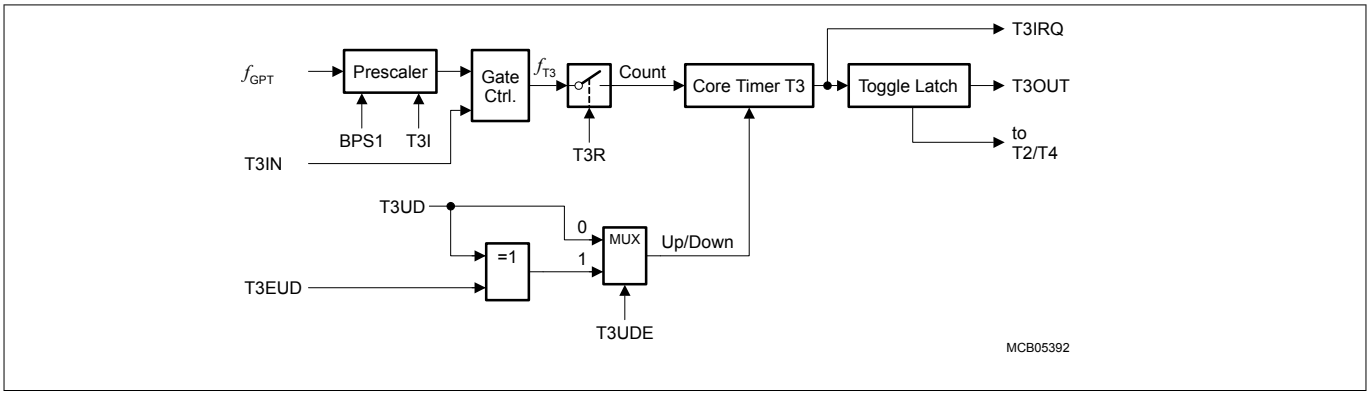


Figure 87 Block diagram of core timer T3 in gated timer mode

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 bit field 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. Bit field T3I in control register T3CON selects the triggering transition (see Table 88).

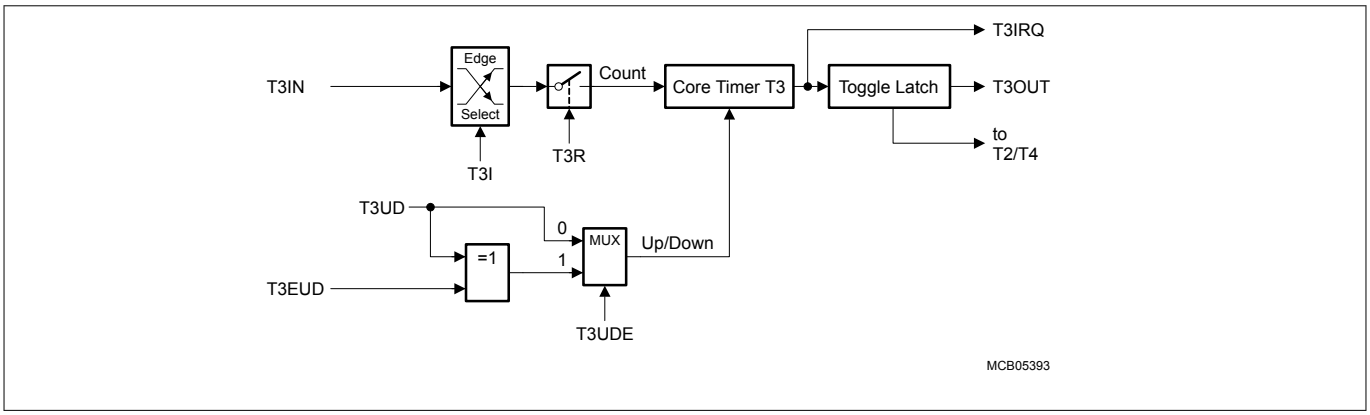


Figure 88 Block diagram of core timer T3 in counter mode

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 Chapter 15.3.5.

Timer T3 in incremental interface mode

Incremental interface mode for the core timer T3 is selected by setting bit field 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.

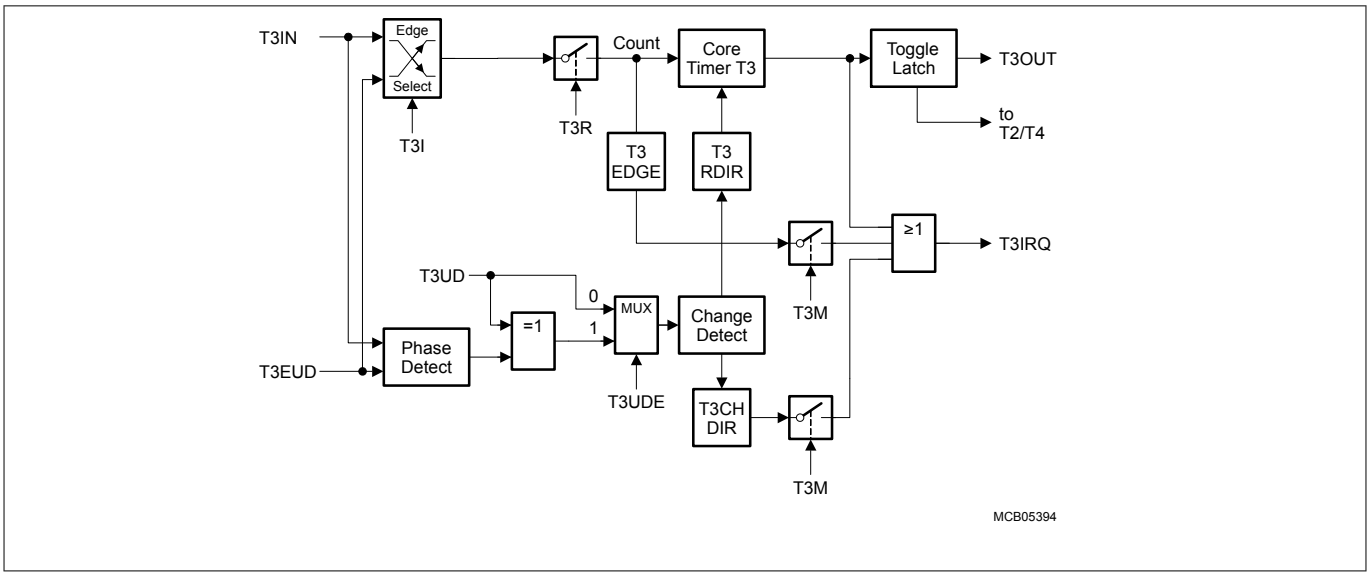


Figure 89 Block diagram of core Timer T3 in incremental interface mode

Bit field T3I in control register T3CON selects the triggering transitions (see Counter mode: Encoding of GPT1 input edge selection). 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 MOTIX™ TLE984xQX 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.

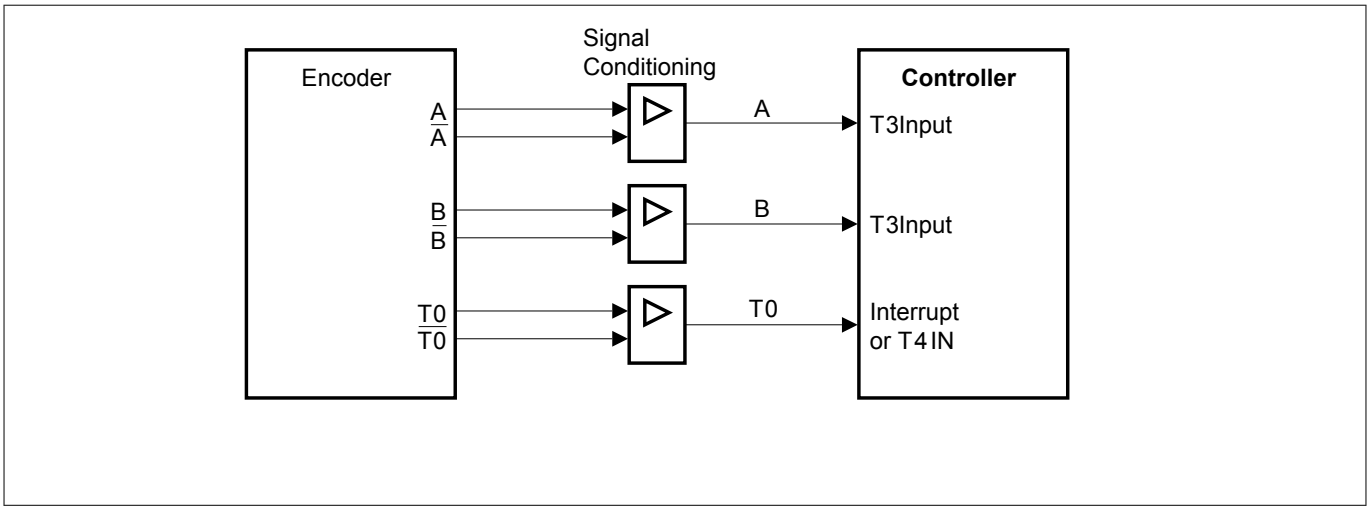


Figure 90 Connection of the encoder to the MOTIX™ TLE984xQX

For incremental interface operation, the following conditions must be met:

- Bit field 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 Chapter 15.3.5.

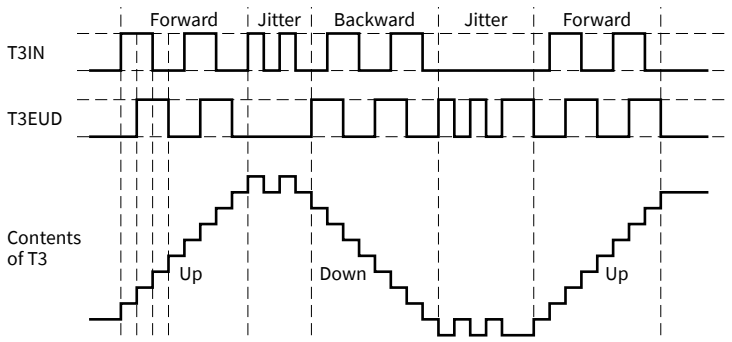
As in incremental interface mode two input signals with a 90° phase shift are evaluated, their maximum input frequency can be half the maximum count frequency.

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 81 summarizes the possible combinations.

Table 81 GPT1 core timer T3 (incremental interface mode) count direction

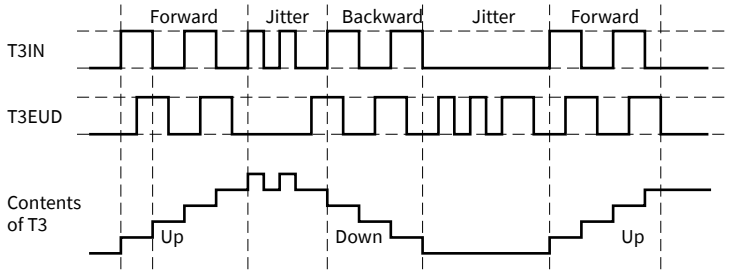
Level on respective other input	T3IN input		T3EUD input	
	Rising ↑	Falling ↓	Rising ↑	Falling ↓
High	Down	Up	Up	Down
Low	Up	Down	Down	Up

Figure 91 and Figure 92 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 91 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 92 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).

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 Timer mode and gated timer mode: Encoding of GPT1 overall prescaler factor.

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 bit field TxM in register TxCON to 000_B.

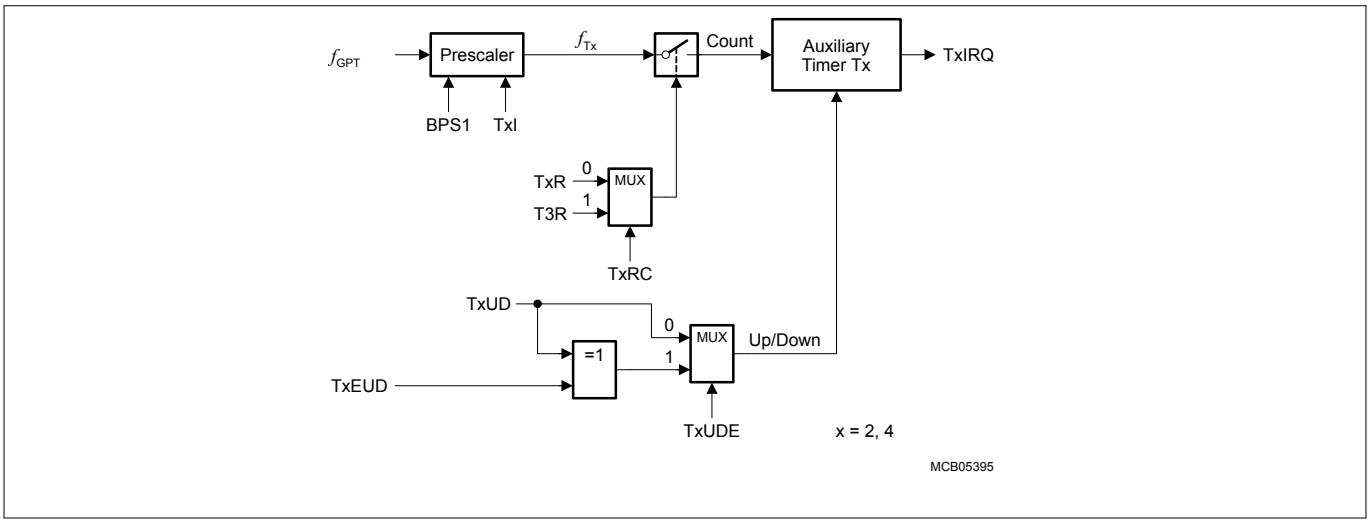


Figure 93 Block diagram of an auxiliary timer in timer mode

Timers T2 and T4 in gated timer mode

Gated timer mode for an auxiliary timer Tx is selected by setting bit field 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.

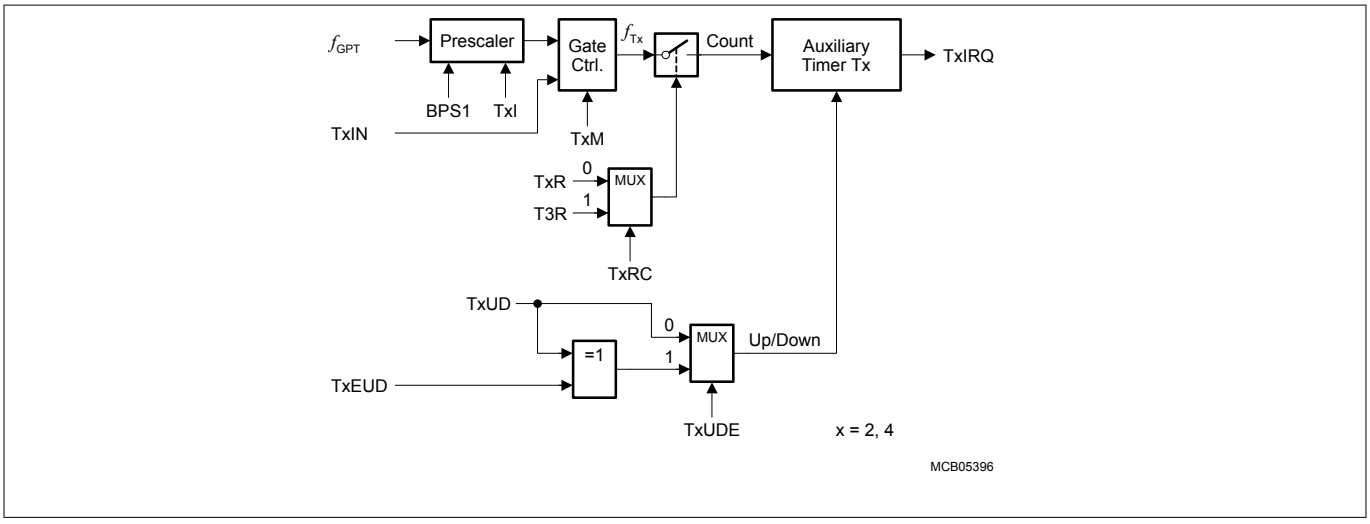


Figure 94 Block diagram of an auxiliary timer in gated timer mode

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 bit field 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. Bit field TxI in control register TxCON selects the triggering transition (see Table 90).

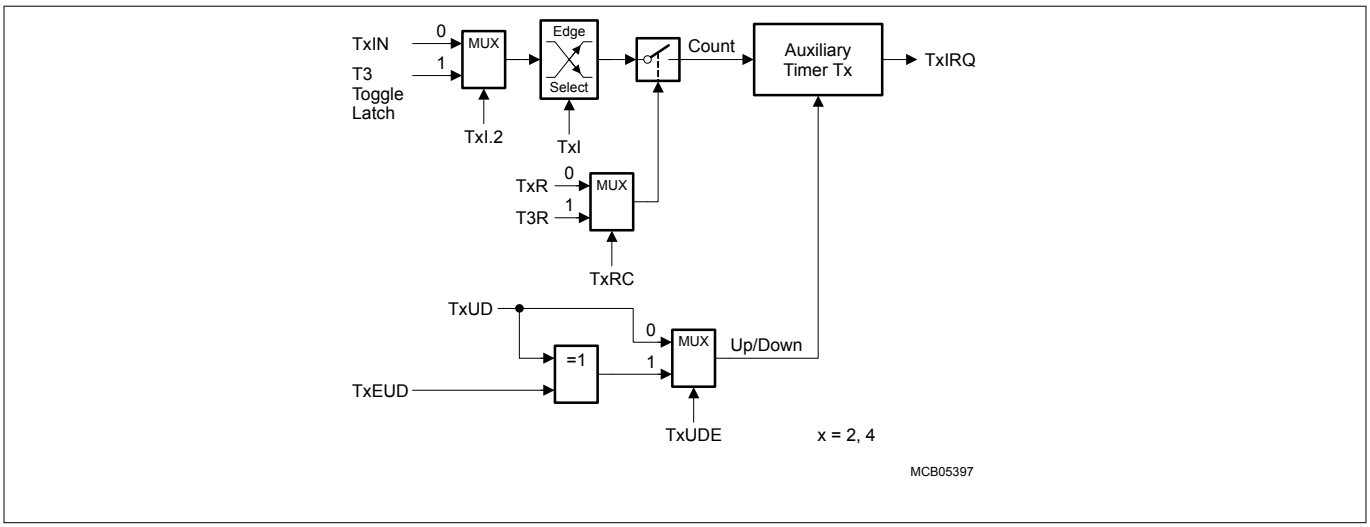


Figure 95 Block diagram of an auxiliary timer in counter mode

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 Chapter 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.

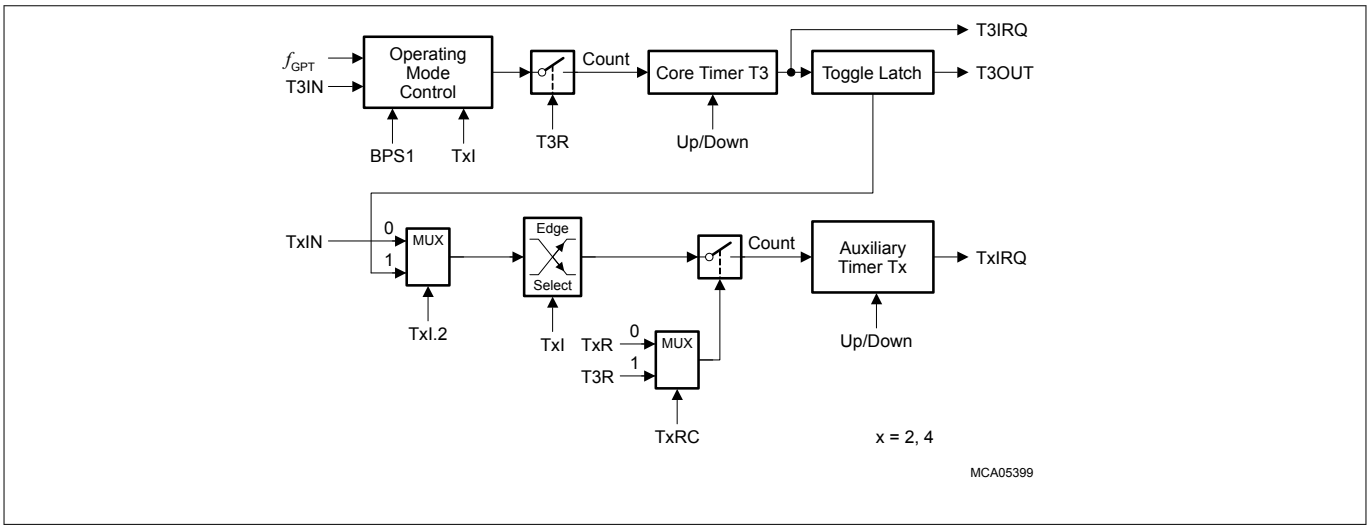


Figure 96 Concatenation of core timer T3 and an auxiliary timer

For measuring longer time periods, the core timer T3 may be concatenated with an auxiliary timer (T2/T4). The core timer contains the low part, and the auxiliary timer contains the high part of the extended timer value.

Timers T2 and T4 in capture mode

Capture mode for an auxiliary timer Tx is selected by setting bit field 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 90). 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.

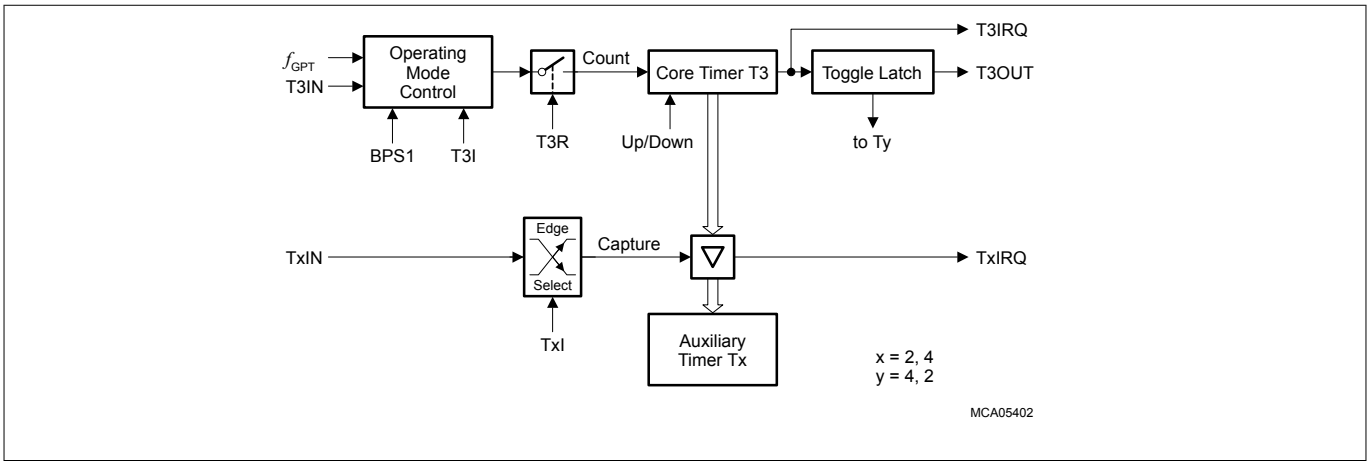


Figure 97 GPT1 auxiliary timer in capture mode

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 Chapter 15.3.5.

Timers T2 and T4 in incremental interface mode

Incremental interface mode for an auxiliary timer Tx is selected by setting bit field 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.

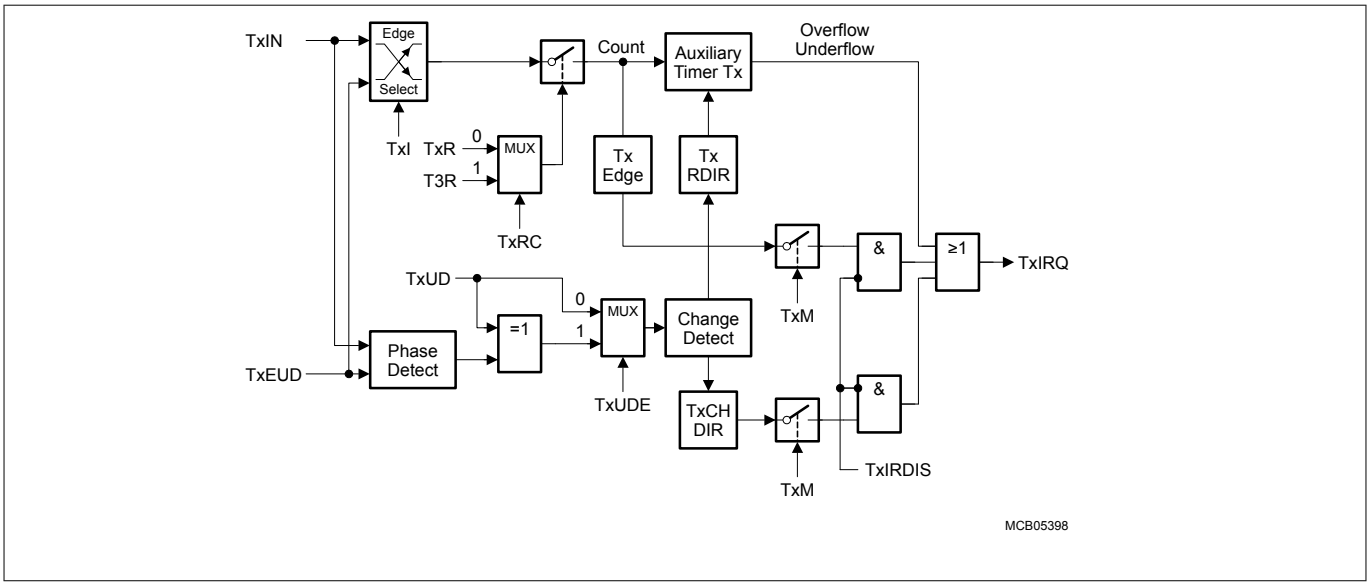


Figure 98 Block diagram of an auxiliary timer in incremental interface mode

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).

Timers T2 and T4 in reload mode

Reload mode for an auxiliary timer Tx is selected by setting bit field 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 90), that is 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.

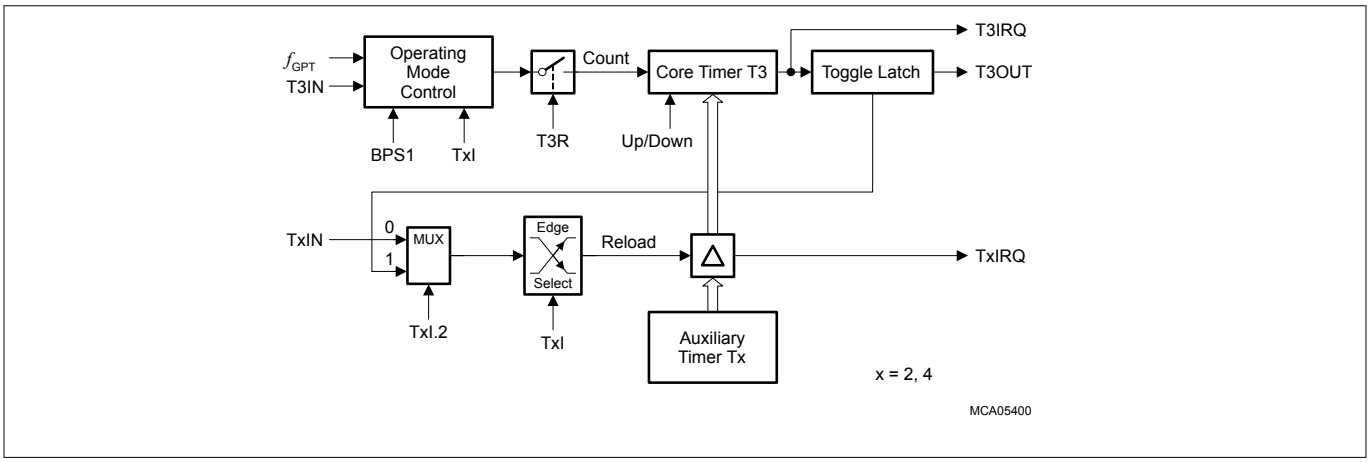


Figure 99 GPT1 auxiliary timer in reload mode

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 Chapter 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 100 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.

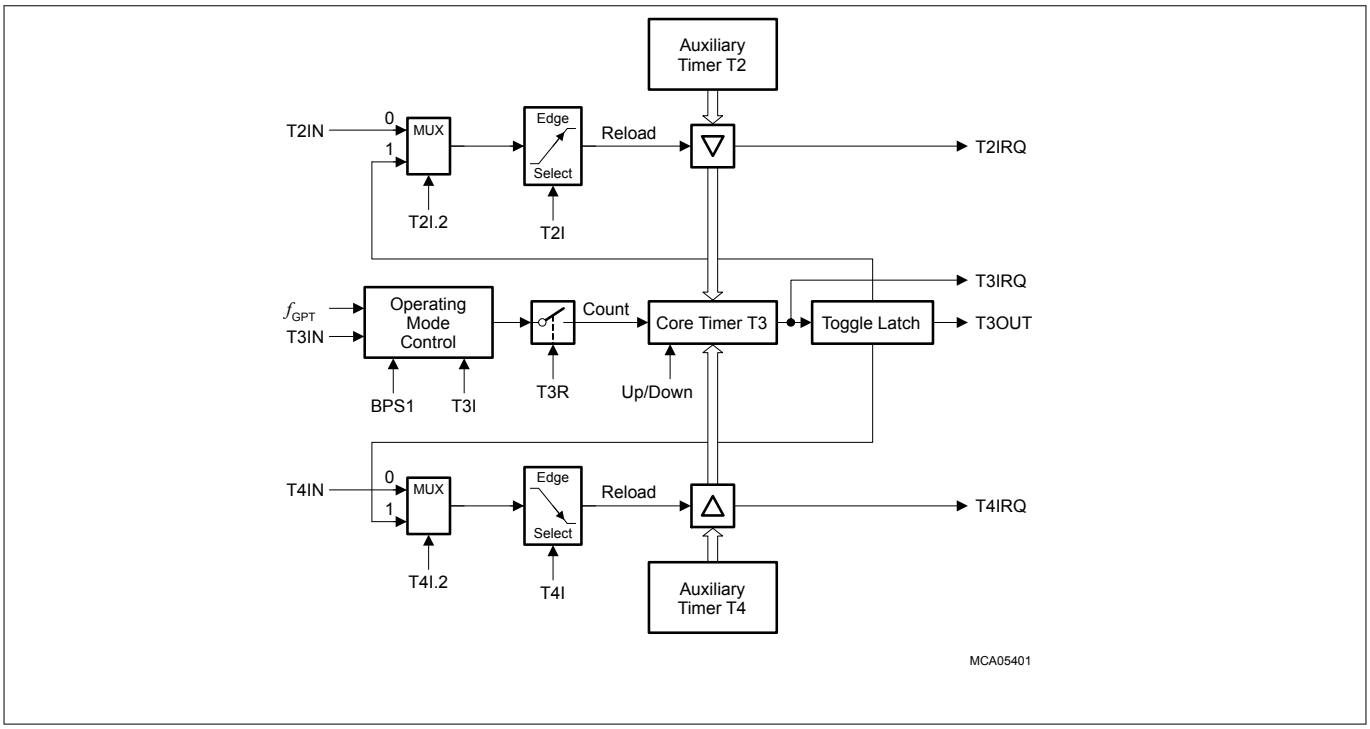


Figure 100 GPT1 timer reload configuration for PWM generation

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 bit field BPS1 in register T3CON (see Figure 82). 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 82 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 bit field 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 bit field 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:

$$f_{Tx} = \frac{f_{GPT}}{F(BPS1) \times 2^{<Txl>}} \quad r_{Tx}[\mu s] = \frac{F(BPS1) \times 2^{<Txl>}}{f_{GPT}[MHz]} \tag{6}$$

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 87 summarizes the resulting overall divider factors for a GPT1 timer that result from these cascaded prescalers.

Table 83 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 83 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 82). 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 84 summarizes the resulting requirements for external GPT1 input signals.

Table 84 GPT1 external input signal limits

GPT1 basic clock = 10 MHz		Input frequency 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 Register overview - GPT1 registers (ascending offset address)

Table 85 Register overview - GPT1 registers (ascending offset address)

Short name	Long name	Offset address	Page number
GPT12E_T2CON	Timer T2 control register	0008_H	455
GPT12E_T3CON	Timer T3 control register	$000C_H$	457
GPT12E_T4CON	Timer T4 control register	0010_H	459
GPT12E_T2	Timer T2 count register	0020_H	461
GPT12E_T3	Timer T3 count register	0024_H	462
GPT12E_T4	Timer T4 count register	0028_H	463

15.3.7.2 GPT1 timer interrupt control registers

The interrupt control and status register are located in the System control unit - digital modules (SCU-DM) module.

15.3.7.3 GPT1 encoding

15.3.7.3.1 Encoding of GPT1 timer count direction control

Table 86 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

15.3.7.3.2 Timer mode and gated timer mode: Encoding of GPT1 overall prescaler factor

Table 87 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

(table continues...)

Table 87 (continued) 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 = 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 bit field BPS1.

15.3.7.3.3 Counter mode: Encoding of GPT1 input edge selection

Table 88 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 89 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 90 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

(table continues...)

Table 90 (continued) GPT1 auxiliary timers T2/T4 input edge selection (counter mode, reload mode)

T2I/T4I	Triggering edge for counter increment/decrement
111_B	Any transition (rising or falling edge) of T3 toggle latch T3OTL

15.3.7.3.4 Incremental interface mode: Encoding of input edge selection

Table 91 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

15.4 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.

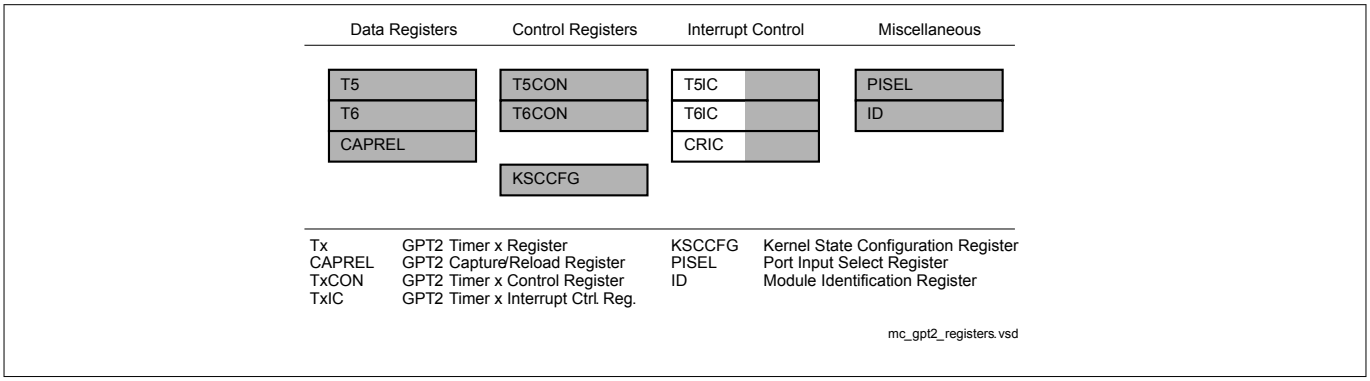


Figure 101 SFRs associated with timer block GPT2

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 GPT2 timer registers). 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 GPTM1IEN and GPTM1IRC. 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 Chapter 15.4.6, Chapter 15.6.1 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 97. 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 102 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 102, 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.

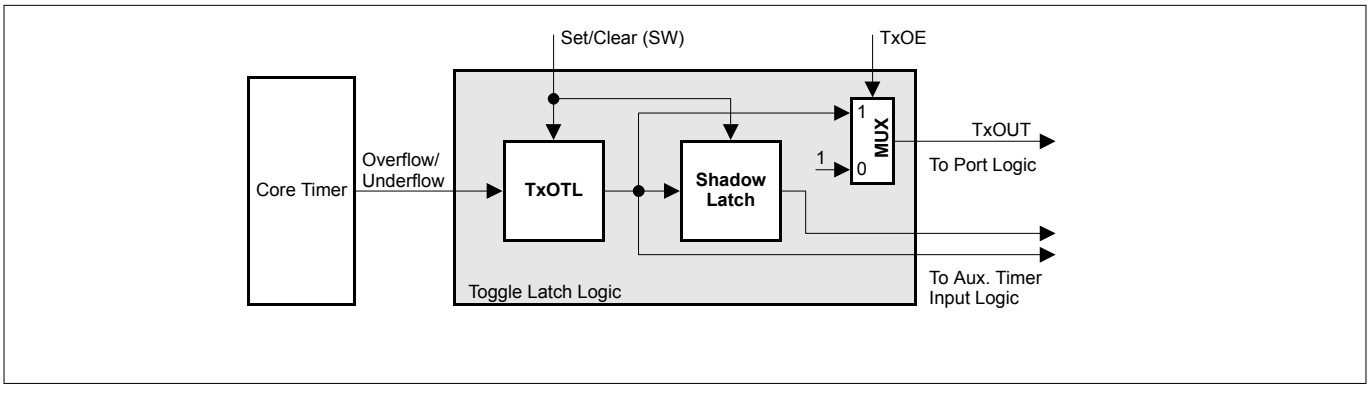


Figure 102 Block diagram of the toggle latch logic of core timer T6 (x = 6)

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 bit field 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 Chapter 15.4.6 for details on the input clock options.

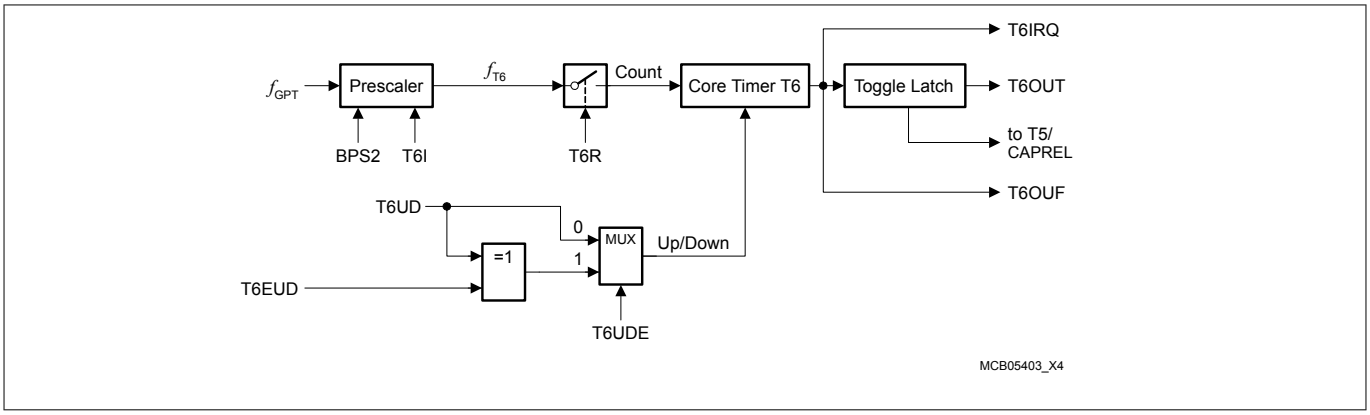


Figure 103 Block diagram of core timer T6 in timer mode

Timer T6 in gated timer mode

Gated timer mode for the core timer T6 is selected by setting bit field 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 Chapter 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.

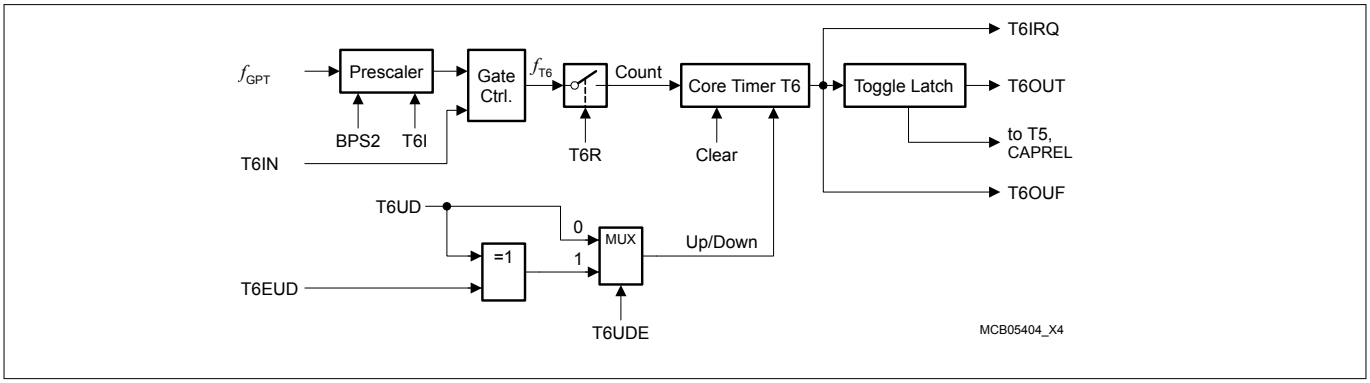


Figure 104 Block diagram of core timer T6 in gated timer mode

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 bit field 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. Bit field T6I in control register T6CON selects the triggering transition (see Chapter 15.4.8.3.2.

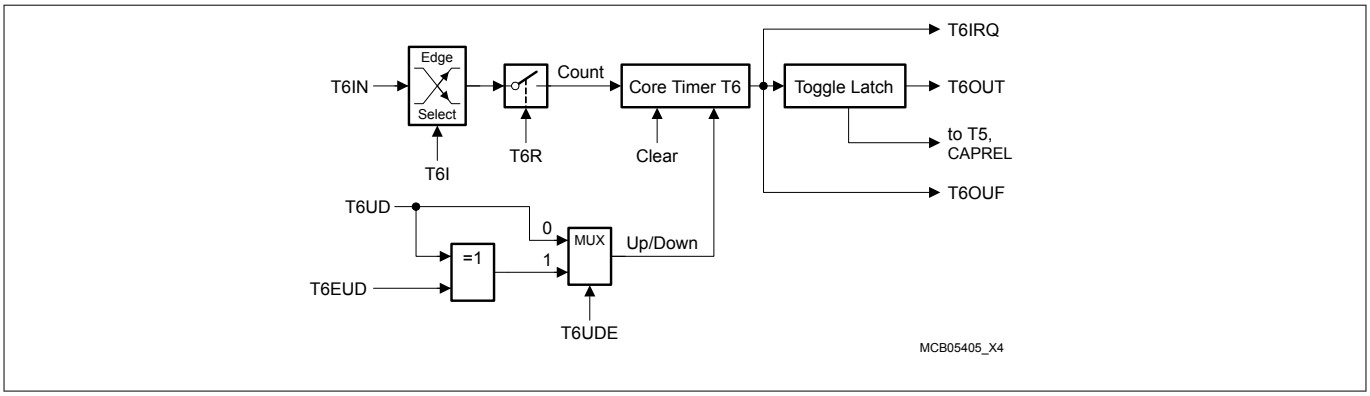


Figure 105 Block diagram of core timer T6 in counter mode

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 Chapter 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 bit field T5M in register T5CON to 000_B.

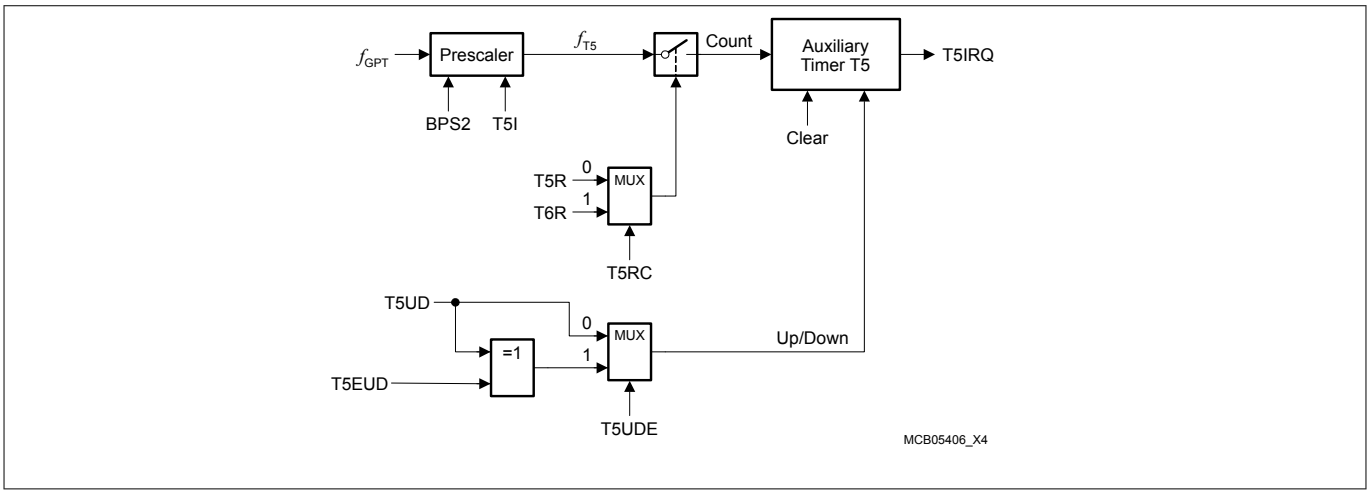


Figure 106 Block diagram of auxiliary timer T5 in timer mode

Timer T5 in gated timer mode

Gated timer mode for the auxiliary timer T5 is selected by setting bit field 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.

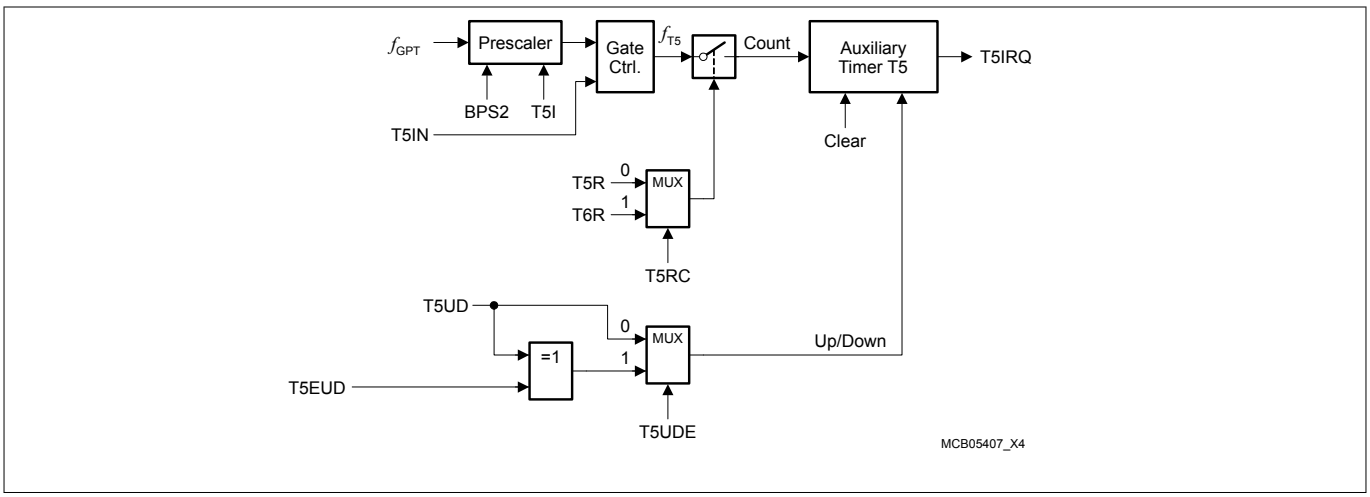


Figure 107 Block diagram of auxiliary timer T5 in gated timer mode

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 bit field 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. Bit field T5I in control register T5CON selects the triggering transition (see Table 99).

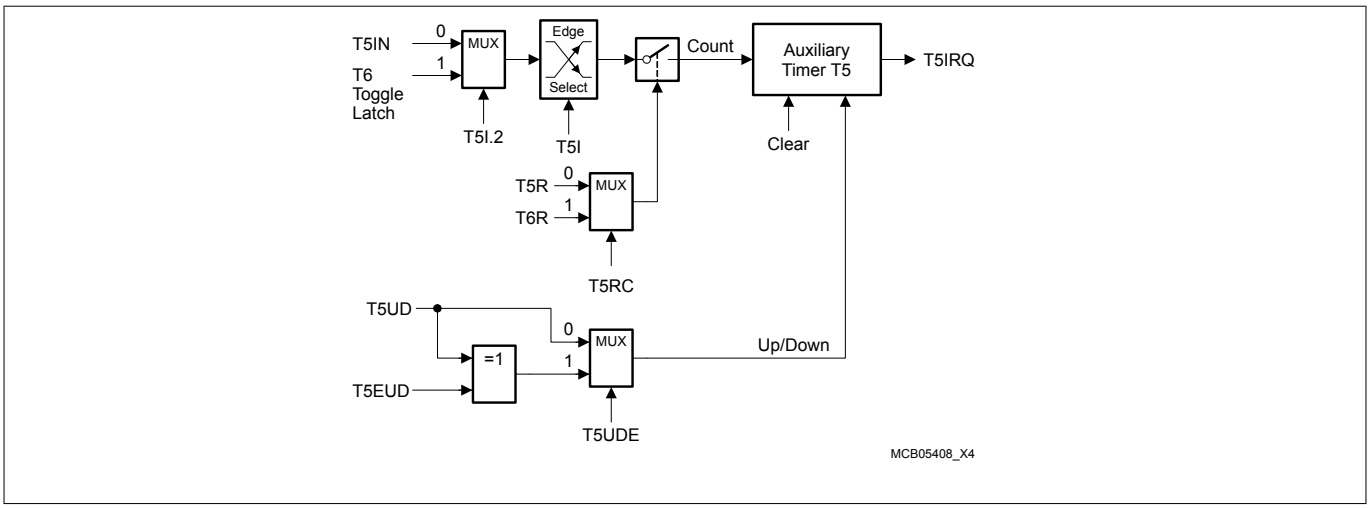


Figure 108 Block diagram of auxiliary timer T5 in counter mode

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 Chapter 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.

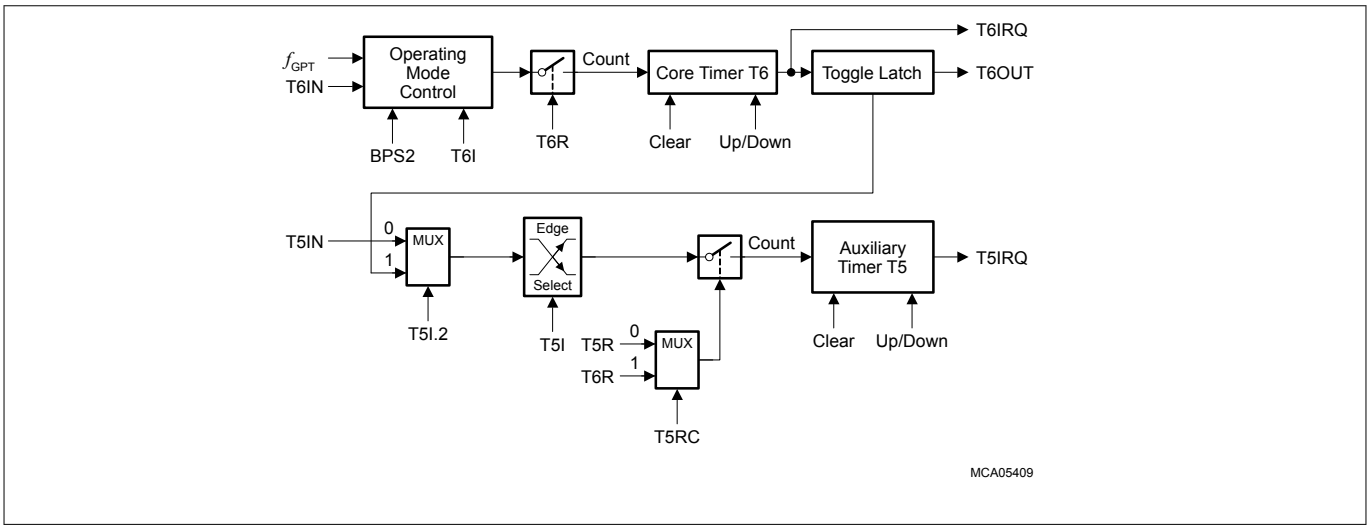


Figure 109 Concatenation of core timer T6 and auxiliary timer T5

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 bit field 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 bit field CI in register T5CON. Table 92 summarizes these options.

Table 92 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).

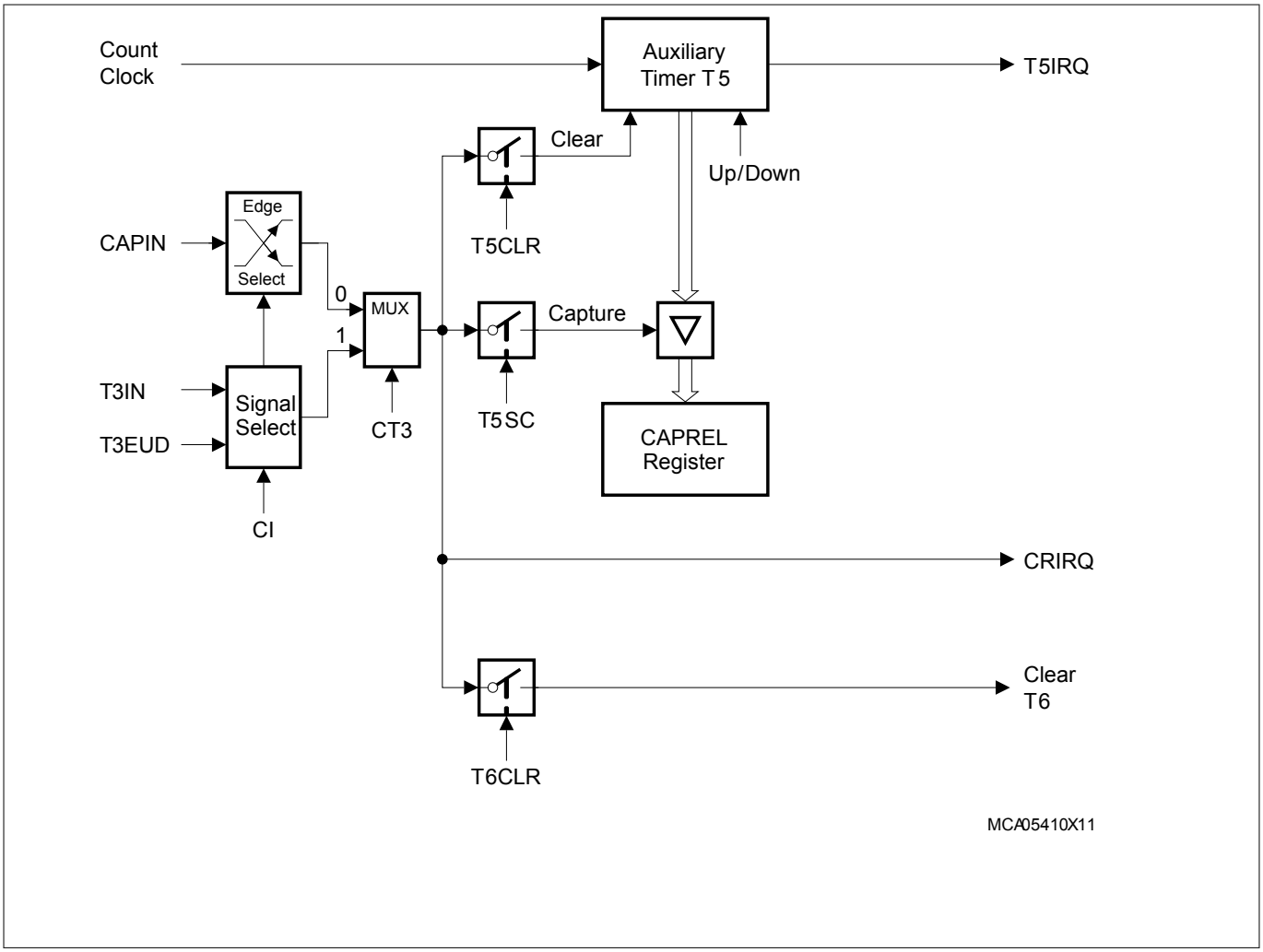


Figure 110 Capture/reload register CAPREL in capture mode

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 GPTM1IEN and GPTM1IRC.

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 Chapter 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.

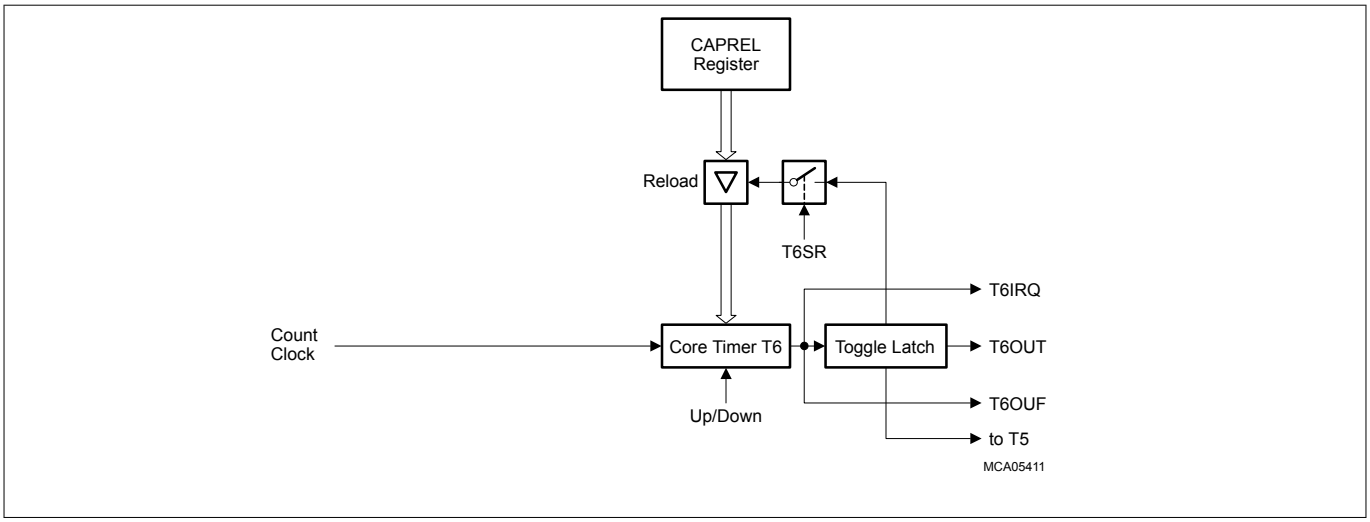


Figure 111 Capture/reload register CAPREL in reload mode

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.

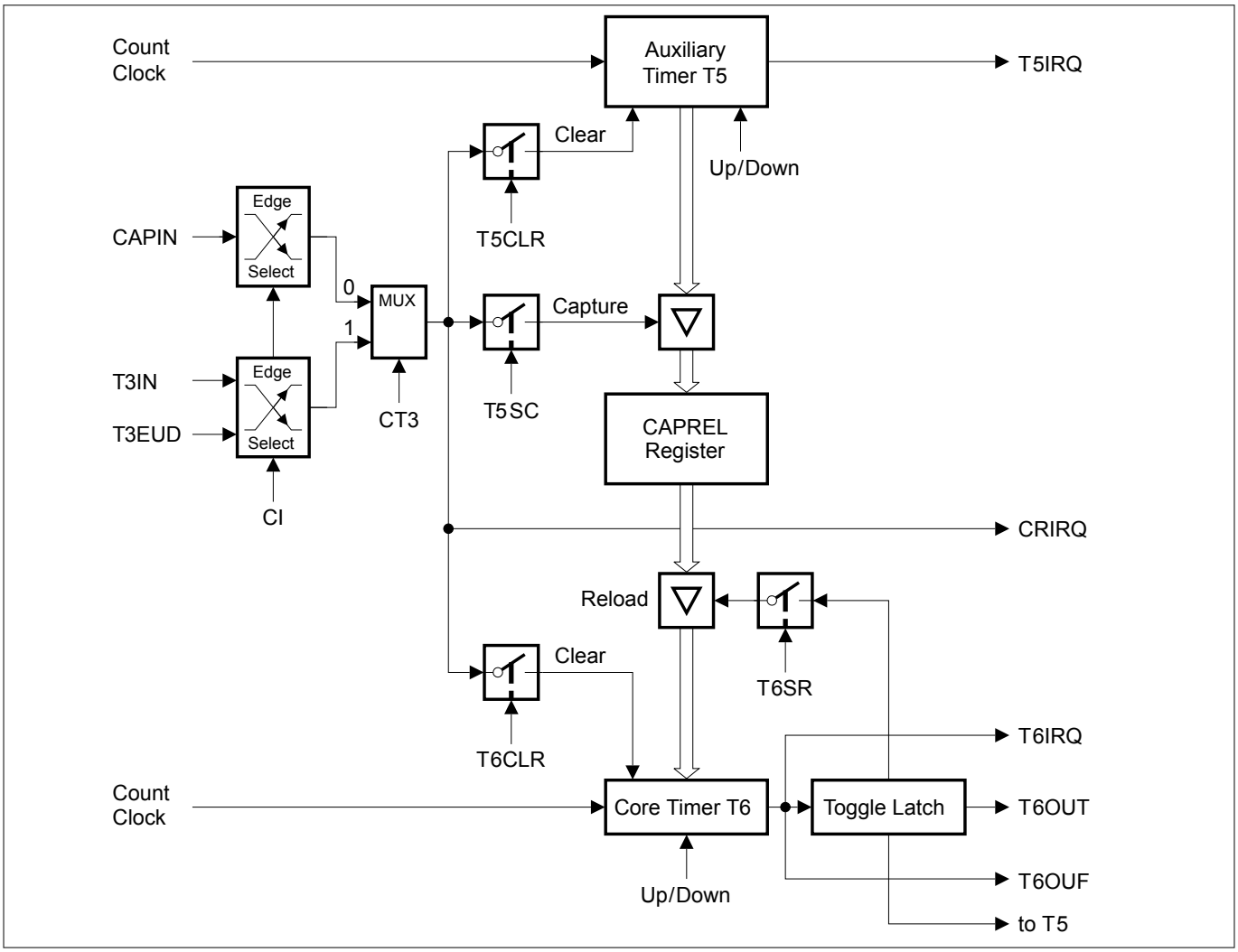


Figure 112 Capture/reload register CAPREL in capture and reload mode

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 (that is 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 bit field 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 bit field CI in register T5CON. Bit field 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 bit field BPS2 in register T6CON (see Figure 83). 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 93 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 bit field 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 bit-field 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:

$$f_{Tx} = \frac{f_{GPT}}{F(BPS2) \times 2^{<TxI>}} \quad r_{Tx}[\mu s] = \frac{F(BPS2) \times 2^{<TxI>}}{f_{GPT}[MHz]} \tag{7}$$

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 98 summarizes the resulting overall divider factors for a GPT2 timer that result from these cascaded prescalers.

Table 94 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 94 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 83). 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 95 summarizes the resulting requirements for external GPT2 input signals.

Table 95 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 Register overview - GPT2 registers (ascending offset address)

Table 96 Register overview - GPT2 registers (ascending offset address)

Short name	Long name	Offset address	Page number
GPT12E_T5CON	Timer T5 control register	0014_H	464
GPT12E_T6CON	Timer T6 control register	0018_H	466
GPT12E_CAPREL	Capture/reload register	$001C_H$	468
GPT12E_T5	Timer 5 count register	$002C_H$	469
GPT12E_T6	Timer 6 count register	0030_H	470

15.4.8.2 GPT2 timer and CAPREL interrupt control registers

The interrupt control register for GPT2 and CAPREL are located in the System control unit - digital modules (SCU-DM) module.

15.4.8.3 GPT2 encoding

15.4.8.3.1 Encoding of timer count direction control

Table 97 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

15.4.8.3.2 Timer mode and gated timer mode: encoding of overall prescaler Factor

Table 98 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 bit field BPS2.

15.4.8.3.3 Counter mode: encoding of input edge selection

Table 99 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

Table 100 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

15.5 Miscellaneous GPT12 registers

The registers listed below are not assigned to a specific timer block. They control general functions and/or give general information.

15.5.1 Register overview - GPT12 registers (ascending offset address)

Table 101 Register overview - GPT12 registers (ascending offset address)

Short name	Long name	Offset address	Page number
GPT12E_ID	Module identification register	0000_H	452
GPT12E_PISEL	Port input select register	0004_H	453

15.6 Implementation of the GPT12 module

This chapter describes the implementation of the GPT12 module in the TLE984x device.

15.6.1 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 TLE984x system clock, so $f_{GPT} = f_{SYS}$.
- External connections interface the timers with external resources via port pins.

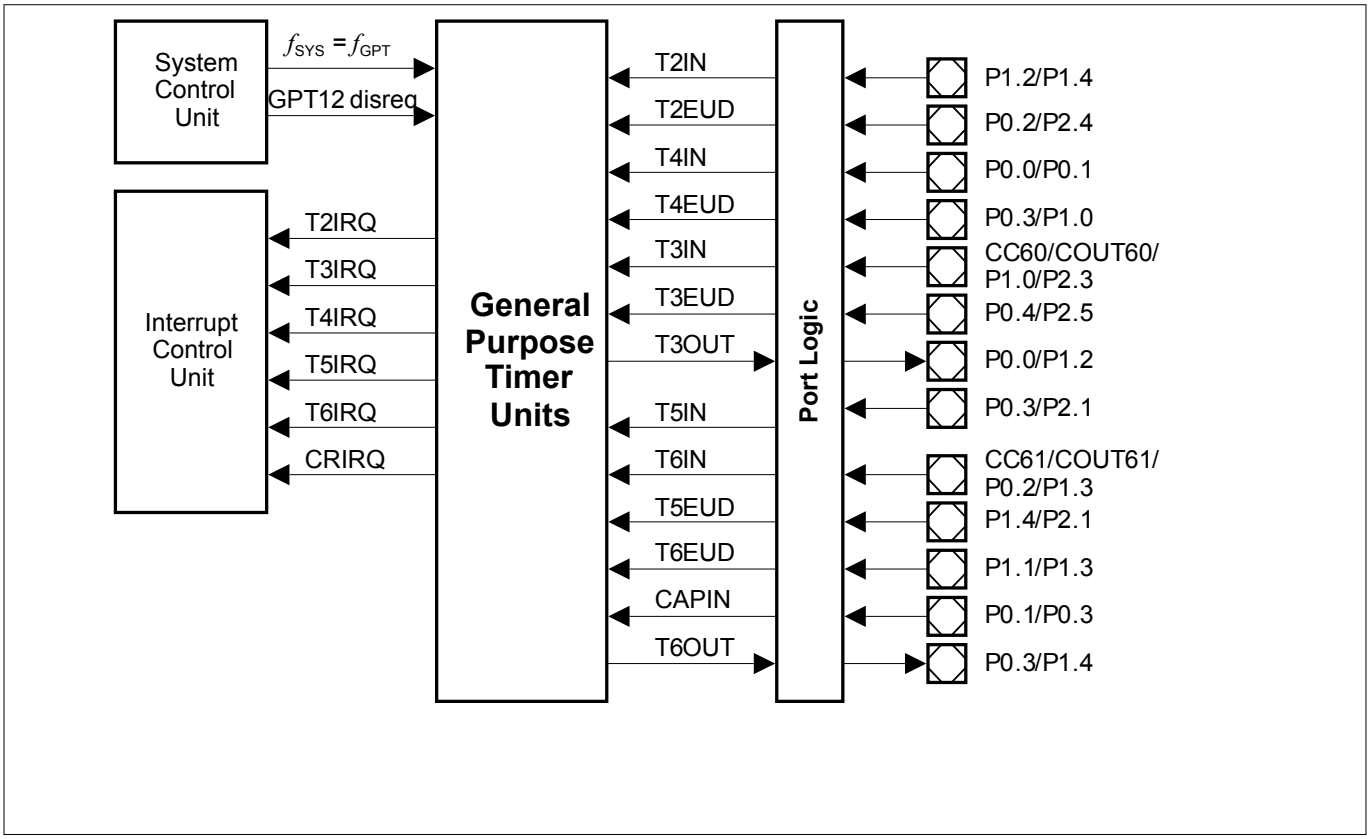


Figure 113 GPT module interfaces

Note: The GPT12E output signal 'T6OFL' is connected to the CAPCOM2 input 'TOUF' and to the GSC.

The following Table 102 (GPT12) shows the digital connections of the GPT12 module with other modules or pins in the TLE984x device.

Table 102 GPT12 digital connections in TLE984x

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	

(table continues...)

Table 102 (continued) GPT12 digital connections in TLE984x

Signal	From/to Module	I/O to GPT	Can be used to/as
T2IRQ	ICU/SCU	O	Interrupt request from timer T2
T3INA	CC60	I	Count input signals for timer T3
T3INB	GPT12PISEL	I	
T3INC	P1.0	I	
T3IND	P2.3	I	
T3EUDA	P0.4	I	Direction input signals for timer T3
T3EUDB	P2.5	I	
T3EUDC	P1.1	I	
T3EUDD	P0.3	I	
T3OUT_0, _1	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	
T4EUDC	P2.5	I	
T4EUDD	P2.6	I	
T4IRQ	ICU/SCU	O	Interrupt request from timer T4
T5INA	P0.5	I	Count input signals for timer T5
T5INB	P1.1	I	
T5EUDA	P1.4	I	Direction input signals for timer T5
T5EUDB	P2.0	I	
T5IRQ	ICU/SCU	O	Interrupt request from timer T5
T6INA	CC61	I	Count input signals for timer T6
T6INB	COUT61	I	
T6EUDA	P1.1	I	Direction input signals for timer T6
T6EUDB	P2.2	I	
T6OUT_1, _0	P0.3	O	Count output signal for timer T6
	P0.1	O	
T6IRQ	ICU/SCU	O	Interrupt request from timer T6

(table continues...)

Table 102 (continued) GPT12 digital connections in TLE984x

Signal	From/to Module	I/O to GPT	Can be used to/as
T6OFL	P0.3	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 (bit field 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 (bit field 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.

15.7 General purpose timer units (GPT12E) register definition

The registers are addressed wordwise.

15.7.1 Register address space - GPT12E

Table 103 Registers address space - GPT12E

Module	Base address	End address	Note
GPT12E	40010000_H	$40013FFF_H$	General purpose timer units (GPT12) registers

15.7.2 Register overview - GPT12E (ascending offset address)

Table 104 Register overview - GPT12E (ascending offset address)

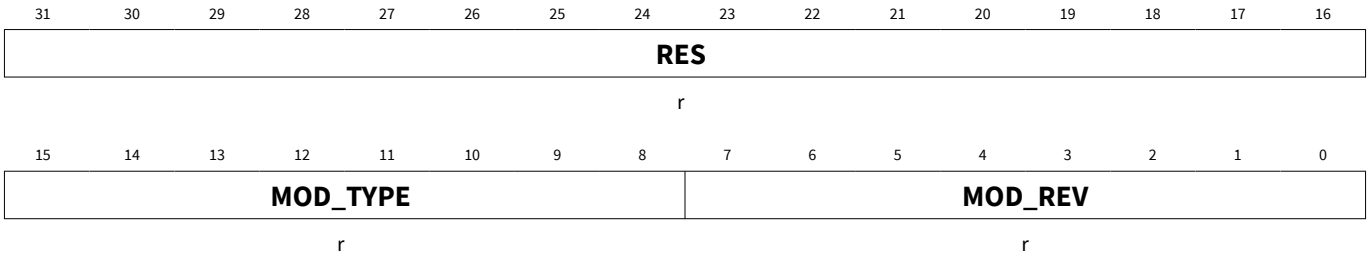
Short name	Long name	Offset address	Page number
GPT12E_ID	Module identification register	0000_H	452
GPT12E_PISEL	Port input select register	0004_H	453
GPT12E_T2CON	Timer T2 control register	0008_H	455
GPT12E_T3CON	Timer T3 control register	$000C_H$	457
GPT12E_T4CON	Timer T4 control register	0010_H	459
GPT12E_T5CON	Timer T5 control register	0014_H	464
GPT12E_T6CON	Timer T6 control register	0018_H	466
GPT12E_CAPREL	Capture/reload register	$001C_H$	468
GPT12E_T2	Timer T2 count register	0020_H	461
GPT12E_T3	Timer T3 count register	0024_H	462
GPT12E_T4	Timer T4 count register	0028_H	463
GPT12E_T5	Timer 5 count register	$002C_H$	469
GPT12E_T6	Timer 6 count register	0030_H	470

15.7.3 Module identification register

Register ID indicates the module version.

GPT12E_ID — Offset address: 0000_H

Module identification register — RESET_TYPE_3 value: $0000\ 5804_H$



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 bit field defines the module identification number (58_H = GPT12E).
RES	31:16	r	**Reserved**

15.7.4 Port input select register

Register PISEL selects timer input signal from several sources under software control.

GPT12E_PISEL Offset address: 0004_H

Port input select register RESET_TYPE_3 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
ISCAPIN		IST6EUD	IST6IN	IST5EUD	IST5IN	IST4EUD		IST4IN		IST3EUD		IST3IN		IST2EUD	IST2IN
rw		rw	rw	rw	rw	rw		rw		rw		rw		rw	rw

Field	Bits	Type	Description
IST2IN	0	rw	**Input select for T2IN** 0_B **T2INA**: Signal T2INA is selected 1_B **T2INB**: Signal T2INB is selected
IST2EUD	1	rw	**Input select for T2EUD** 0_B **T2EUDA**: Signal T2EUDA is selected 1_B **T2EUDB**: Signal T2EUDB is selected
IST3IN	3:2	rw	**Input select for T3IN** 00_B **T3INA**: Signal T3INA is selected 01_B **T3INB**: Signal T3INB is selected 10_B **T3INC**: Signal T3INC is selected 11_B **T3IND**: Signal T3IND is selected
IST3EUD	5:4	rw	**Input select for T3EUD** 00_B **T3EUDA**: Signal T3EUDA is selected 01_B **T3EUDB**: Signal T3EUDB is selected 10_B **T3EUDC**: Signal T3EUDC is selected 11_B **T3EUDD**: Signal T3EUDD is selected
IST4IN	7:6	rw	**Input select for T4IN** 00_B **T4INA**: Signal T4INA is selected 01_B **T4INB**: Signal T4INB is selected 10_B **T4INC**: Signal T4INC is selected 11_B **T4IND**: Signal T4IND is selected
IST4EUD	9:8	rw	**Input select for TEUD** 00_B **T4EUDA**: Signal T4EUDA is selected 01_B **T4EUDB**: Signal T4EUDB is selected 10_B **T4EUDC**: Signal T4EUDC is selected 11_B **T4EUDD**: Signal T4EUDD is selected
IST5IN	10	rw	**Input select for T5IN** 0_B **T5INA**: Signal T5INA is selected

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **T5INB**: Signal T5INB is selected
IST5EUD	11	rw	**Input select for T5EUD** 0_B **T5EUDA**: Signal T5EUDA is selected 1_B **T5EUDB**: Signal T5EUDB is selected
IST6IN	12	rw	**Input select for T6IN** 0_B **T6INA**: Signal T6INA is selected 1_B **T6INB**: Signal T6INB is selected
IST6EUD	13	rw	**Input select for T6EUD** 0_B **T6EUDA**: Signal T6EUDA is selected 1_B **T6EUDB**: Signal T6EUDB is selected
ISCAPIN	15:14	rw	**Input select for CAPIN** 00_B **CAPINA**: Signal CAPINA is selected 01_B **CAPINB**: Signal CAPINB is selected 10_B **CAPINC**: Signal CAPINC (read trigger from T3) is selected 11_B **CAPIND**: Signal CAPIND (read trigger from T2 or T3 or T4) is selected
RES	31:16	r	**Reserved**

15.7.5 Timer T2 control register

GPT12E_T2CON Offset address: 0008_H

Timer T2 control register RESET_TYPE_3 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
T2DIR	T2CHDIR	T2EDGE	T2IRIDIS	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 "GPT1 overall prescaler factors for internal count clock (Timer mode and Gated Timer mode)" Table "GPT1 auxiliary timers T2/T4 input edge selection (Counter mode, Reload mode)" Table "GPT1 auxiliary timers T2/T4 input edge selection (Capture mode)" Table "GPT1 core timer T3 input edge selection (Incremental Interface mode)"
T2M	5:3	rw	**Timer T2 input mode control** 000_B **TIMER_MODE**: Timer mode 001_B **COUNTER_MODE**: Counter mode 010_B **GATED_LOW**: Gated timer mode with gate active low 011_B **GATED_HIGH**: Gated timer mode with gate active high 100_B **RELOAD_MODE**: Reload mode 101_B **CAPTURE_MODE**: Capture mode 110_B **INCREMENTAL_INTERFACE_MODE**: Rotation detection mode 111_B **INCREMENTAL_INTERFACE_MODE**: Edge detection mode
T2R	6	rw	**Timer T2 input run bit** 0_B **STOP**: Timer T2 stops 1_B **RUN**: Timer T2 runs
T2UD	7	rw	**Timer T2 up/down control** See Table "GPT2 timer count direction control for encoding of bits T2UD and T2UDE". 0_B **UP**: Timer T2 counts up 1_B **DOWN**: Timer T2 counts down
T2UDE	8	rw	**Timer T2 external up/down enable** See Table "GPT2 timer count direction control for encoding of bits T2UD and T2UDE".

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **T2UD**: Count direction is controlled by bit T2UD; input T2EUD is disconnected 1_B **T2EUD**: Count direction is controlled by input T2EUD
T2RC	9	rw	**Timer T2 remote control** 0_B **T2R**: Timer T2 is controlled by its own run bit T2R 1_B **T3R**: Timer T2 is controlled by the run bit T3R of core timer T3, not by bit T2R
RES	11:10, 31:16	r	**Reserved** Read as 0; should be written with 0.
T2IRIDIS	12	rw	**Timer T2 interrupt disable** 0_B **ENABLED**: Interrupt generation for T2CHDIR and T2EDGE interrupts in incremental interface mode is enabled 1_B **DISABLED**: 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**: No count edge was detected 1_B **COUNT**: A count edge was detected
T2CHDIR	14	rwh	**Timer T2 count direction change** This bit is set each time the count direction of timer T2 changes. T2CHDIR must be cleared by software. 0_B **NO_CHANGE**: No change of count direction was detected 1_B **CHANGE**: A change of count direction was detected
T2DIR	15	rh	**Timer T2 rotation direction** 0_B **UP**: Timer T2 counts up 1_B **DOWN**: Timer T2 counts down

15.7.6 Timer T3 control register

GPT12E_T3CON Offset address: $000C_H$

Timer T3 control register RESET_TYPE_3 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
T3DIR	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 "GPT1 overall prescaler factors for internal count clock (Timer mode and Gated Timer mode)" Table "GPT1 core timer T3 input edge selection (Counter mode)" Table "GPT1 core timer T3 input edge selection (Incremental Interface mode)"
T3M	5:3	rw	**Timer T3 input mode control** 000_B **TIMER_MODE**: Timer mode 001_B **COUNTER_MODE**: Counter mode 010_B **GATED_LOW**: Gated timer mode with gate active low 011_B **GATED_HIGH**: 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 input run bit** 0_B **STOP**: Timer T3 stops 1_B **RUN**: Timer T3 runs
T3UD	7	rw	**Timer T3 up/down control** See Table "GPT2 timer count direction control for encoding of bits T3UD and T3UDE". 0_B **UP**: Timer T3 counts up 1_B **DOWN**: Timer T3 counts down
T3UDE	8	rw	**Timer T3 external up/down enable** See Table "GPT2 timer count direction control for encoding of bits T3UD and T3UDE". 0_B **T3UD**: Count direction is controlled by bit T3UD; input T3EUD is disconnected 1_B **T3EUD**: Count direction is controlled by input T3EUD

(table continues...)

(continued)

Field	Bits	Type	Description
T3OE	9	rw	**Overflow/underflow output enable** 0_B **DISABLED**: Alternate output function disabled 1_B **T3OUT**: State of T3 toggle latch is output on pin T3OUT
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** Select basic clock for block GPT1 (see also Chapter "GPT1 clock signal control)" 00_B **8**: fGPT/8 01_B **4**: fGPT/4 10_B **32**: fGPT/32 11_B **16**: fGPT/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**: No count edge was detected 1_B **COUNT**: 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**: No change of count direction was detected 1_B **CHANGE**: A change of count direction was detected
T3DIR	15	rh	**Timer T3 rotation direction flag** 0_B **UP**: Timer T3 counts up 1_B **DOWN**: Timer T3 counts down
RES	31:16	r	**Reserved** Read as 0; should be written with 0.

15.7.7 Timer T4 control register

GPT12E_T4CON Offset address: 0010_H

Timer T4 control register RESET_TYPE_3 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
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 "GPT1 overall prescaler factors for internal count clock (Timer mode and Gated Timer mode)" Table "GPT1 auxiliary timers T2/T4 input edge selection (Counter mode, Reload mode)" Table "GPT1 auxiliary timers T2/T4 input edge selection (Capture mode)" Table "GPT1 core timer T3 input edge selection (Incremental Interface mode)"
T4M	5:3	rw	**Timer T4 mode control (basic operating mode)** 000_B **TIMER_MODE**: Timer mode 001_B **COUNTER_MODE**: Counter mode 010_B **GATED_LOW**: Gated timer mode with gate active low 011_B **GATED_HIGH**: Gated timer mode with gate active high 100_B **RELOAD_MODE**: Reload mode 101_B **CAPTURE_MODE**: Capture mode 110_B **INCREMENTAL_INTERFACE_MODE**: Rotation detection mode 111_B **INCREMENTAL_INTERFACE_MODE**: Edge detection mode
T4R	6	rw	**Timer T4 input run bit** 0_B **STOP**: Timer T4 stops 1_B **RUN**: Timer T4 runs
T4UD	7	rw	**Timer T4 up/down control** See Chapter "GPT1 clock signal control for encoding of bits T4UD and T4UDE". 0_B **UP**: Timer T4 counts up 1_B **DOWN**: Timer T4 counts down
T4UDE	8	rw	**Timer T4 external up/down enable** See Chapter "GPT1 clock signal control for encoding of bits T4UD and T4UDE".

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **T4UD**: Count direction is controlled by bit T4UD; input T4EUD is disconnected 1_B **T4EUD**: Count direction is controlled by input T4EUD
T4RC	9	rw	**Timer T4 remote control** 0_B **T4R**: Timer T4 is controlled by its own run bit T4R 1_B **T3R**: Timer T4 is controlled by the run bit T3R of core timer T3, but not by bit T4R
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**: No effect of T4EUD on timer T2 1_B **CLEAR**: 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**: No effect of T4IN on timer T3 1_B **CLEAR**: A falling edge on T4In clears timer T3
T4IRDIS	12	rw	**Timer T4 interrupt disable** 0_B **ENABLED**: Interrupt generation for T4CHDIR and T4EDGE interrupts in incremental interface mode is enabled 1_B **DISABLED**: Interrupt generation for T4CHDIR and T4EDGE interrupts in incremental interface mode is disabled
T4EDGE	13	rwh	**Timer T4 edge direction** The bit is set each time a count edge is detected. T4EDGE has to be cleared by software. 0_B **NO_COUNT**: No count edge was detected 1_B **COUNT**: A count edge was detected
T4CHDIR	14	rwh	**Timer T4 count direction change** The bit is set each time a count direction of timer T4 changes. T4EDGE must be cleared by software 0_B **NO_CHANGE**: No change in count direction was detected 1_B **CHANGE**: A change in count direction was detected
T4RDIR	15	rh	**Timer T4 rotation direction** 0_B **UP**: Timer T4 counts up 1_B **DOWN**: Timer T4 counts down
RES	31:16	r	**Reserved** Read as 0; should be written with 0.

15.7.8 Timer T2 count register

GPT12E_T2 Offset address: 0020_H

Timer T2 count register RESET_TYPE_3 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
T2															
rwh															

Field	Bits	Type	Description
T2	15:0	rwh	**Timer T2 current value** Contains the current value of the timer T2.
RES	31:16	r	**Reserved**

15.7.9 Timer T3 count register

GPT12E_T3 Offset address: 0024_H

Timer T3 count register RESET_TYPE_3 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
T3															
rwh															

Field	Bits	Type	Description
T3	15:0	rwh	**Timer T3 current value** Contains the current value of the timer T3.
RES	31:16	r	**Reserved**

15.7.10 Timer T4 count register

GPT12E_T4 Offset address: 0028_H

Timer T4 count register RESET_TYPE_3 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
T4															
rw															

Field	Bits	Type	Description
T4	15:0	rw	**Timer T4 current value** Contains the current value of the timer T4.
RES	31:16	r	**Reserved**

15.7.11 Timer T5 control register

GPT12E_T5CON Offset address: 0014_H

Timer T5 control register RESET_TYPE_3 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
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 "GPT2 overall prescaler factors for internal count clock (Timer mode and Gated Timer mode)" Table "GPT2 Auxiliary timer T5 input edge selection (Counter mode)"
T5M	4:3	rw	**Timer T5 input mode control** 00_B **TIMER_MODE**: Timer mode 01_B **COUNTER_MODE**: Counter mode 10_B **GATED_LOW**: Gated timer mode with gate active low 11_B **GATED_HIGH**: Gated timer mode with gate active high
RES	5, 11, 31:16	r	**Reserved** Bit 5 contains the current value of the CAPREL register.
T5R	6	rw	**Timer T5 run bit** 0_B **STOP**: Timer T5 stops 1_B **RUN**: Timer T5 runs
T5UD	7	rw	**Timer T5 up/down control** See Table "GPT2 timer count direction control for encoding of bits T5UD and T5UDE". Rising edge must be selected if capturing is triggered by the internal GPT1 read signals (see register PISEL and "Combined capture modes"). 0_B **UP**: Timer T5 counts up 1_B **DOWN**: Timer T5 counts down
T5UDE	8	rw	**Timer T5 external up/down enable** See Table "GPT2 timer count direction control for encoding of bits T5UD and T5UDE". 0_B **T5UD**: Count direction is controlled by bit T5UD; input T5EUD is disconnected 1_B **T5EUD**: Count direction is controlled by input T5EUD
T5RC	9	rw	**Timer T5 remote control**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **T5R**: Timer T5 is controlled by its own run bit T5R 1_B **T6R**: 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 **CAPIN**: Capture trigger from input line CAPIN 1_B **T3IN**: Capture trigger from T3 input lines T3IN and/or T3EUD
CI	13:12	rw	**Register CAPREL capture trigger selection** To define the respective trigger source signal, also bit CT3 must be regarded (see Table "CAPREL register input edge selection"). 00_B **DISABLED**: Capture disabled 01_B **POSITIVE**: Positive transition (rising edge) on CAPIN Rising edge must be selected if capturing is triggered by the internal GPT1 read signals (see register PISEL and Combined Capture modes) or any transition on T3IN 10_B **NEGATIVE**: Negative transition (falling edge) on CAPIN or any transition on T3EUD 11_B **ANY**: 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 **NOT_CLEARED**: Timer T5 is not cleared on a capture event 1_B **CLEARED**: Timer T5 is cleared on a capture event
T5SC	15	rw	**Timer T5 capture mode enable** 0_B **DISABLED**: Capture into register CAPREL disabled 1_B **ENABLED**: Capture into register CAPREL enabled

15.7.12 Timer T6 control register

GPT12E_T6CON Offset address: 0018_H

Timer T6 control register RESET_TYPE_3 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
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 "GPT2 overall prescaler factors for internal count clock (Timer mode and Gated Timer mode)" Table "GPT2 auxiliary timer T5 input edge selection (Counter mode)
T6M	5:3	rw	**Timer T6 mode control** 000_B **TIMER_MODE**: Timer mode 001_B **COUNTER_MODE**: Counter mode 010_B **GATED_LOW**: Gated timer mode with gate active low 011_B **GATED_HIGH**: Gated timer mode with gate active high 100_B **RESERVED**: Do not use this combination … 111_B **RESERVED**: Do not use this combination
T6R	6	rw	**Timer T6 input run bit** 0_B **STOP**: Timer T3 stops 1_B **RUN**: Timer T3 runs
T6UD	7	rw	**Timer T6 up/down control** See Table "GPT2 timer count direction control for encoding of bits T6UD and T6UDE". 0_B **UP**: Timer T3 counts up 1_B **DOWN**: Timer T3 counts down
T6UDE	8	rw	**Timer T6 external up/down enable** See Table "GPT2 timer count direction control for encoding of bits T6UD and T6UDE". 0_B **T6UD**: Count direction is controlled by bit T6UD; input T6EUD is disconnected 1_B **T6EUD**: Count direction is controlled by input T6EUD
T6OE	9	rw	**Overflow/underflow output enable** 0_B **DISABLED**: Alternate output function disabled 1_B **T6OUT**: State of T6 toggle latch is output on pin T6OUT

(table continues...)

(continued)

Field	Bits	Type	Description
T6OTL	10	rwh	**Timer T6 overflow toggle latch** Toggles on each overflow/underflow of T6. Can be set or cleared by software (see separate description).
BPS2	12:11	rw	**GPT2 block prescaler control** Select basic clock for block GPT1 (see also Chapter "GPT2 clock signal control" 00_B **4**: fGPT/4 01_B **2**: fGPT/2 10_B **16**: fGPT/16 11_B **8**: fGPT/8
RES	13, 31:16	r	**Reserved**
T6CLR	14	rw	**Timer T6 clear enable bit** 0_B **NOT_CLEARED**: Timer T6 is not cleared on a capture event 1_B **CLEARED**: Timer T6 is cleared on a capture event
T6SR	15	rw	**Timer T6 reload mode enable** 0_B **DISABLED**: Reload from register CAPREL disabled 1_B **ENABLED**: Reload from register CAPREL enabled

15.7.13 Capture/reload register

GPT12E_CAPREL Offset address: $001C_H$

Capture/reload register RESET_TYPE_3 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
CAPREL															
rwh															

Field	Bits	Type	Description
CAPREL	15:0	rwh	**Current reload value or captured value** Contains the current value of the timer CAPREL register.
RES	31:16	r	**Reserved**

15.7.14 Timer 5 count register

GPT12E_T5 Offset address: $002C_H$

Timer 5 count register RESET_TYPE_3 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
T5															
rwh															

Field	Bits	Type	Description
T5	15:0	rwh	**Timer T5 current value** Contains the current value of the timer T5.
RES	31:16	r	**Reserved**

15.7.15 Timer 6 count register

GPT12E_T6 Offset address: 0030_H

Timer 6 count register RESET_TYPE_3 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
T6															
rwh															

Field	Bits	Type	Description
T6	15:0	rwh	**Timer T6 current value** Contains the current value of the timer T6.
RES	31:16	r	**Reserved**

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
- Baud-rate generator for U(S)ART

16.2 Introduction

Two functionally identical timers are implemented: Timer2 and Timer21. The description refers to Timer2 only, but applies to Timer21 as well.

The timer modules are general purpose 16-bit timer. Timer2 can function as a timer or counter in each of its modes. As a timer, it counts with an input clock of $f_{sys}/12$ (if prescaler is disabled). As a counter, Timer2 counts 1-to-0 transitions on pin T2. In the counter mode, the maximum resolution for the count is $f_{sys}/24$ (if 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 modes overview

Table 105 Timer2 and Timer21 modes

Mode	Description
Auto-reload	Up/down count disabled • Count up only • Start counting from 16-bit reload value, overflow at $FFFF_H$ • Reload event configurable for trigger by overflow condition only, or by negative/ positive edge at input pin T2EX as well • Programmable reload value in register RC2 • Interrupt is generated with reload events
Auto-reload	Up/down count enabled • Count up or down, direction determined by level at input pin T2EX • No interrupt is generated • Count up - Start counting from 16-bit reload value, overflow at $FFFF_H$ - Reload event triggered by overflow condition - Programmable reload value in register RC2 • Count down - Start counting from $FFFF_H$, underflow at value defined in register RC2

(table continues...)

Table 105 (continued) Timer2 and Timer21 modes

Mode	Description
	- Reload event triggered by underflow condition - Reload value fixed at $FFFF_H$
Channel capture	• Count up only • Start counting 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 • Interrupt is generated with reload or capture event

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 114.

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.

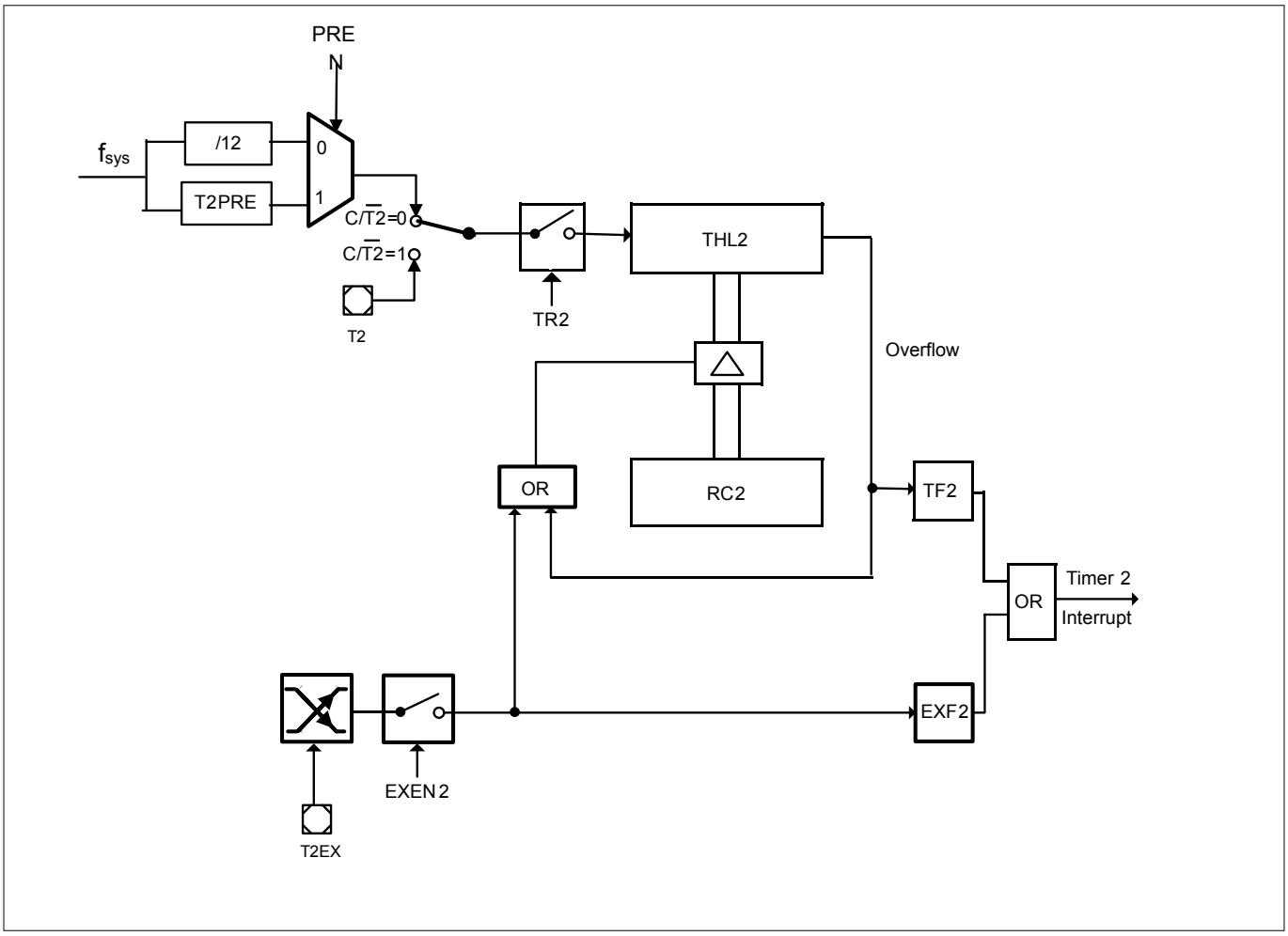


Figure 114 Auto-reload mode (DCEN = 0)

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 115.

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.

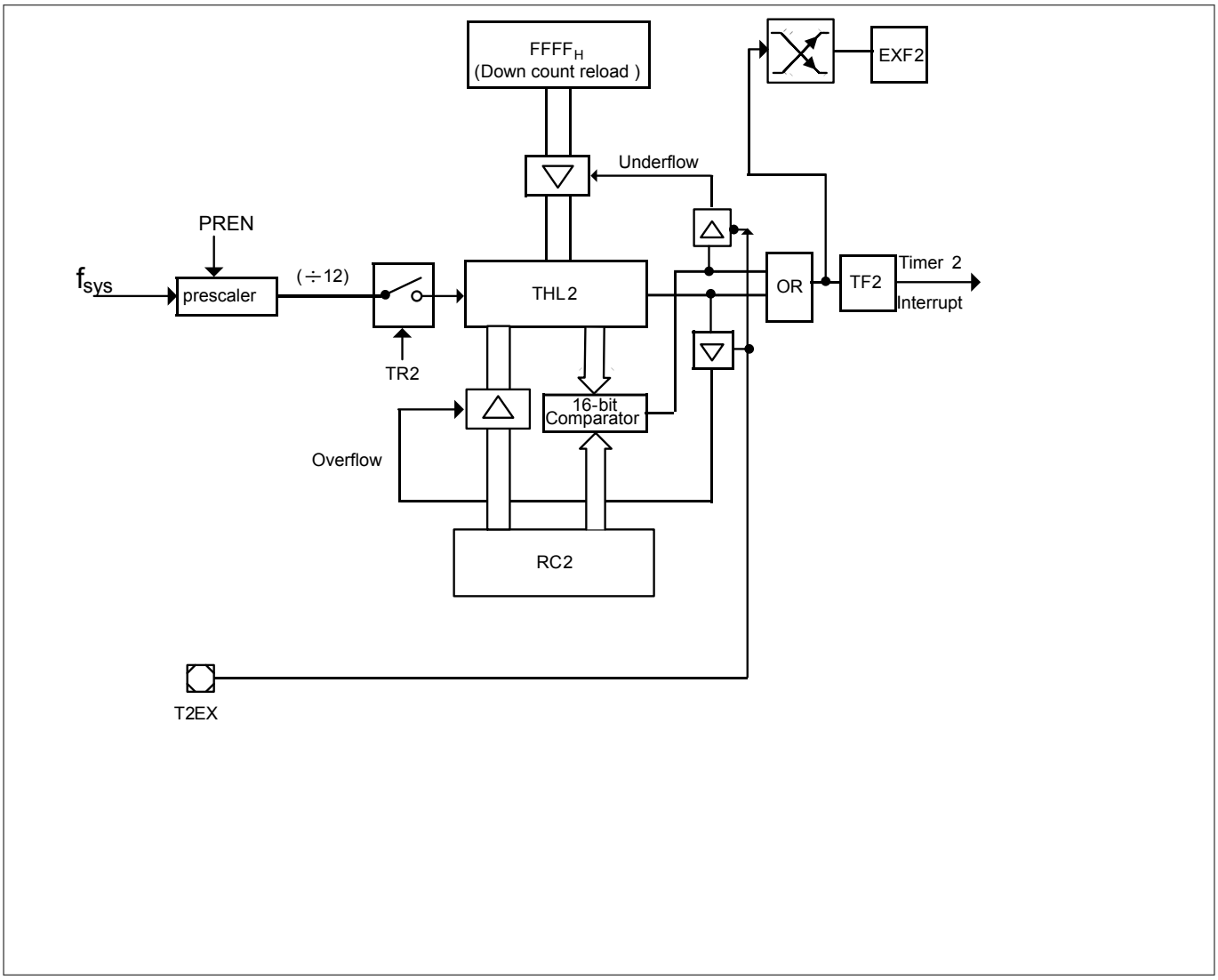


Figure 115 Auto-reload mode (DCEN = 1)

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 f_{sys} cycle. When a sampled input shows a low (high) level in one f_{sys} cycle and a high (low) in the next f_{sys} 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 116 describes the capture function of Timer2.

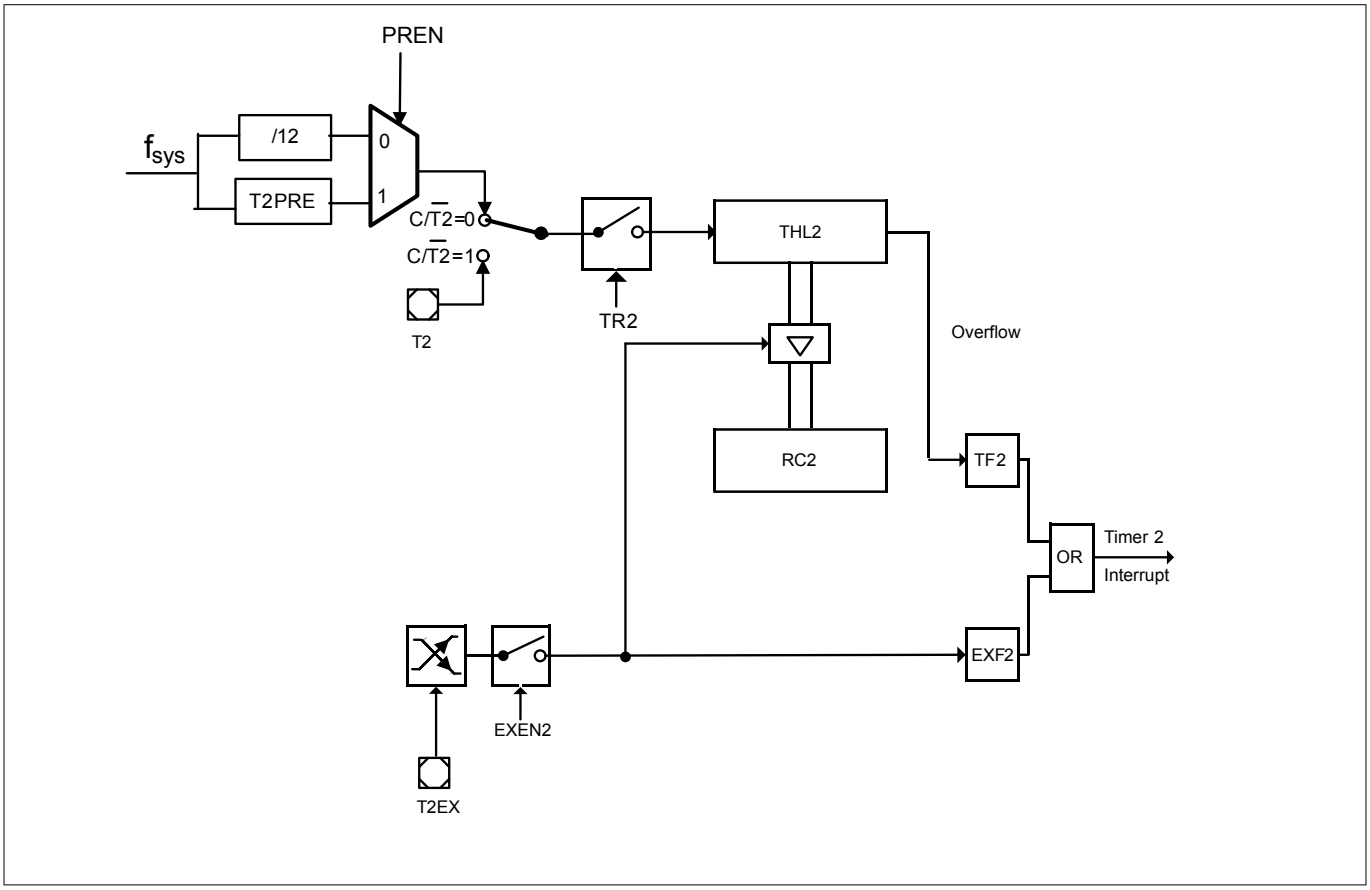


Figure 116 Capture mode

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 $f_{sys}/12$ (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 f_{sys} 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.

16.3.4 Interrupt generation

When an interrupt event happened, the corresponding interrupt flag bit EXF2/TF2 is set. If enabled by the related interrupt enable bit EXF2EN/TF2EN in register T2CON1, 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.

16.4 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.

The registers are addressed wordwise.

Mode register

The T2_MOD register is used to configure Timer2 for various modes of operation.

Control register

The control registers T2_CON, T2_CON1 and T2_ICLR are used to control the operating modes and interrupt of Timer2.

Timer2 reload/capture register

The T2_RC2 register is used for a 16-bit reload of the timer count upon an overflow or a capture of the current timer count depending on the mode selected.

Timer2 count register

The T2_CNT register holds the current 16-bit value of the Timer2 count.

16.4.1 Timer2 and Timer21 (TIMER) register definition

16.4.1.1 Register address space - TIMER

Table 106 Registers address space - TIMER

Module	Base address	End address	Note
T2	48004000_H	$48004FFF_H$	Timer2
T21	48005000_H	$48005FFF_H$	Timer21

16.4.1.2 Register overview - TIMER (ascending offset address)

Table 107 Register overview - TIMER (ascending offset address)

Short name	Long name	Offset address	Page number
T2_CON	Timer2 control register	0000_H	482
T2_MOD	Timer2 mode register	0004_H	480
T2_RC	Timer2 reload/capture register	0008_H	485
T2_CNT	Timer2 count register	0010_H	486
T2_ICLR	Timer2 interrupt clear register	0018_H	483
T2_CON1	Timer2 control 1 register	$001C_H$	484

16.4.1.3 Timer2 mode register

T2_MOD Offset address: 0004_H

Timer2 mode register RESET_TYPE_3 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								T2REGS	T2RHEN	EDGESEL	PREN	T2PRE			DCEN
r								rw	rw	rw	rw	rw			rw

Field	Bits	Type	Description
DCEN	0	rw	**Up/down counter enable** 0_B **DISABLED**: Up/down counter function is disabled 1_B **ENABLED**: 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 **DIV1**: fT2=fSYS 001_B **DIV2**: fT2=fSYS/2 010_B **DIV4**: DfT2=fSYS/4 011_B **DIV8**: fT2=fSYS/8 100_B **DIV16**: fT2=fSYS/16 101_B **DIV32**: fT2=fSYS/32 110_B **DIV64**: fT2=fSYS/64 111_B **DIV128**: fT2=fSYS/128
PREN	4	rw	**Prescaler enable** 0_B **DISABLED**: Prescaler is disabled and the 2 or 12 divider takes effect 1_B **ENABLED**: Prescaler is enabled (see T2PRE bit) and the 2 or 12 divider is bypassed
EDGESEL	5	rw	**Edge select in capture mode/reload mode** 0_B **FALLING**: The falling edge at Pin T2EX is selected 1_B **RISING**: The rising edge at Pin T2EX is selected
T2RHEN	6	rw	**Timer2 external start enable** 0_B **DISABLED**: Timer2 external start is disabled 1_B **ENABLED**: Timer2 external start is enabled
T2REGS	7	rw	**Edge select for Timer2 external start** 0_B **FALLING**: The falling edge at pin T2EX is selected 1_B **RISING**: The rising edge at Pin T2EX is selected
RES	31:8	r	**Reserved**

(table continues...)

(continued)

Field	Bits	Type	Description
			Returns 0 if read. Should be written with 0.

16.4.1.4 Timer2 control register

T2_CON Offset address: 0000_H

Timer2 control register RESET_TYPE_3 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								TF2	EXF2	RES		EXEN2	TR2	C_T2	CP_RL2
r								r	r	r		rw	rwhis	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 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	rwhis	**Timer2 start/stop control** 0_B **STOP**: Timer2 1_B **START**: Timer2
EXEN2	3	rw	**Timer2 external enable control** 0_B **DISABLED**: External events are disabled 1_B **ENABLED**: External events are enabled in capture/reload
RES	5:4, 31:8	r	**Reserved** Returns 0 if read. Should be written with 0.
EXF2	6	r	**Timer2 external flag** In capture/reload/baud-rate generator 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.

16.4.1.5 Timer2 interrupt clear register

T2_ICLR Offset address: 0018_H

Timer2 interrupt clear register RESET_TYPE_3 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								TF2CLR	EXF2CLR	RES					
r								w	w	r					

Field	Bits	Type	Description
RES	5:0, 31:8	r	**Reserved** Always read as 0
EXF2CLR	6	w	**External interrupt clear flag** 0_B **N_A**: External interrupt is not cleared 1_B **Clear**: External interrupt
TF2CLR	7	w	**Overflow/underflow interrupt clear flag** 0_B **N_A**: Overflow/underflow interrupt is not cleared 1_B **Clear**: Overflow/underflow interrupt

16.4.1.6 Timer2 control 1 register

T2_CON1 Offset address: $001C_H$

Timer2 control 1 register RESET_TYPE_3 value: $0000\ 0003_H$

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

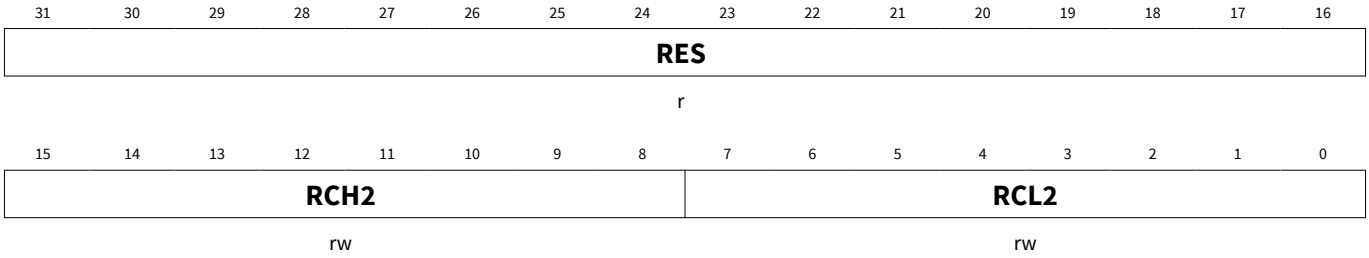
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES														**TF2EN**	**EXF2EN**
r														rw	rw

Field	Bits	Type	Description
EXF2EN	0	rw	**External interrupt enable** 0_B **DISABLE**: External interrupt 1_B **ENABLE**: External interrupt
TF2EN	1	rw	**Overflow/underflow interrupt enable** 0_B **DISABLE**: Overflow/underflow interrupt 1_B **ENABLE**: Overflow/underflow interrupt
RES	31:2	r	**Reserved** Always read as 0

16.4.1.7 Timer2 reload/capture register

T2_RC Offset address: 0008_H

Timer2 reload/capture register RESET_TYPE_3 value: $0000\ 0000_H$

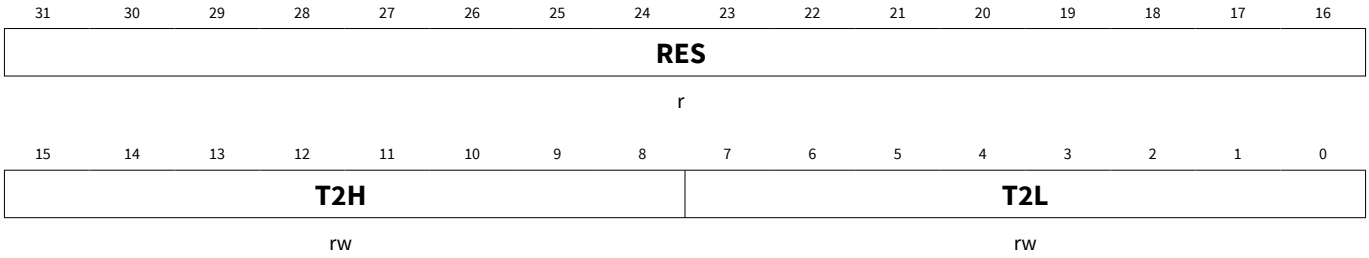


Field	Bits	Type	Description
RCL2	7:0	rw	**Reload/capture value** *Note: Reload/capture value can be set by software (highest priority) and is updated by hardware during capture mode.* These contents are loaded into the timer register upon an overflow condition, if CP_RL2 = 0. If CP_RL2 = 1, this register is loaded with the current timer count upon a negative/positive transition at pin T2EX when EXEN2 = 1.
RCH2	15:8	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.
RES	31:16	r	**Reserved** Always read as 0

16.4.1.8 Timer2 count register

T2_CNT Offset address: 0010_H

Timer2 count register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
T2L	7:0	rw	**Timer2 value** These bits indicate the current timer value T2[7:0]. *Note: Timer2 can be updated by software (highest priority) and is updated by hardware if T2R is set.*
T2H	15:8	rw	**Timer2 value** These bits indicate the current timer value T2[15:8]. *Note: Timer2 can be updated by software (highest priority) and is updated by hardware if T2R is set.*
RES	31:16	r	**Reserved** Always read as 0

16.5 Timer2 and Timer21 implementation details

This section describes:

- the MOTIX™ TLE984xQX module related interfaces such as port connections and interrupt control
- all MOTIX™ TLE984xQX module related registers with their addresses

16.5.1 Interfaces of the Timer2 and Timer21

Overviews of the Timer2 and Timer21 kernel I/O interfaces and interrupt signals are shown in Figure 117 and Figure 118.

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 MOTIX™ TLE984xQX, 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).

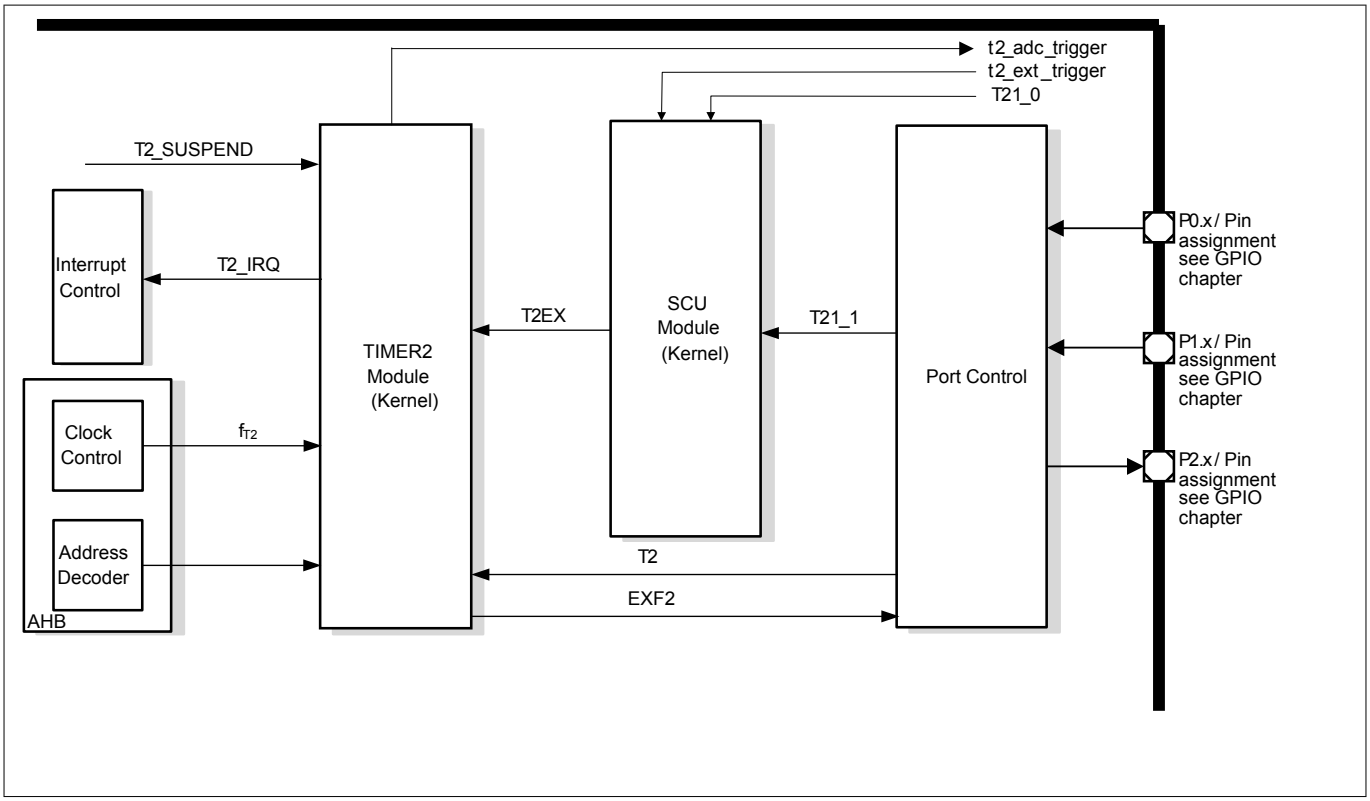


Figure 117 Timer2 module I/O interface

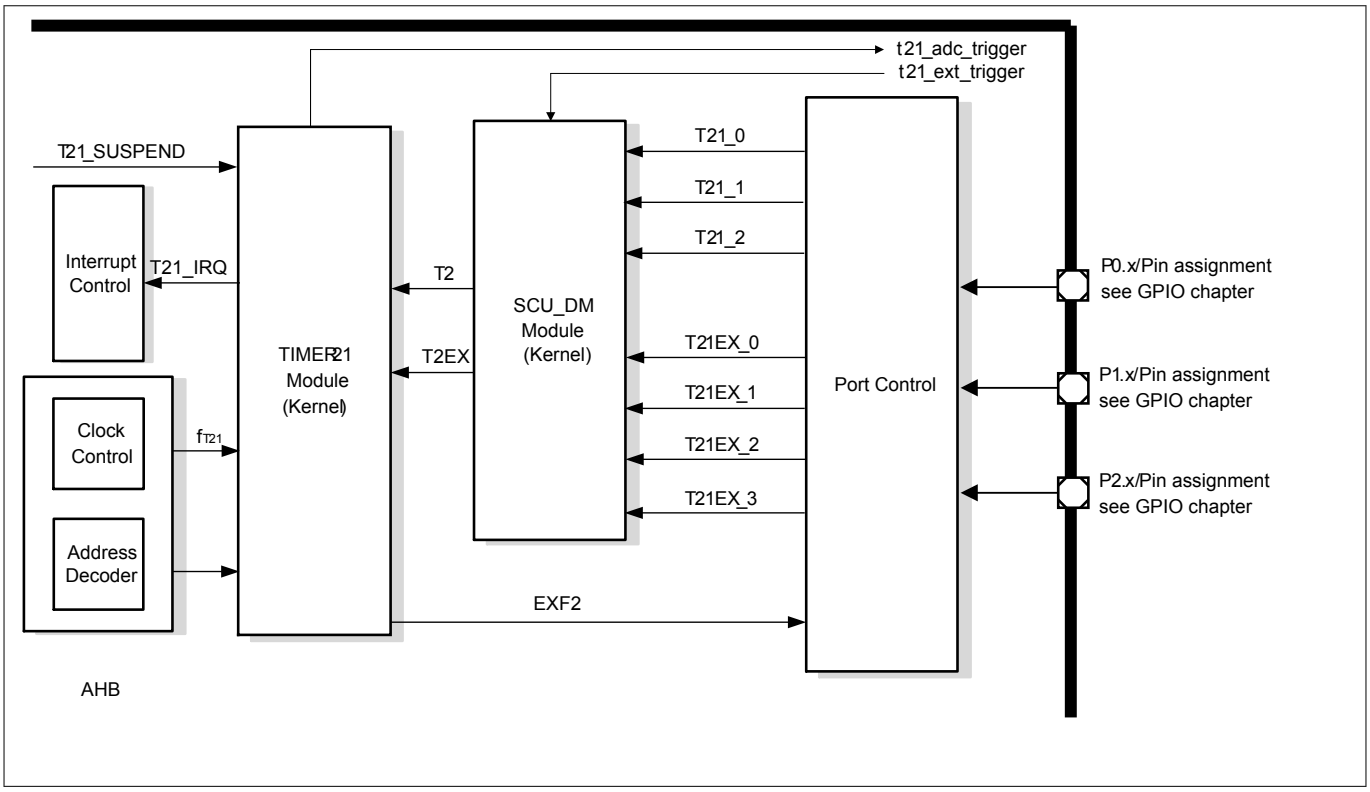


Figure 118 Timer21 module I/O interface

17 Capture/compare unit 6 (CCU6)

This chapter is structured as follows:

- Functional description of the CCU6 kernel (see Chapter 17.2)
  - Introduction (see Chapter 17.2)
  - Operating T12 (see Chapter 17.3)
  - Operating T13 (see Chapter 17.4)
  - Trap handling (see Chapter 17.5)
  - Multichannel mode (see Chapter 17.6)
  - Hall sensor mode (see Chapter 17.7)
  - Interrupt handling (see Chapter 17.8)
  - General module operation (see Chapter 17.9)
- CCU6 kernel registers description (see Chapter 17.10)
- MOTIX™ TLE984xQX implementation specific details (see Chapter 17.11)

17.1 Feature set overview

This section gives an overview over the different building blocks and their main features.

Timer 12 **block features**

- Three capture/compare channels, each channel can be used either as capture or as compare channel
- Generation of a three-phase PWM supported (six outputs, individual 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 T12 registers
- Center-aligned and edge-aligned PWM can be generated
- Single-shot mode supported
- Start can be controlled by external events
- Capability of counting external events
- Multiple interrupt request sources
- Hysteresis-like control mode

Timer 13 **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 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 supported for position input signals

- 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 multichannel AC-drives
- Output levels can be selected and adapted to the power stage

17.2 Introduction

The CCU6 unit is made up of a timer T12 block with three capture/compare channels and a timer T13 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 AC 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 means for efficient software control.

Note: The capture/compare module itself is named CCU6 (capture/compare unit 6). A capture/compare channel inside this module is named CC6x.

17.2.1 Block diagram

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 in compare mode only. The multichannel control unit generates output patterns which can be modulated by T12 and/or T13. The modulation sources can be selected and combined for the signal modulation.

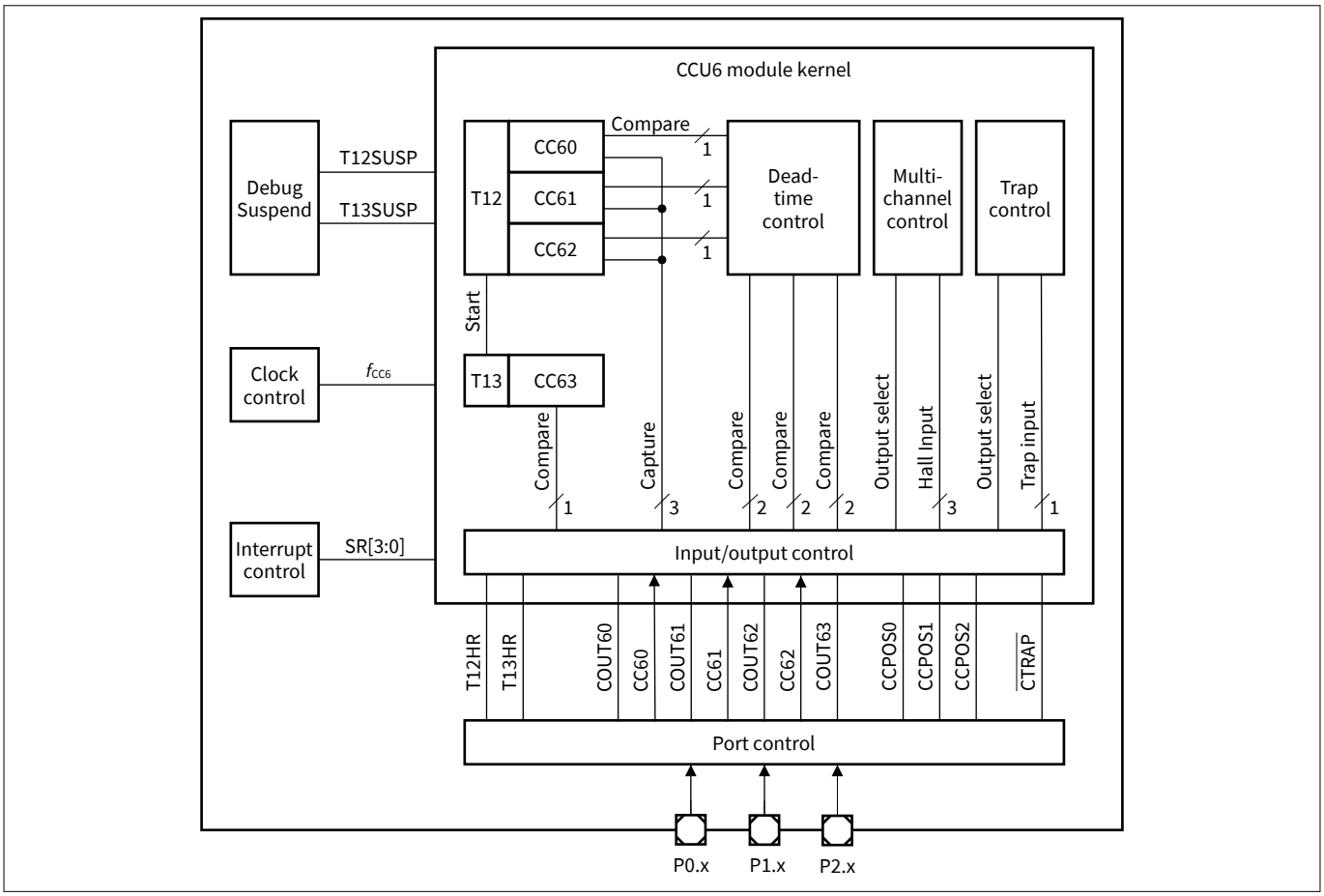


Figure 119 CCU6 block diagram

17.3 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, which 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 Chapter 17.3.1)
- Counting scheme (see Chapter 17.3.2)
- Compare modes (see Chapter 17.3.3)
- Compare mode output path (see Chapter 17.3.4)
- Capture modes (see Chapter 17.3.5)
- Shadow transfer (see Chapter 17.3.6)
- T12 operating mode selection (see Chapter 17.3.7)

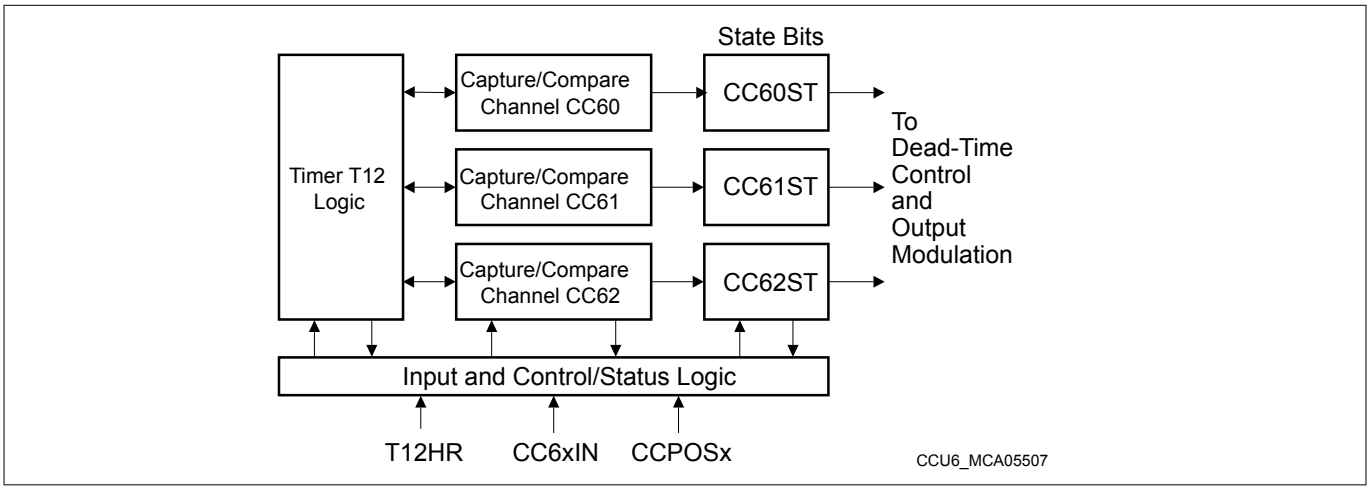


Figure 120 Overview diagram of the timer T12 block

17.3.1 T12 overview

Figure 121 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 108). T12 can count up or down, depending on the selected operation mode. A direction flag, CDIR, indicates the current counting direction.

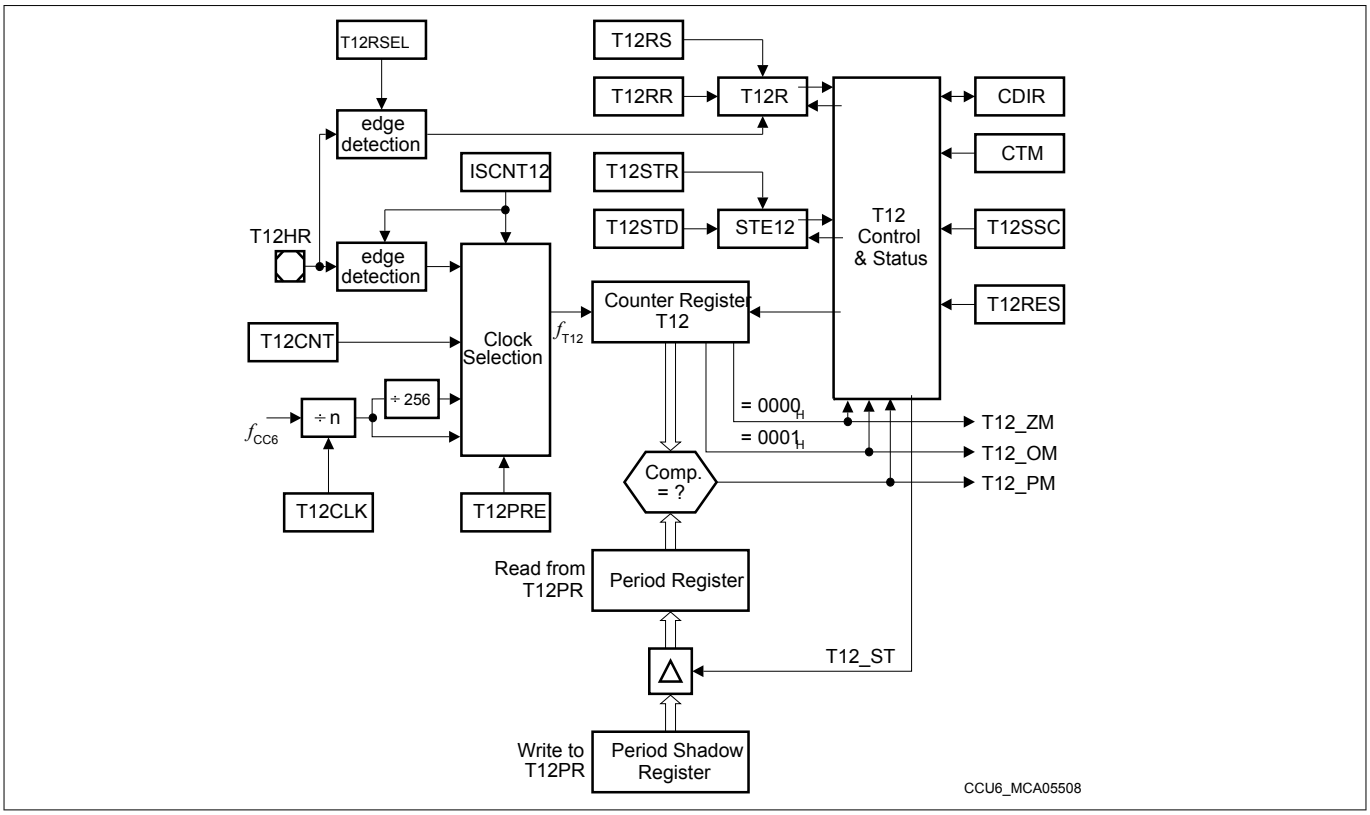


Figure 121 Timer T12 logic and period comparators

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 Chapter 17.3.6) 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 122) or Center-Aligned mode (Figure 123), 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 124 and Figure 125).

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.

17.3.2 T12 counting scheme

This section describes the clocking and counting capabilities of T12.

17.3.2.1 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 108. The prescaler of T12 is cleared while T12 is not running (TCTR0.T12R = 0) to ensure reproducible timings and delays.

Table 108 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

17.3.2.2 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).

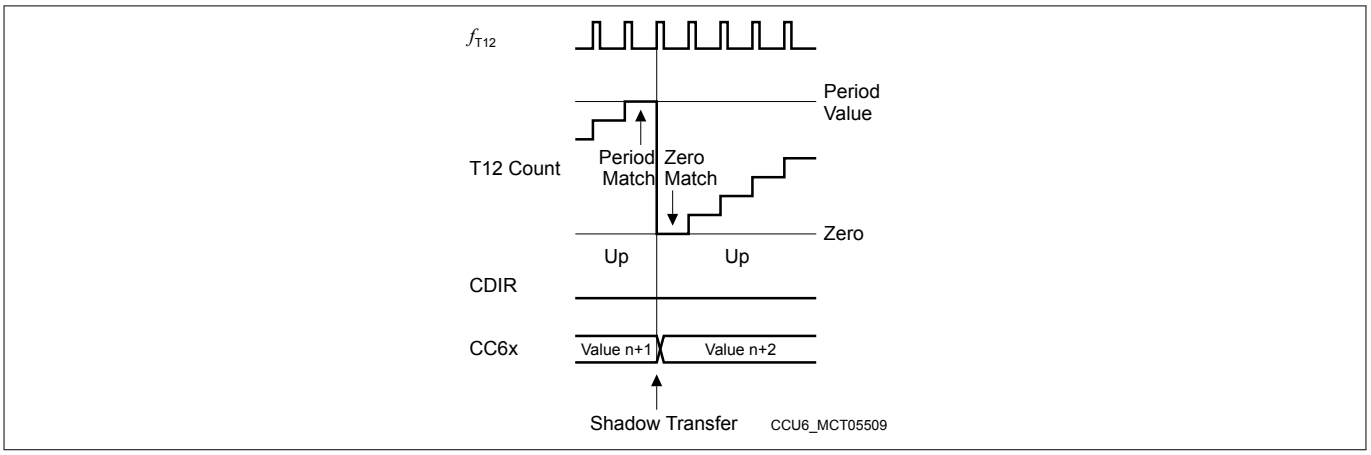


Figure 122 T12 operation in edge-aligned mode

As a result, in edge-aligned mode, the timer period is given by:

$$T12_{\mathrm{PER}} = \text{<Period-Value>} + 1;\ \text{in}\ T12\ \text{clocks}\ (f_{T12}) \tag{8}$$

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:

$$T12_{\mathrm{PER}} = (\text{<Period-Value>} + 1) \times 2;\ \text{in}\ T12\ \text{clocks}\ (f_{T12}) \tag{9}$$

- 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.

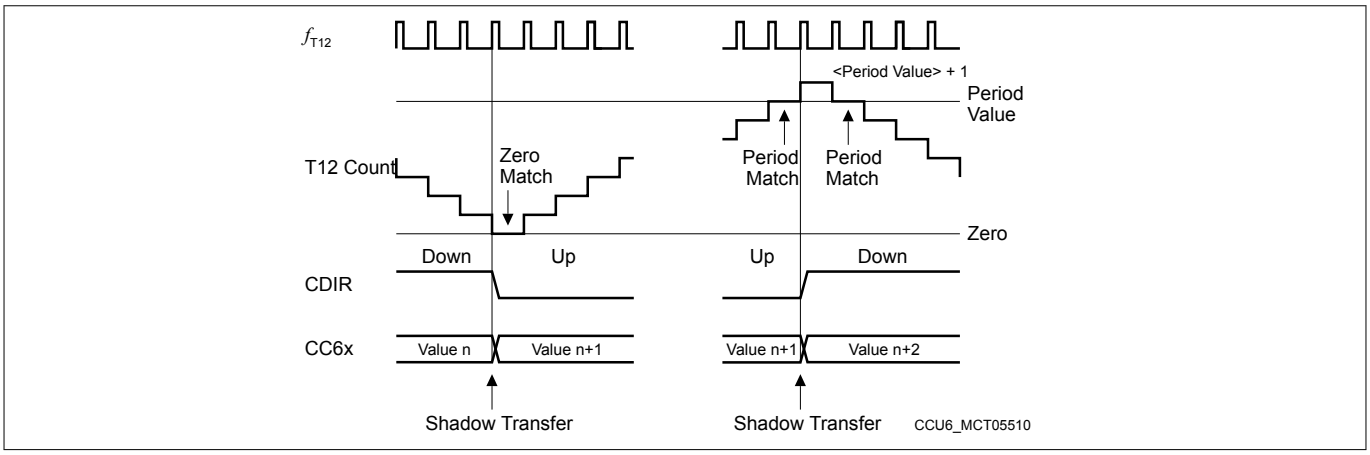


Figure 123 T12 operation in center-aligned mode

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 123).

17.3.2.3 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 124).

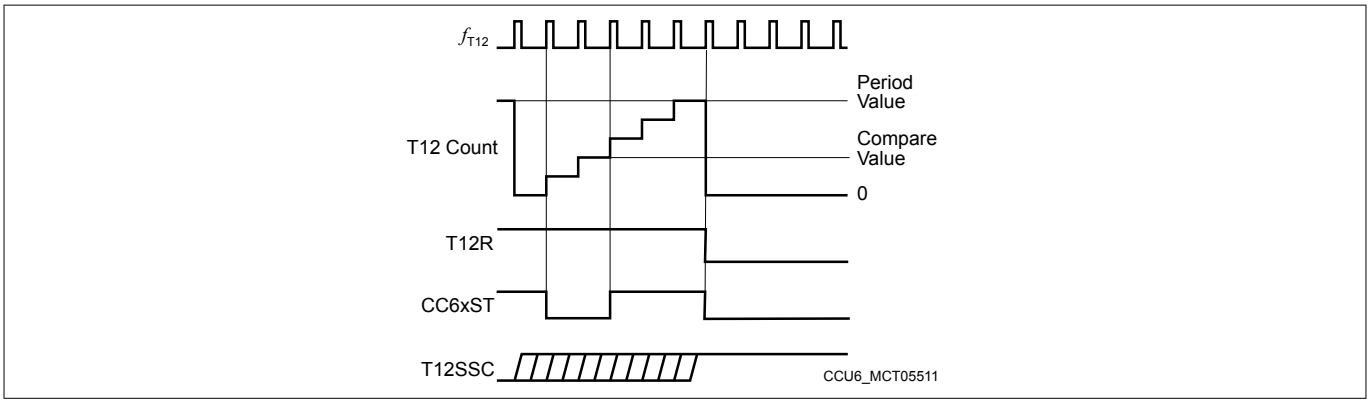


Figure 124 Single-shot operation in edge-aligned mode

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 125).

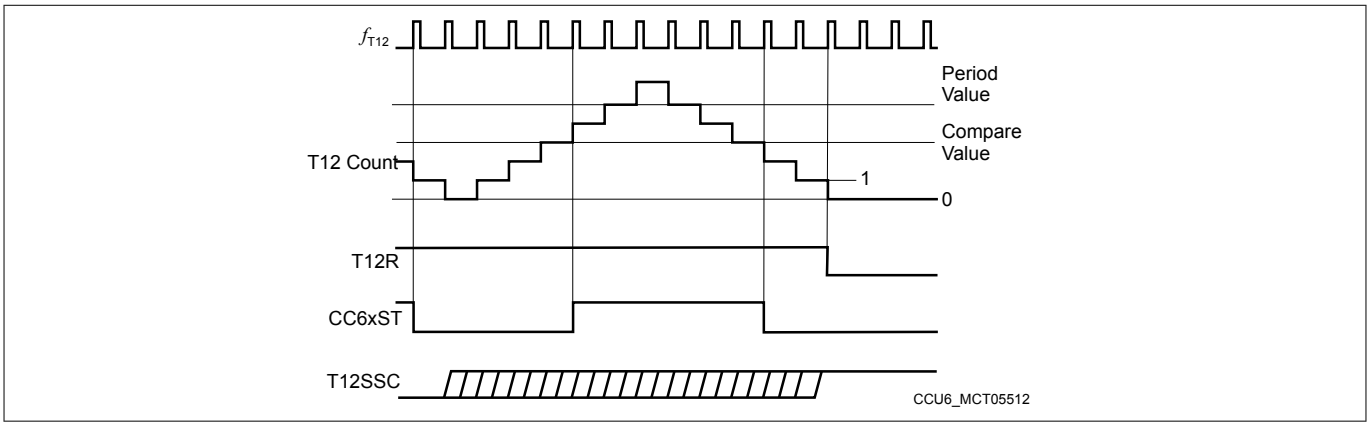


Figure 125 Single-shot operation in center-aligned mode

17.3.3 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 Chapter 17.3.5.

17.3.3.1 Compare channels

In compare mode (see Figure 126), the three individual compare channels CC60 CC61, and CC62 can generate a three-phase PWM pattern.

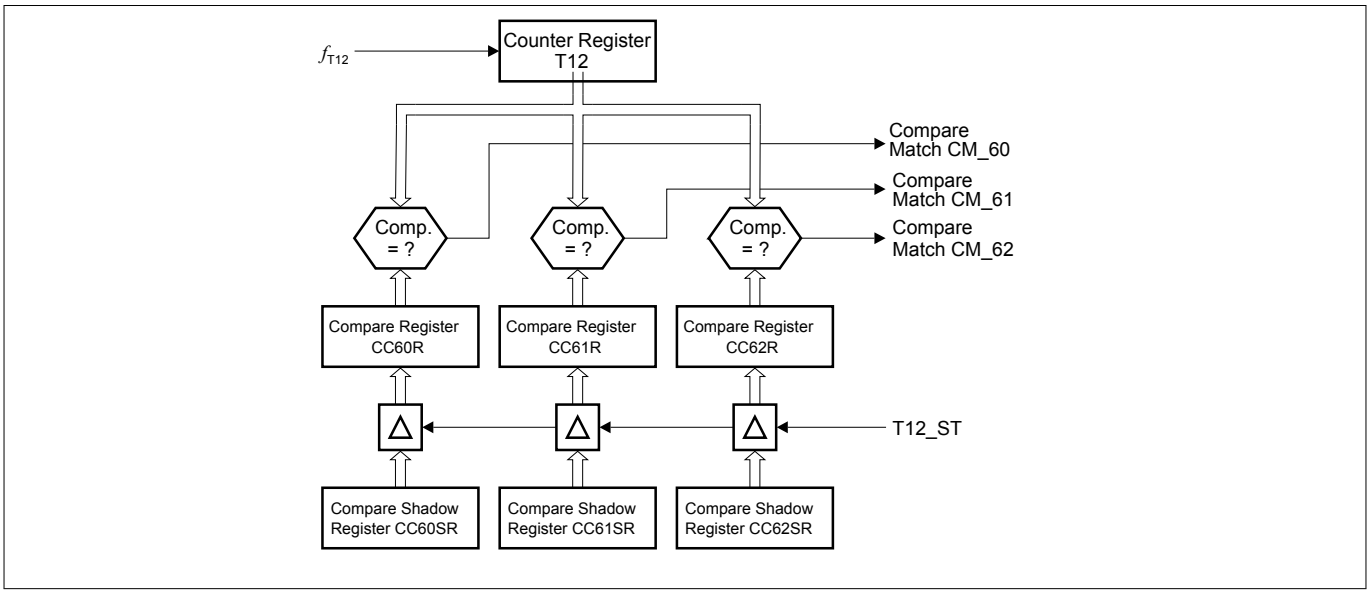


Figure 126 T12 channel comparators

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.

17.3.3.2 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 127). In compare mode, the state bits are modified according to a set of switching rules, depending on the current status of timer T12.

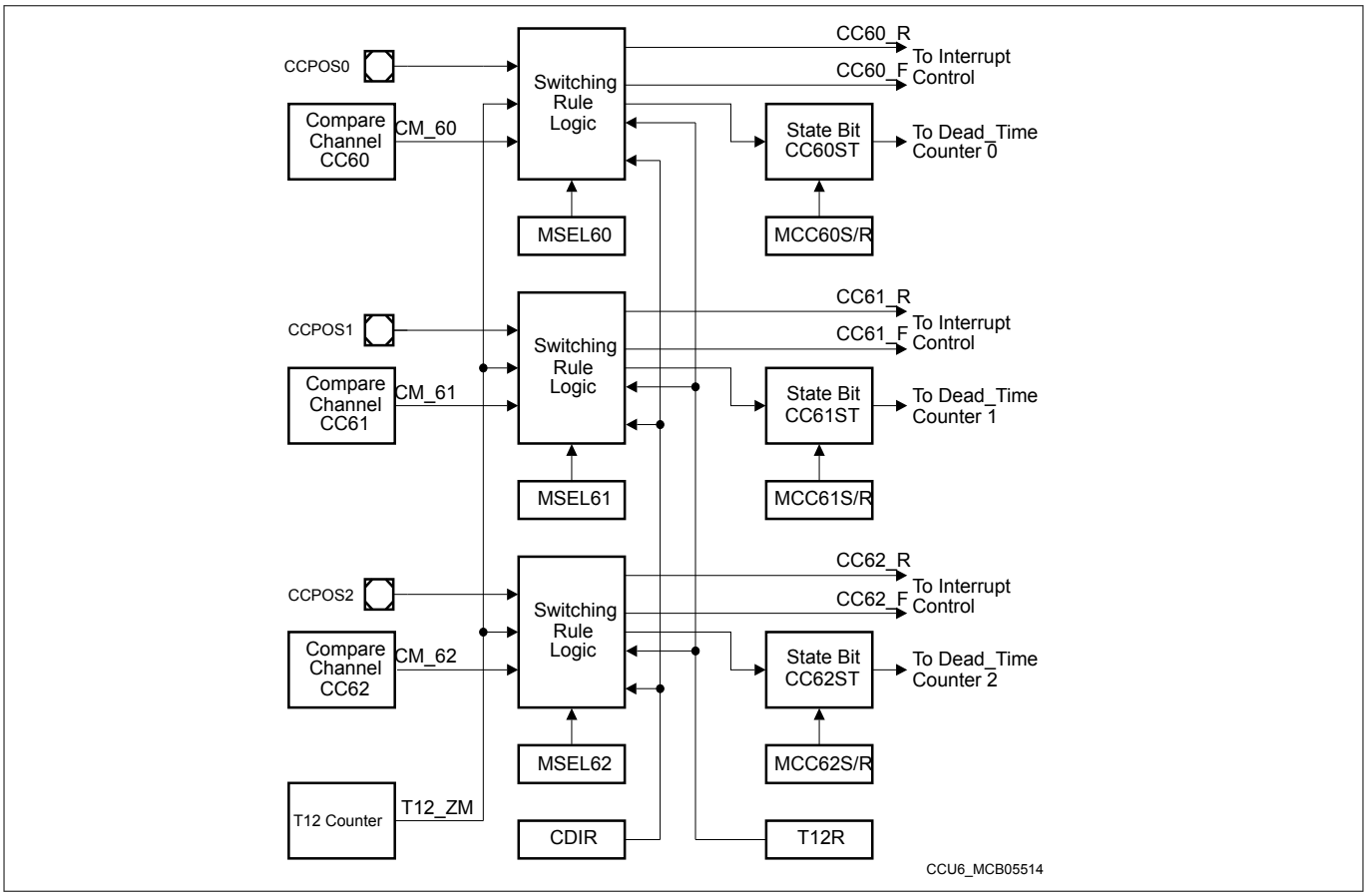


Figure 127 Compare state bits for compare mode

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 128 and Figure 129):

A state bit CC6xST is set to 1:

- with the next T12 clock (f_{T12}) after a compare-match when T12 is counting up (that is, 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 (that is, 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

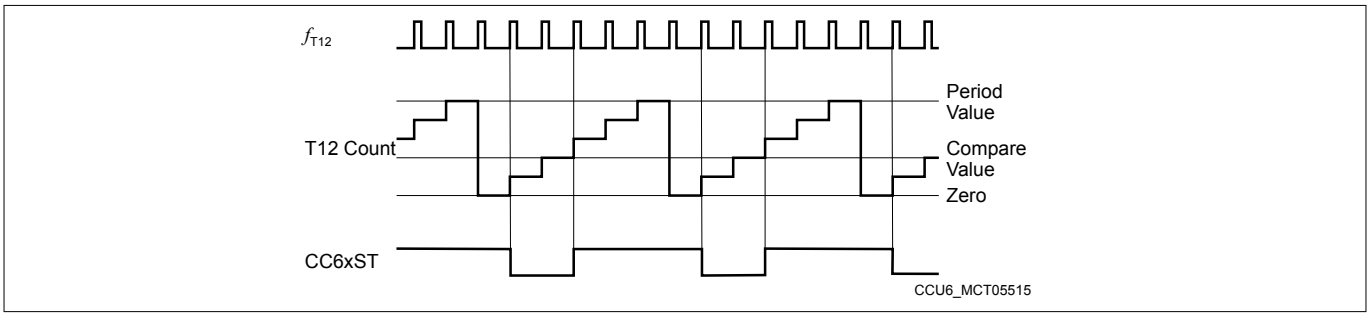


Figure 128 Compare operation, edge-aligned mode

Figure 130 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.

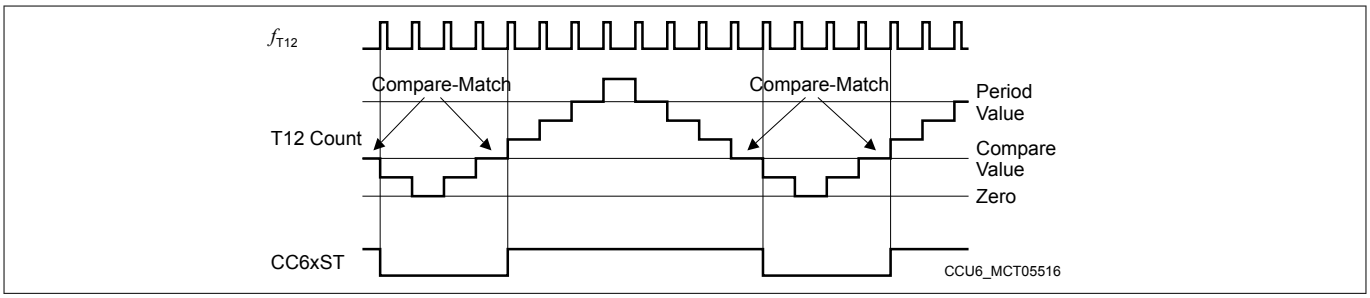


Figure 129 Compare operation, center-aligned mode

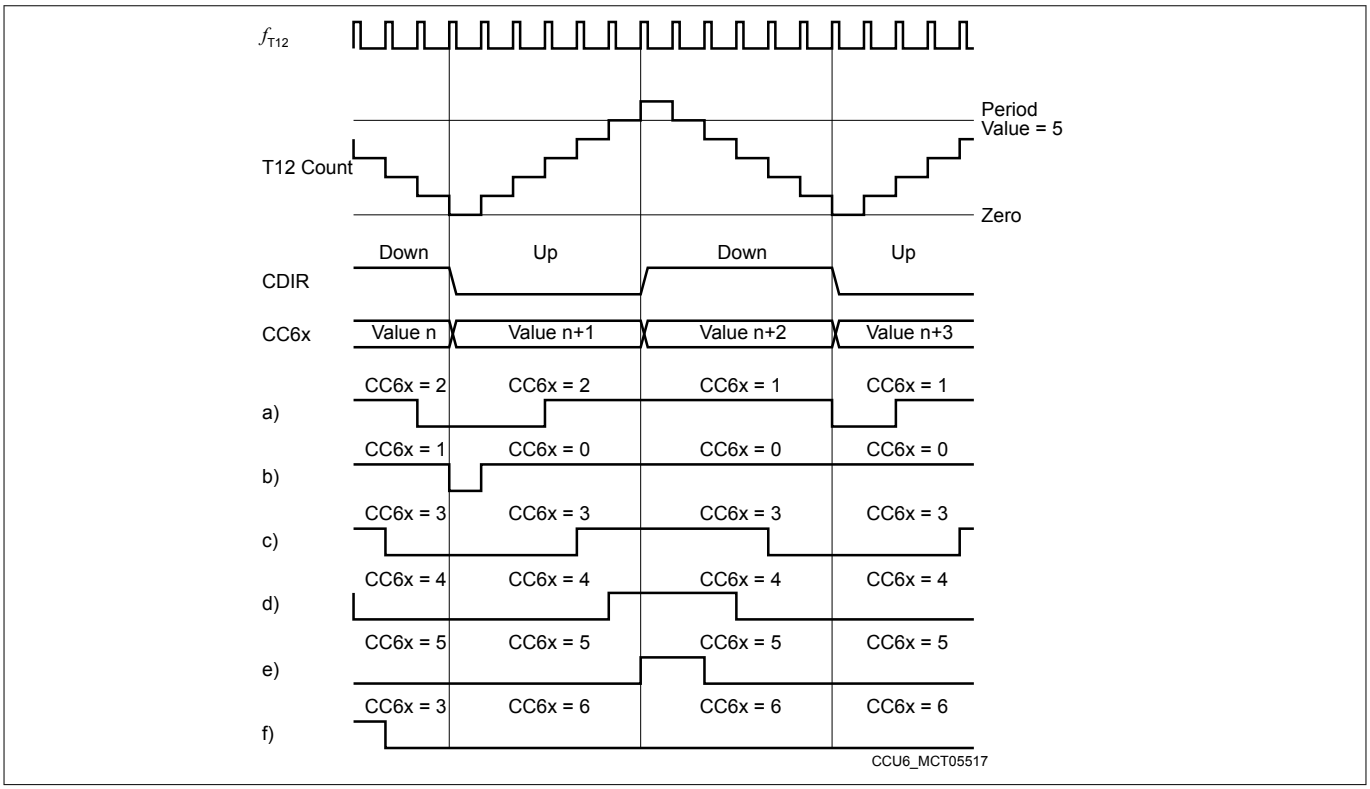


Figure 130 Compare waveform examples

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 131 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.

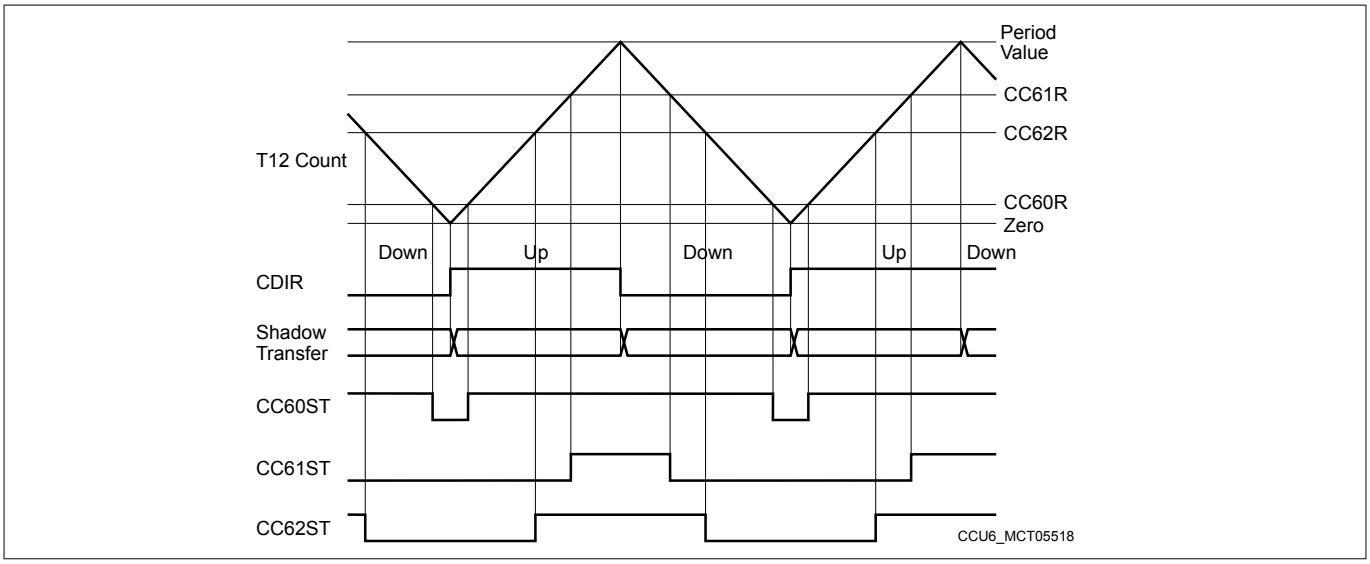


Figure 131 Three-channel compare waveforms

17.3.3.3 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 127 for the state bit logic and Figure 132 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.

17.3.4 Compare mode output path

Figure 132 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 Chapter 17.3.4.3 for details on the output modulation.

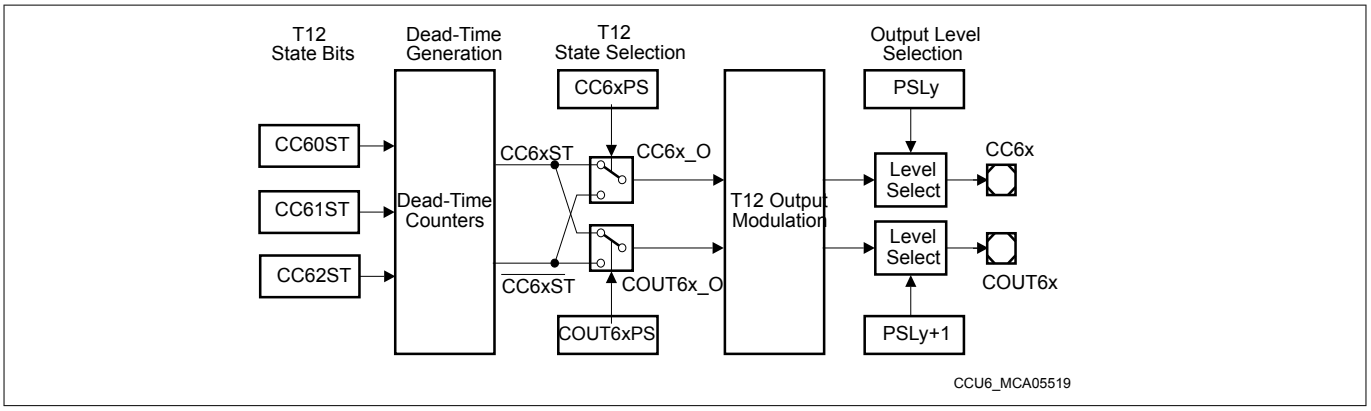


Figure 132 Compare mode simplified output path diagram

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 multichannel 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.

17.3.4.1 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 133, 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.

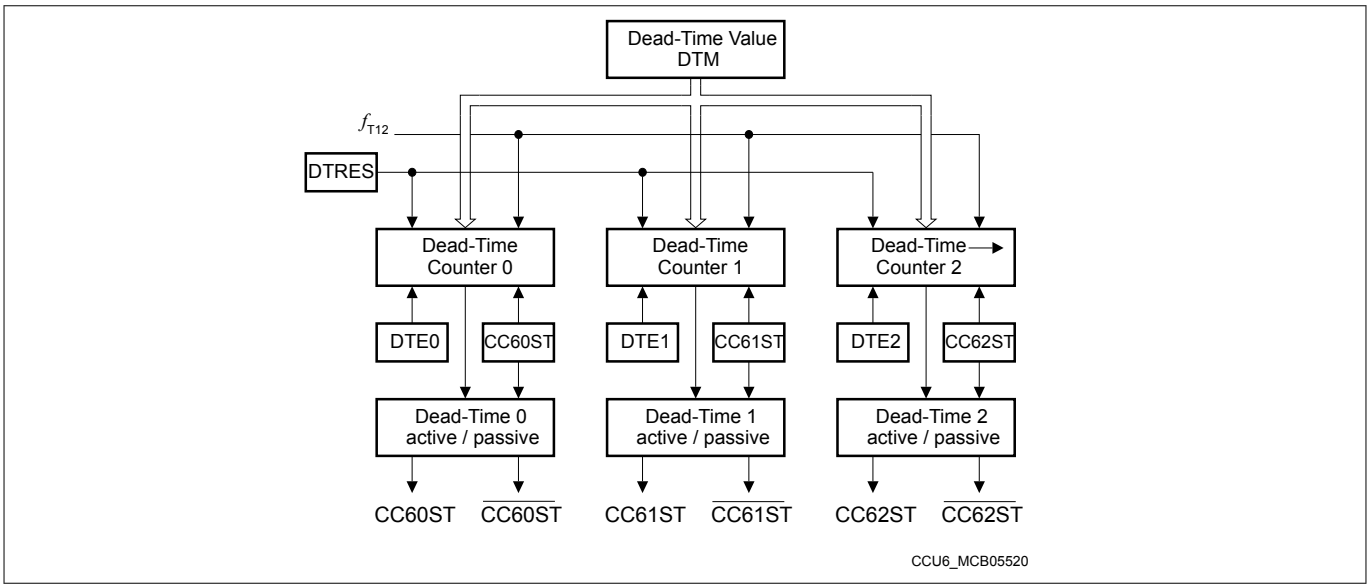


Figure 133 Dead-time generation block diagram

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 $\overline{\text{CC6xST}}$ for each compare channel, each masked with the effect of the related dead-time counters (waveforms illustrated in Figure 134).

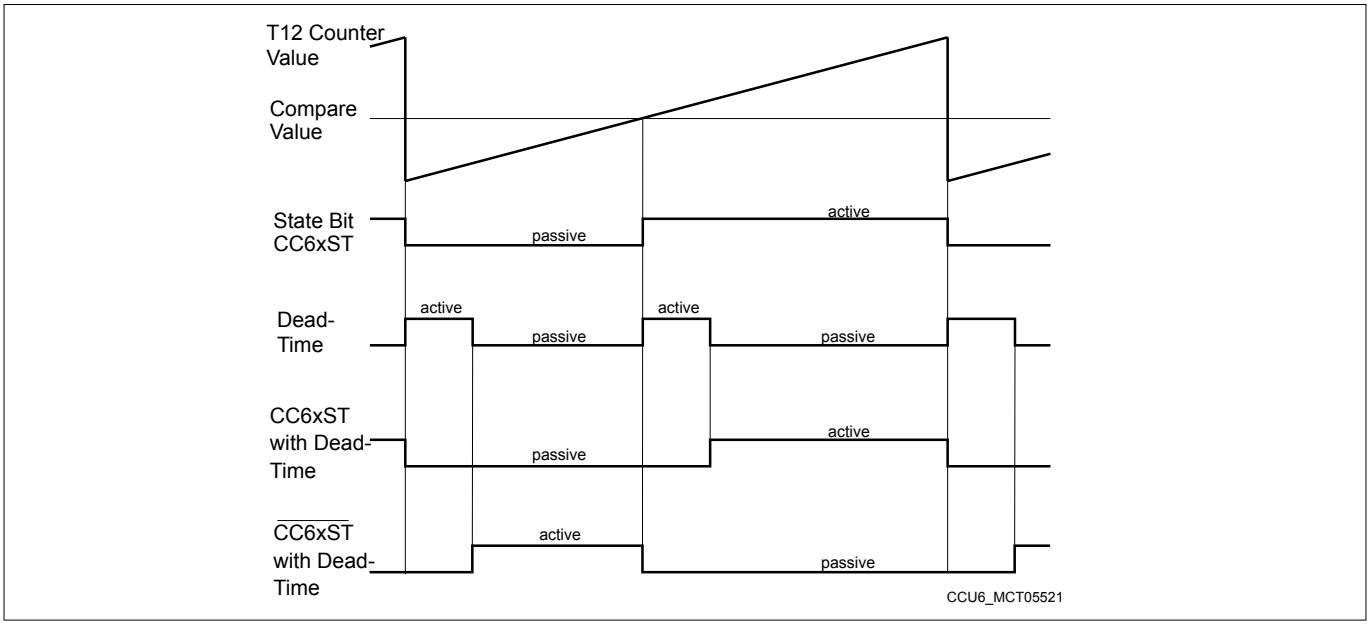


Figure 134 Dead-time generation waveforms

17.3.4.2 State selection

To support a wide range of power switches and drivers, the state selection offers the flexibility to define when 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 132). 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.

17.3.4.3 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 135 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 multichannel 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 135 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.

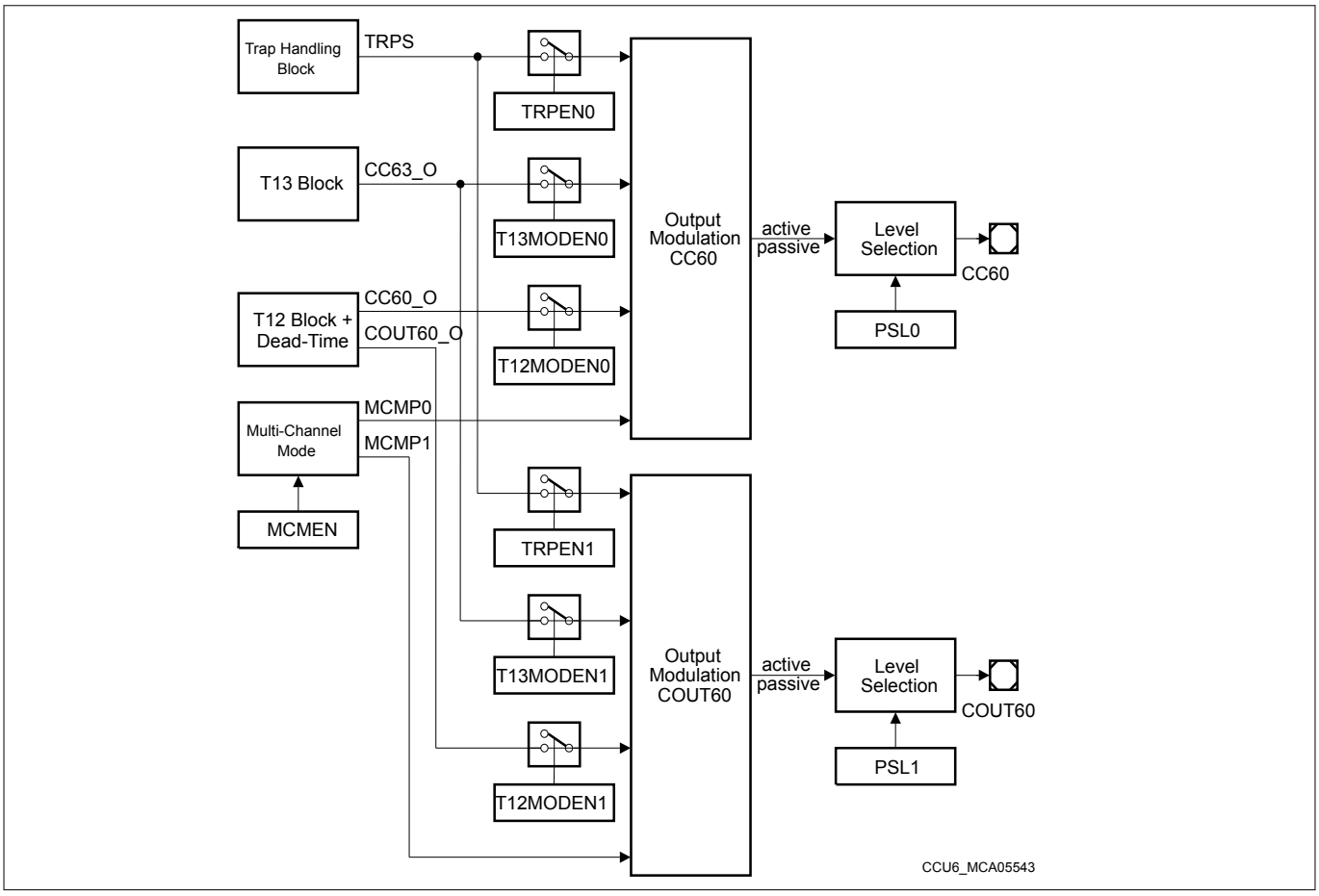


Figure 135 Output modulation for compare channel CC60

17.3.5 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 109 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 136 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.

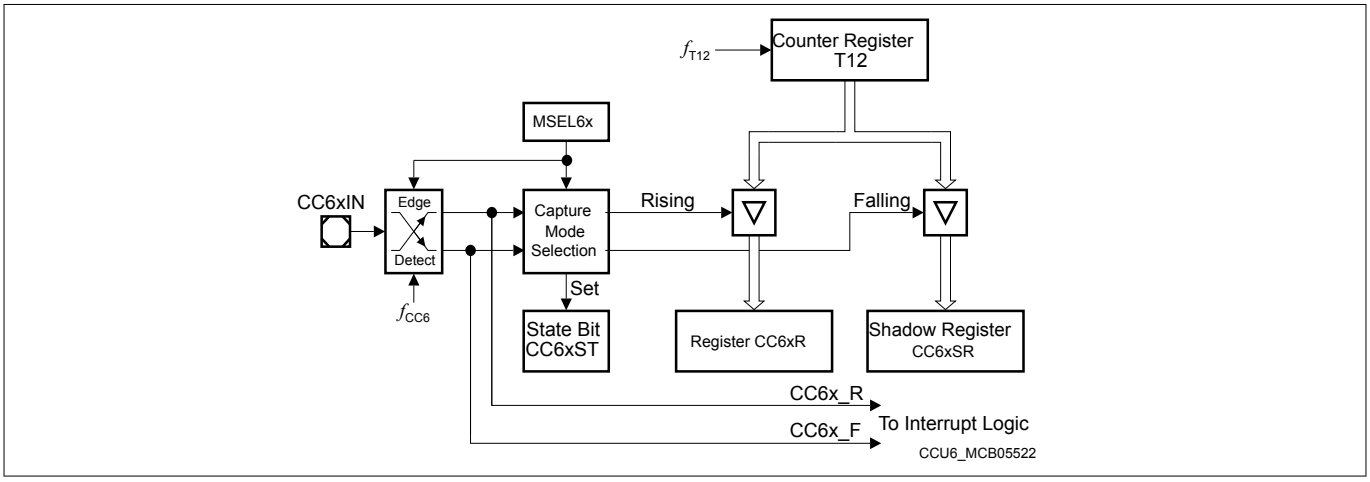


Figure 136 Capture mode 1 block diagram

Capture modes 2, 3 and 4 are shown in Figure 137. 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 109. These capture modes are very useful in cases where there is little time between two consecutive edges of the input signal.

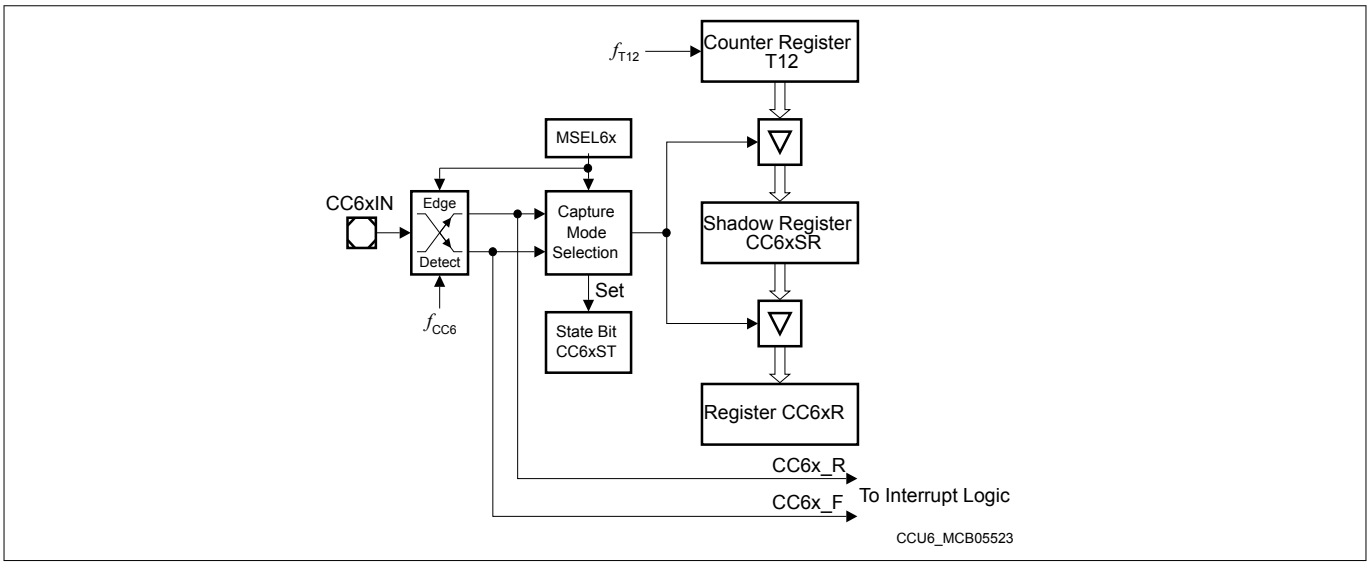


Figure 137 Capture modes 2, 3 and 4 block diagram

Five further capture modes are called multi-input capture modes, as they use two different external inputs, signal CC6xIN and signal CCPOSx.

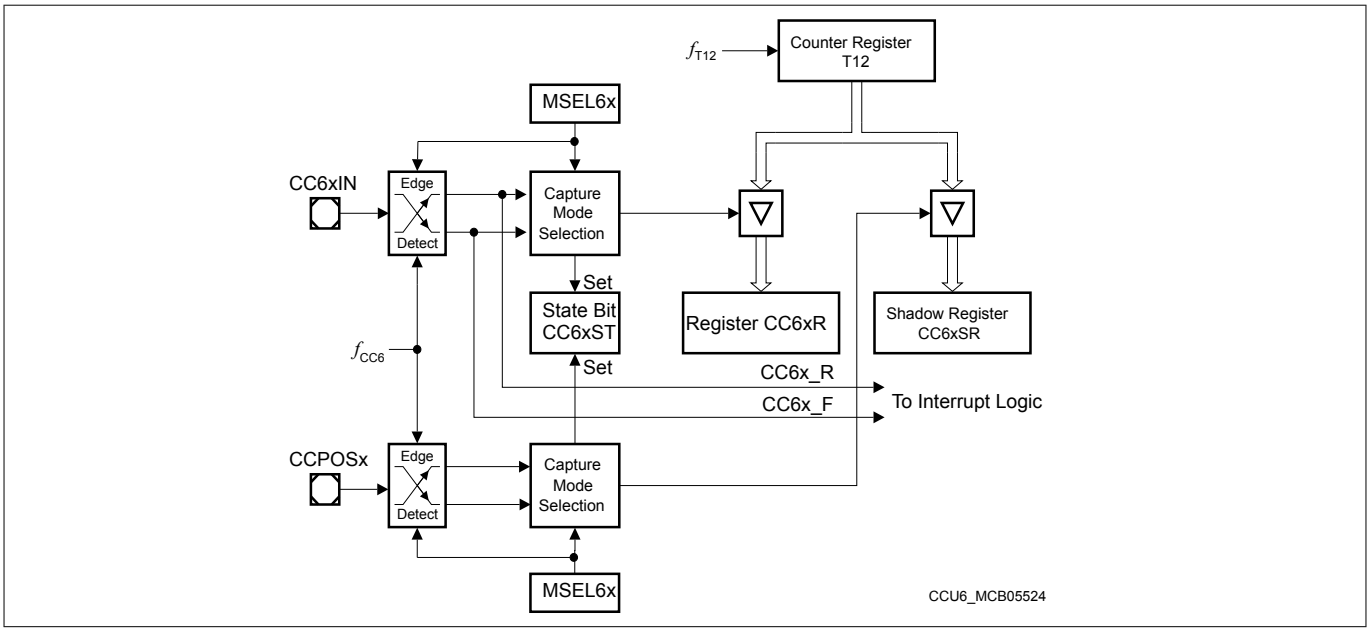


Figure 138 Multi-input capture modes block diagram

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 110.

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 Chapter 17.8).

Table 110 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)		

17.3.6 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).

The following figure shows the shadow register structure and the shadow transfer signals, as well as on the read/write accessibility of the various registers.

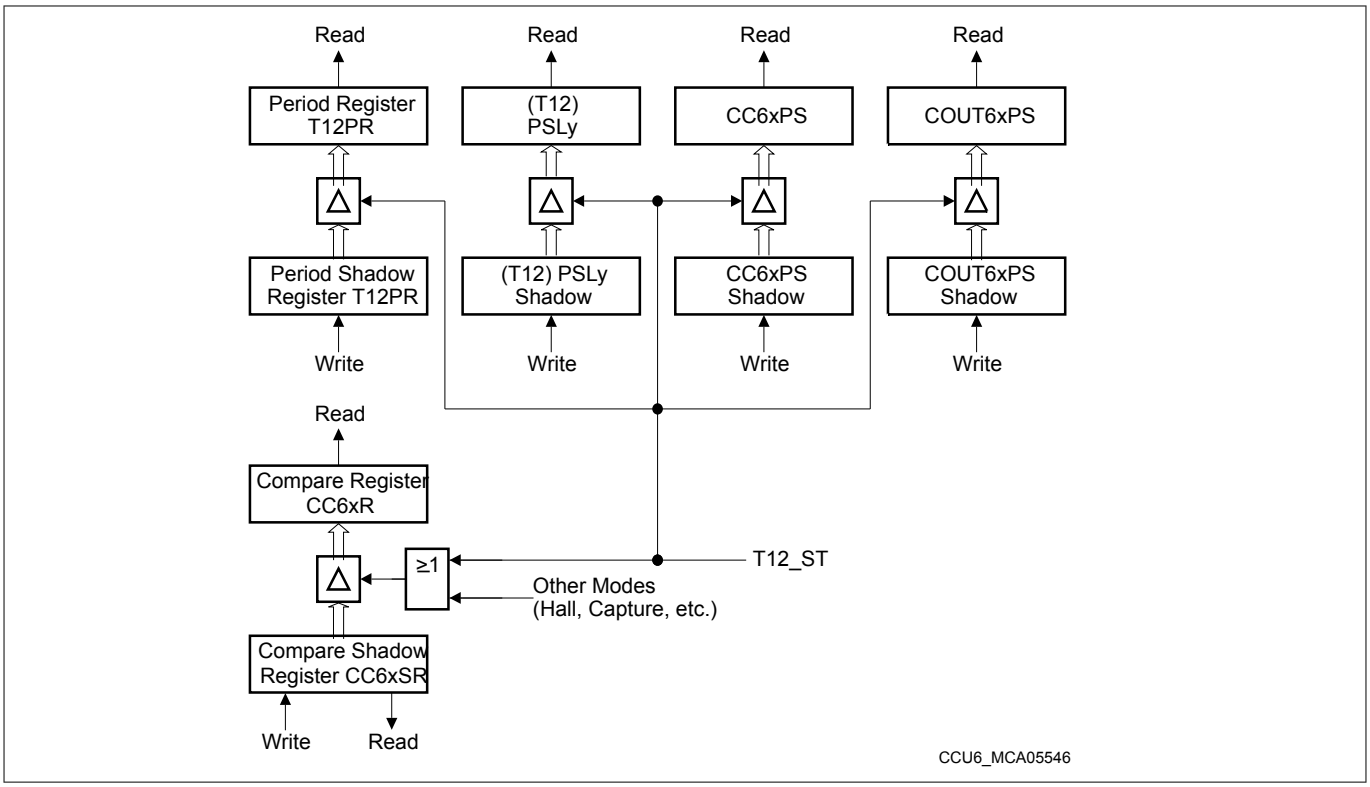


Figure 139 T12 shadow register overview

A T12 shadow register transfer takes place (T12_ST active):

- while timer T12 is not running (T12R = 0), or
- 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.

17.3.7 Timer T12 operating mode selection

The operating mode for the T12 channels are defined by the bit fields T12MSEL.MSEL6x.

Table 111 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 Chapter 17.3.3 Same behavior for all three codings
$01XX_B$	Double-register capture modes, see Chapter 17.3.5
1000_B	Hall sensor mode, see Chapter 17.7 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 Chapter 17.3.3.3
1010_B, 1011_B, 1100_B, 1101_B, 1110_B	Multi-input capture modes, see Chapter 17.3.5

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.

17.4 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 Chapter 17.4.1)
- Counting scheme (see Chapter 17.4.2)
- Compare mode (see Chapter 17.4.3)
- Compare output path (see Chapter 17.4.4)
- Shadow register transfer (see Chapter 17.4.5)

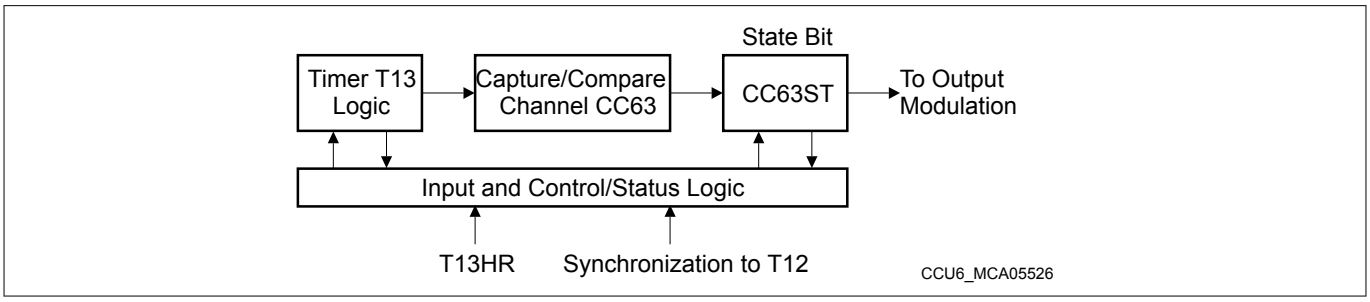


Figure 140 Overview diagram of the timer T13 block

17.4.1 T13 overview

Figure 141 shows a detailed block diagram of timer T13. The functions of the timer T12 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 T12).

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 Chapter 17.4.5). 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 143).

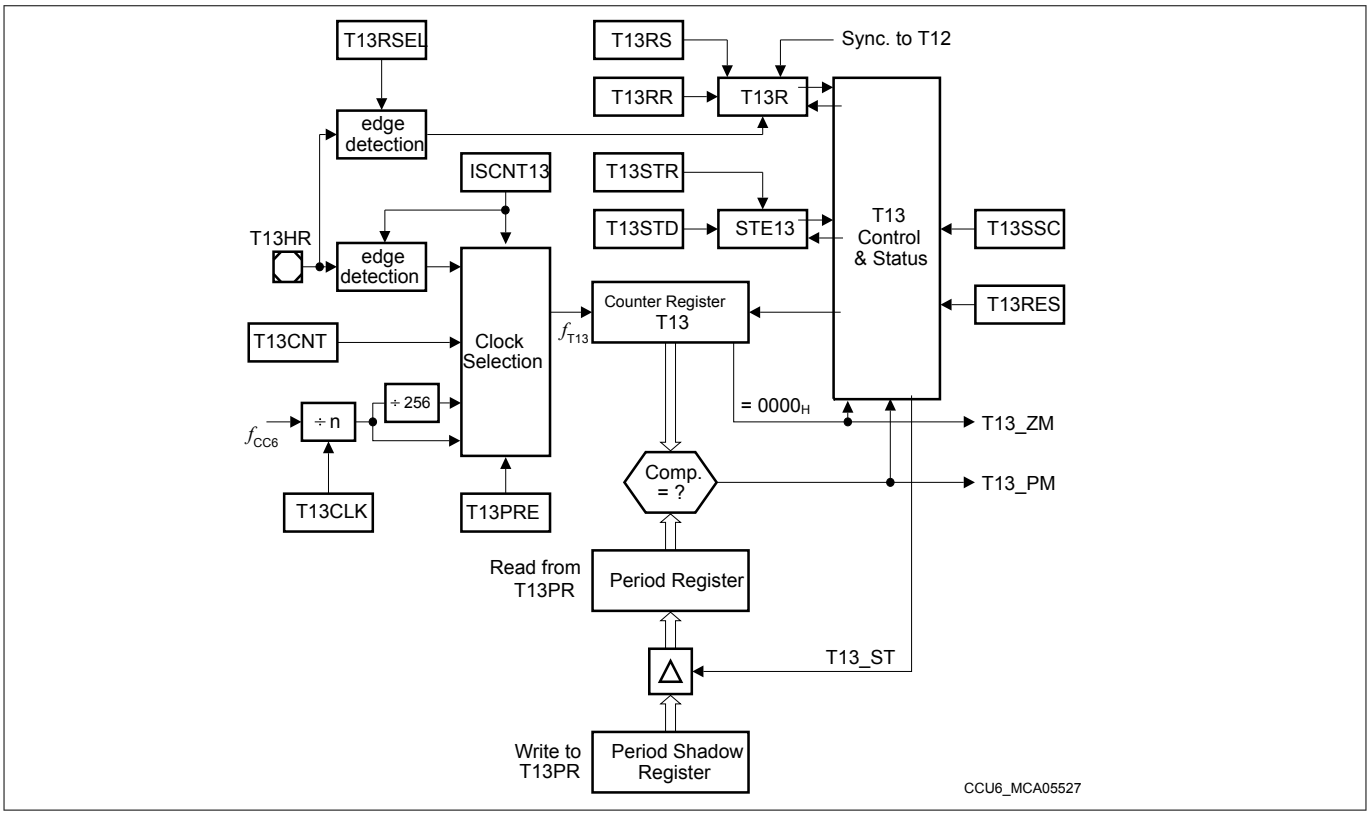


Figure 141 T13 counter logic and period comparators

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.

17.4.2 T13 counting scheme

This section describes the clocking and the counting capabilities of T13.

17.4.2.1 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 112. The prescaler of T13 is cleared while T13 is not running (TCTR0.T13R = 0) to ensure reproducible timings and delays.

Table 112 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

17.4.2.2 T13 counting

The period of the timer is determined by the value in the period register T13PR according to the following formula:

$$T13_{PER} = <\text{Period-Value}> + 1; \quad \text{in } T13 \text{ clocks } (f_{T13}) \tag{10}$$

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 the following figure.

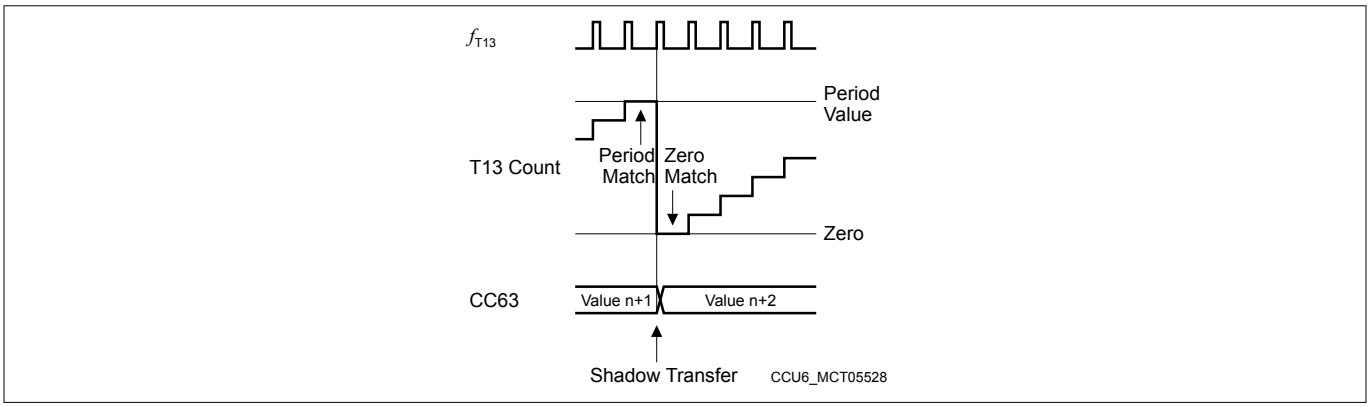


Figure 142 T13 counting sequence

17.4.2.3 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.

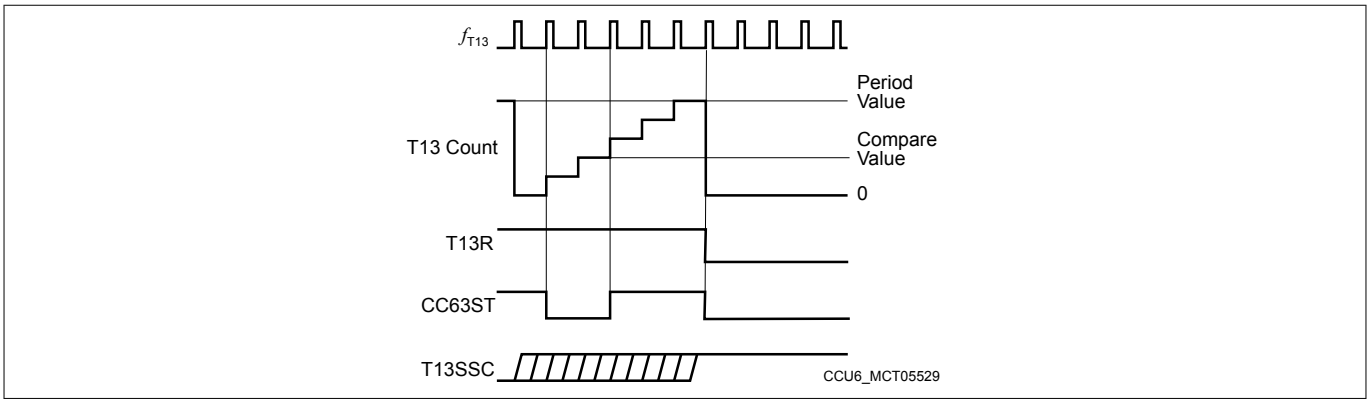


Figure 143 Single-shot operation of timer T13

17.4.2.4 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 144 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.

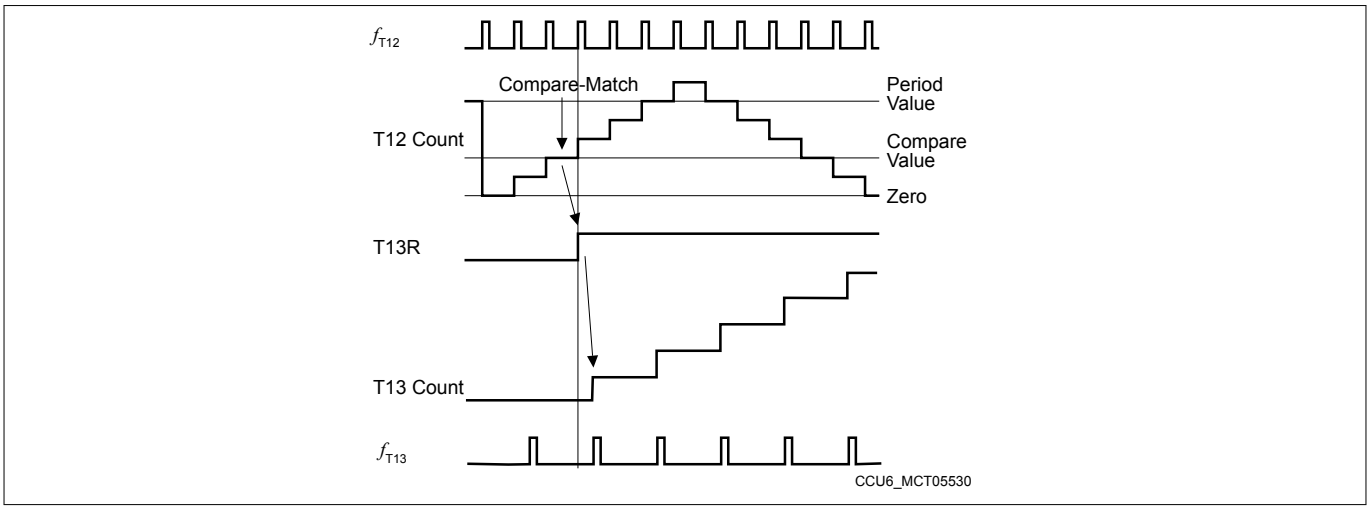


Figure 144 Synchronization of T13 to T12 compare match

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 113. Bit field T13TED additionally specifies for which count direction of T12 the selected trigger event should be regarded (see Table 114).

Table 113 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 114 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

17.4.3 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 140 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 145 gives an overview on the logic for the State Bit.

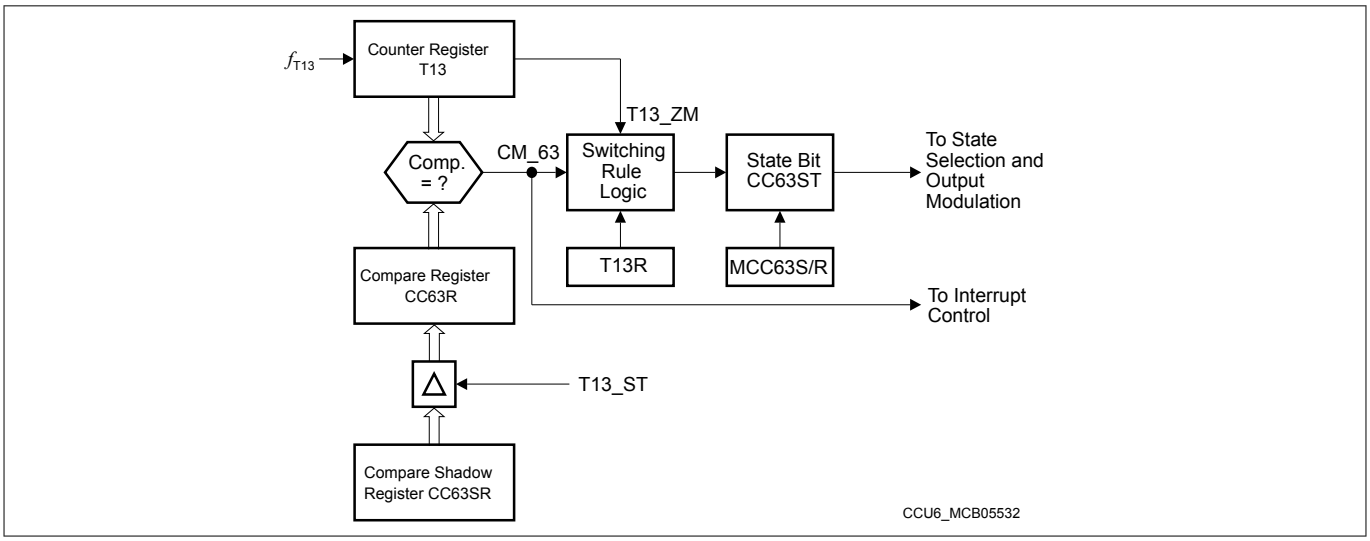


Figure 145 T13 state bit block diagram

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, that is, 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.

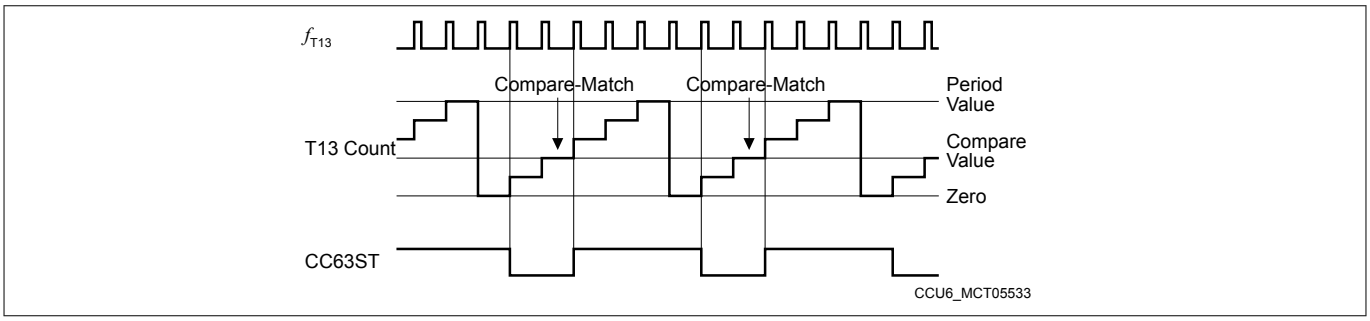


Figure 146 T13 compare operation

17.4.4 Compare mode output path

Figure 147 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 Chapter 17.3.4.3 for detailed information on the output modulation for T12 signals.

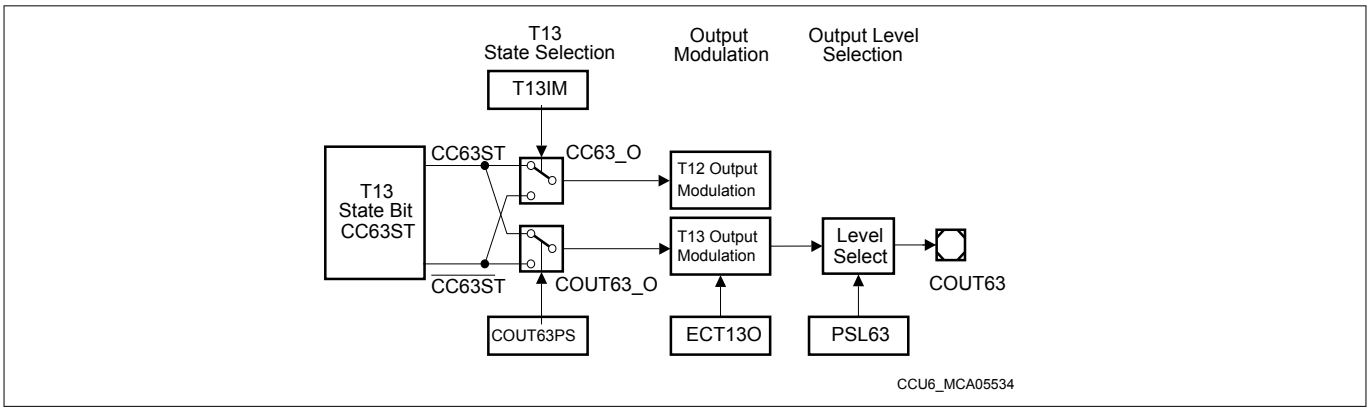


Figure 147 Channel 63 output path

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 148):

- 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.

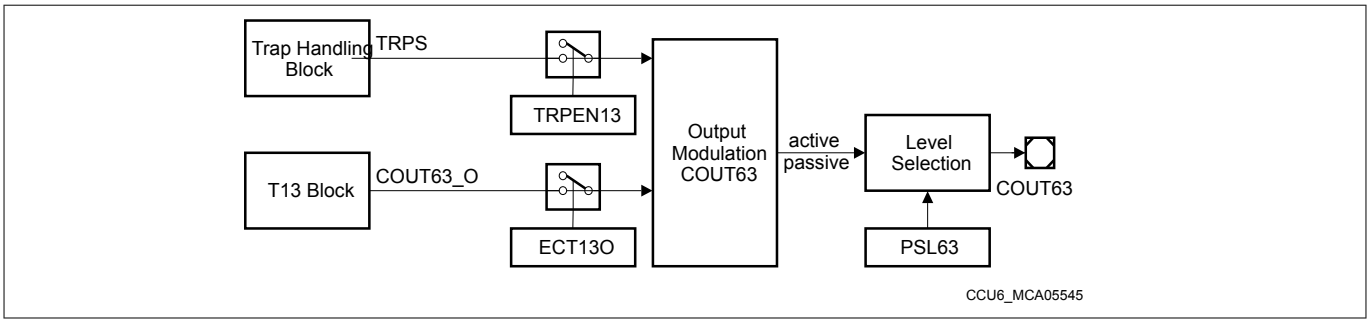


Figure 148 T13 output modulation

17.4.5 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

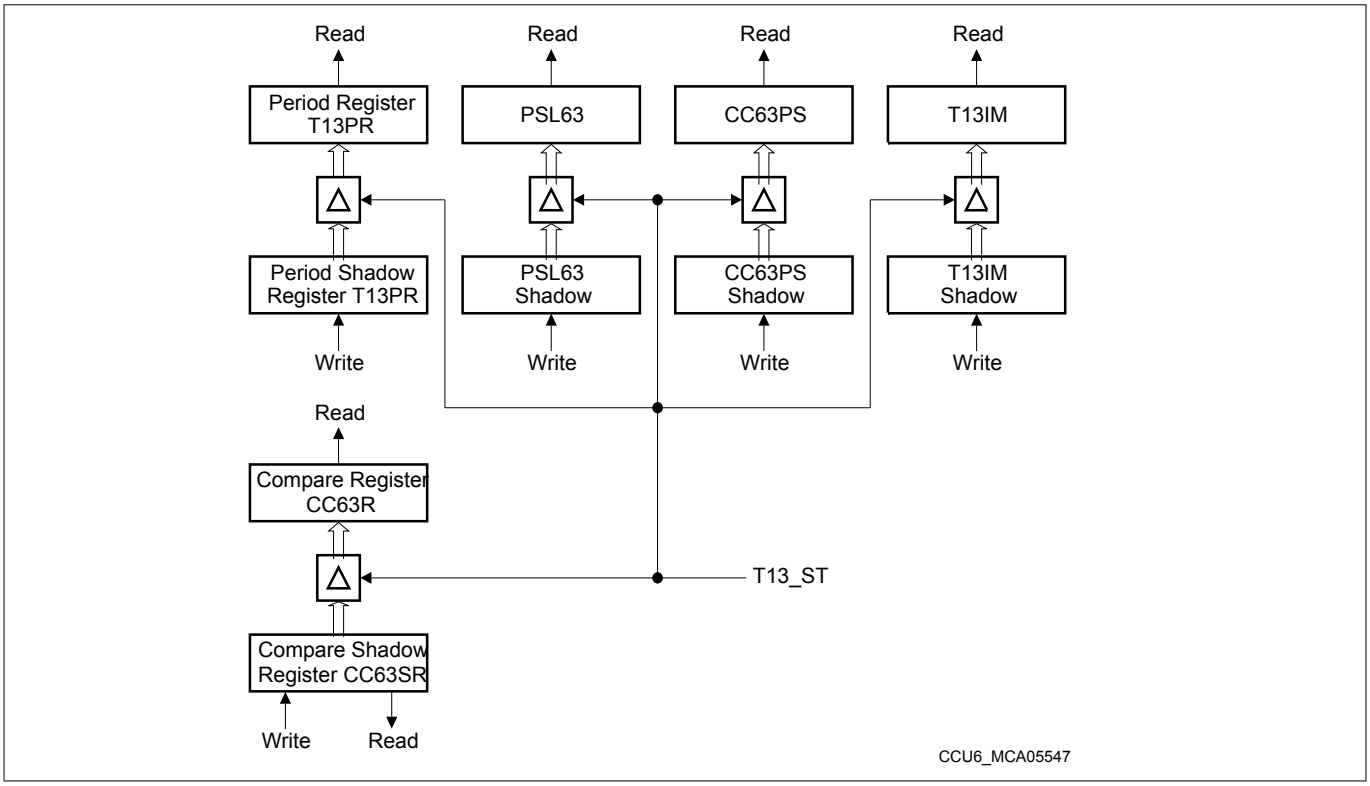


Figure 149 T13 shadow register overview

17.5 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 150 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.

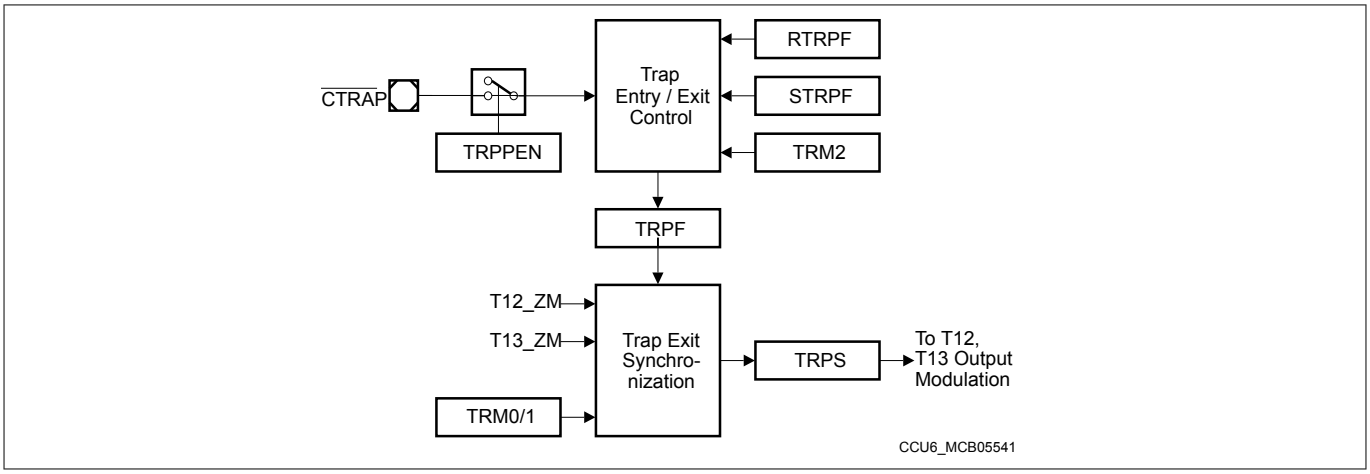


Figure 150 Trap logic block diagram

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 151 gives an overview on the associated operation.

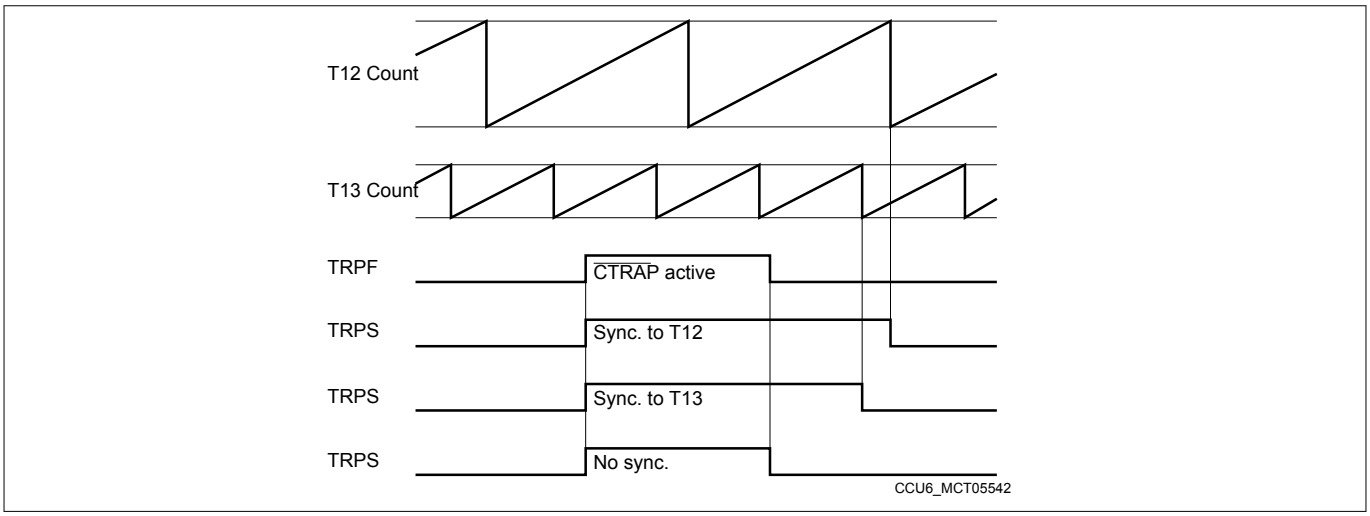


Figure 151 Trap state synchronization (with TRM2 = 0)

17.6 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.

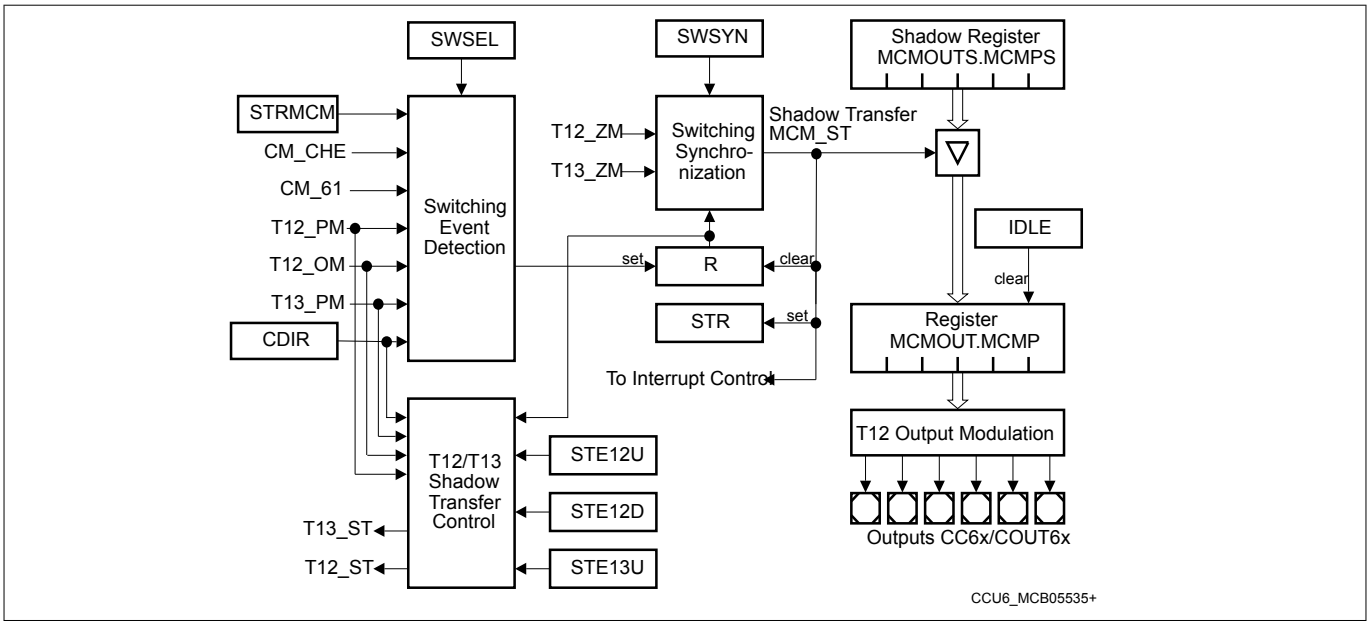


Figure 152 Multi-channel mode block diagram

Figure 152 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 115 Multi-channel mode switching event selection

SWSEL	Selected event (see register MCMCTR)
000_B	No automatic event detection

(table continues...)

Table 115 (continued) Multi-channel mode switching event selection

SWSEL	Selected event (see register MCMCTR)
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 116 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

17.7 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)

17.7.1 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.

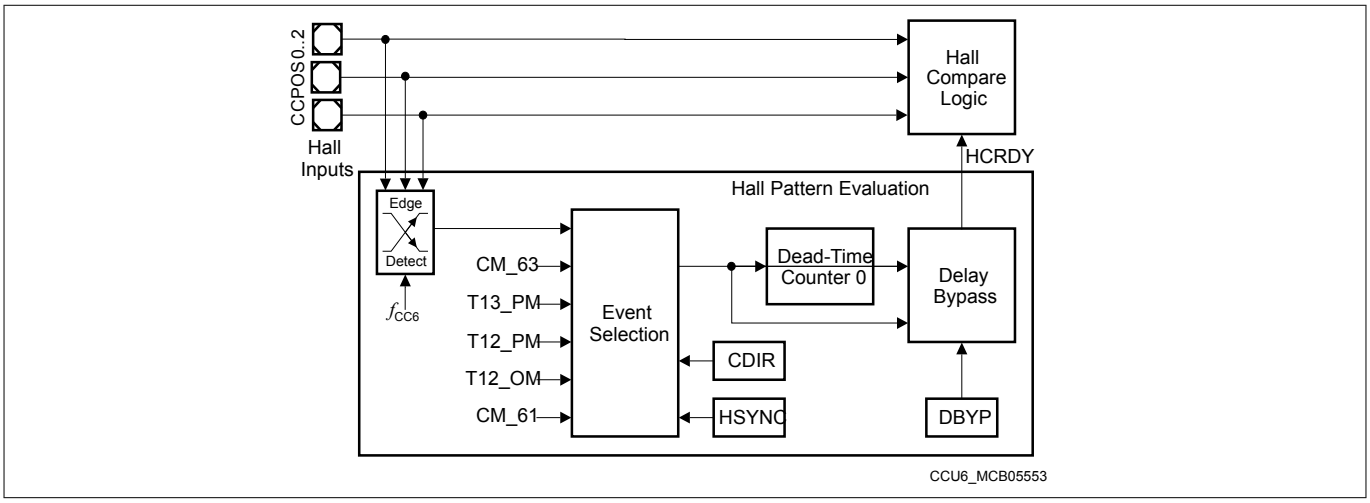


Figure 153 Hall pattern evaluation

If the evaluation signal HCRDY (Hall compare ready, see Figure 154) becomes activated, the Hall inputs are sampled and the Hall compare logic starts the evaluation of the Hall inputs.

Figure 153 illustrates the events for Hall pattern evaluation and the noise filter logic, Table 117 summarizes the selectable trigger input signals.

Table 117 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
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)

(table continues...)

Table 117 (continued) Hall sensor mode trigger event selection

HSYNC	Selected event (see register T12MSEL)
111_B	A T12 compare-match of compare channel CC61 while counting down (CM_61 and CDIR = 1)

17.7.2 Hall pattern compare logic

Figure 154 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.

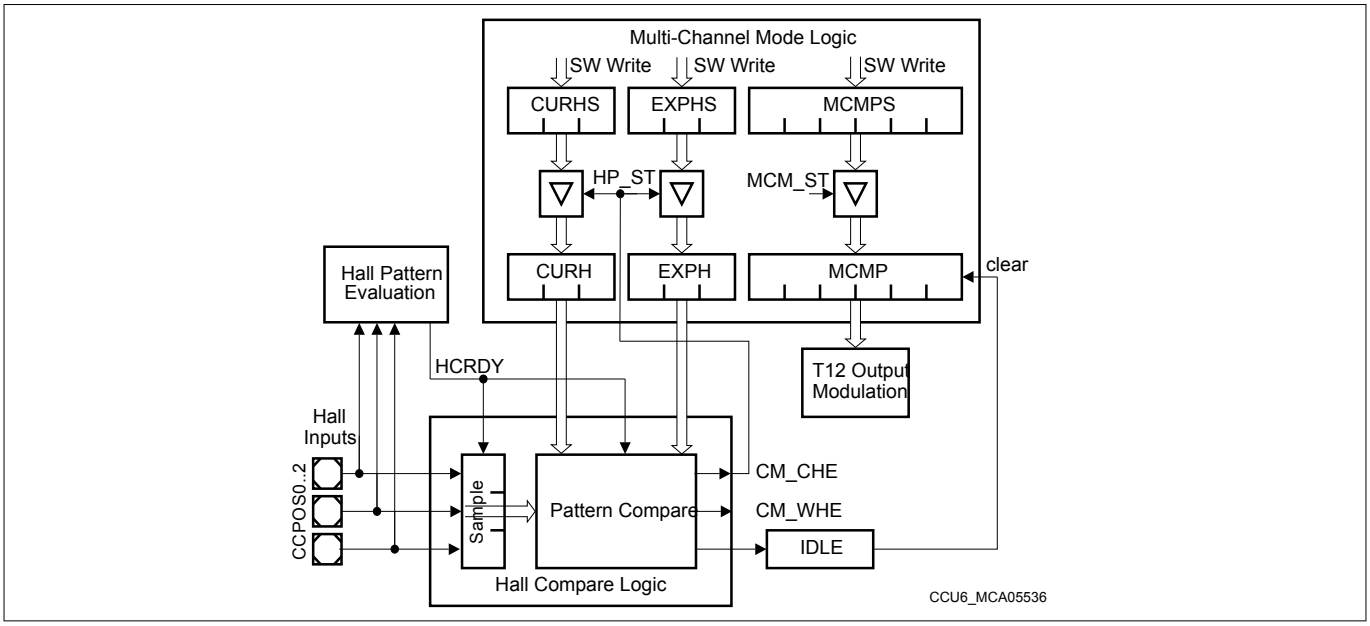


Figure 154 Hall pattern compare logic

- 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_CWE) 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).

17.7.3 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 152).

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.

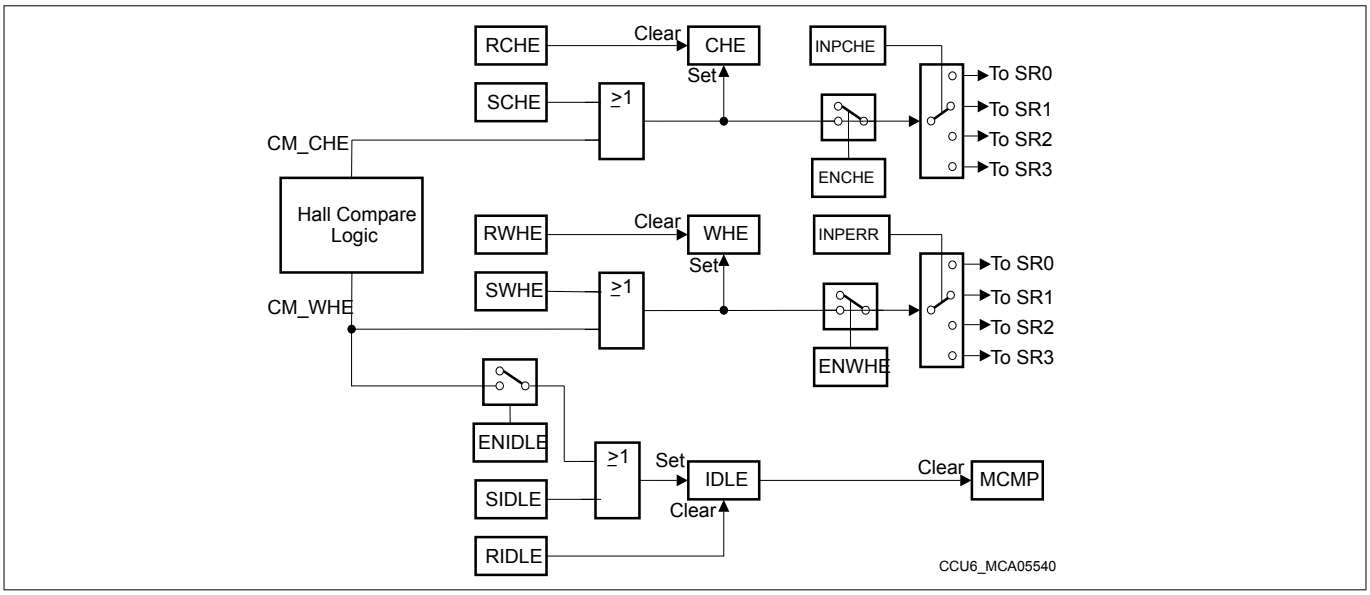


Figure 155 Hall mode flags

17.7.4 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 156, 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.

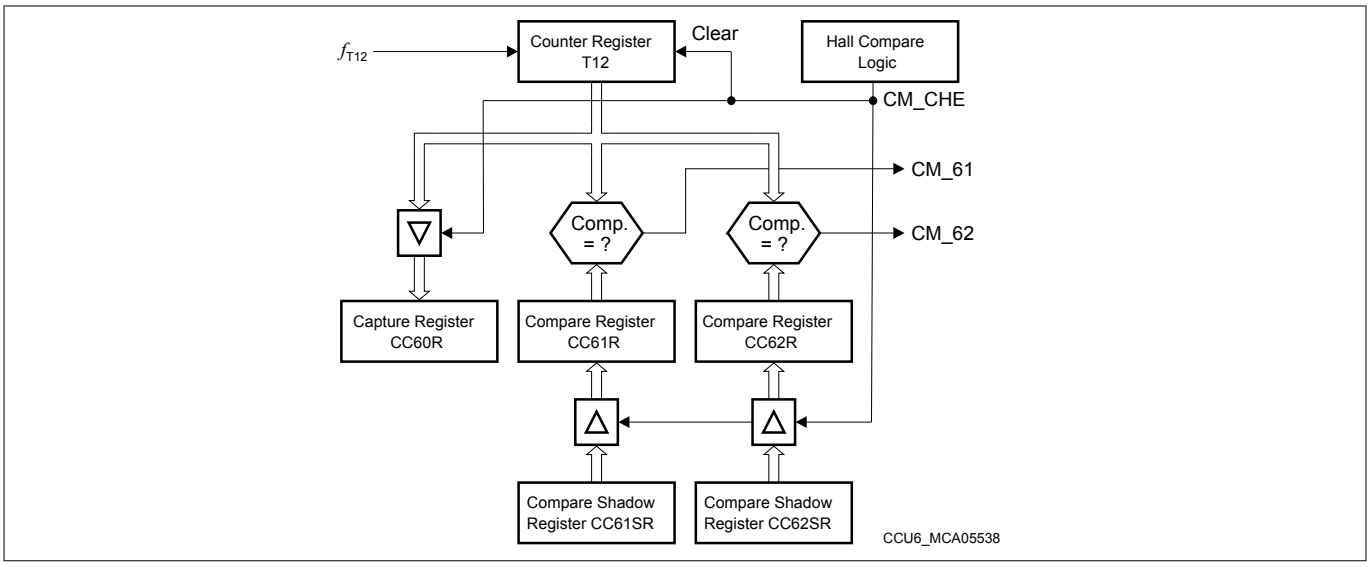


Figure 156 T12 block in Hall sensor mode

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.

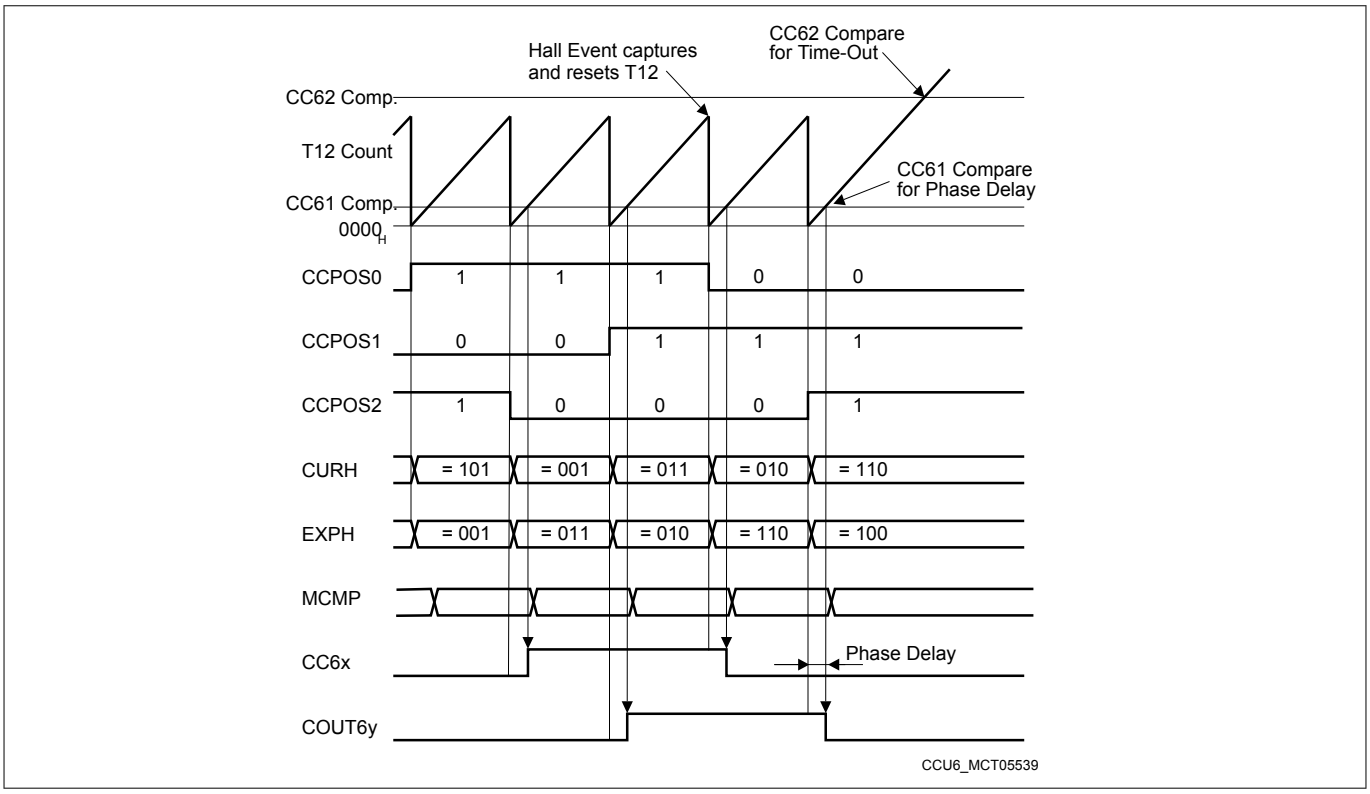


Figure 157 Brushless DC-motor control example (all MSEL6x = 1000_B)

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 Chapter 17.6). 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.

17.8 Interrupt handling

This section describes the interrupt handling of the CCU6 module.

17.8.1 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 158).

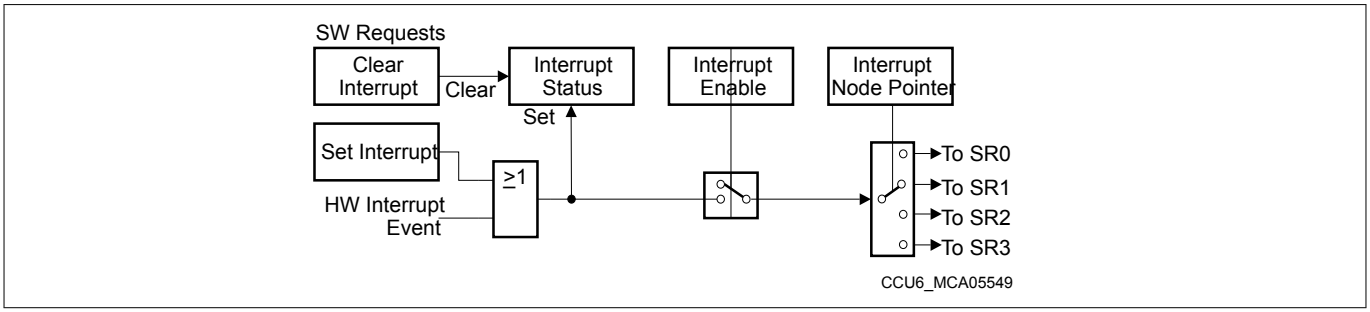


Figure 158 General interrupt structure

The available interrupt events in the CCU6 are shown in Figure 159.

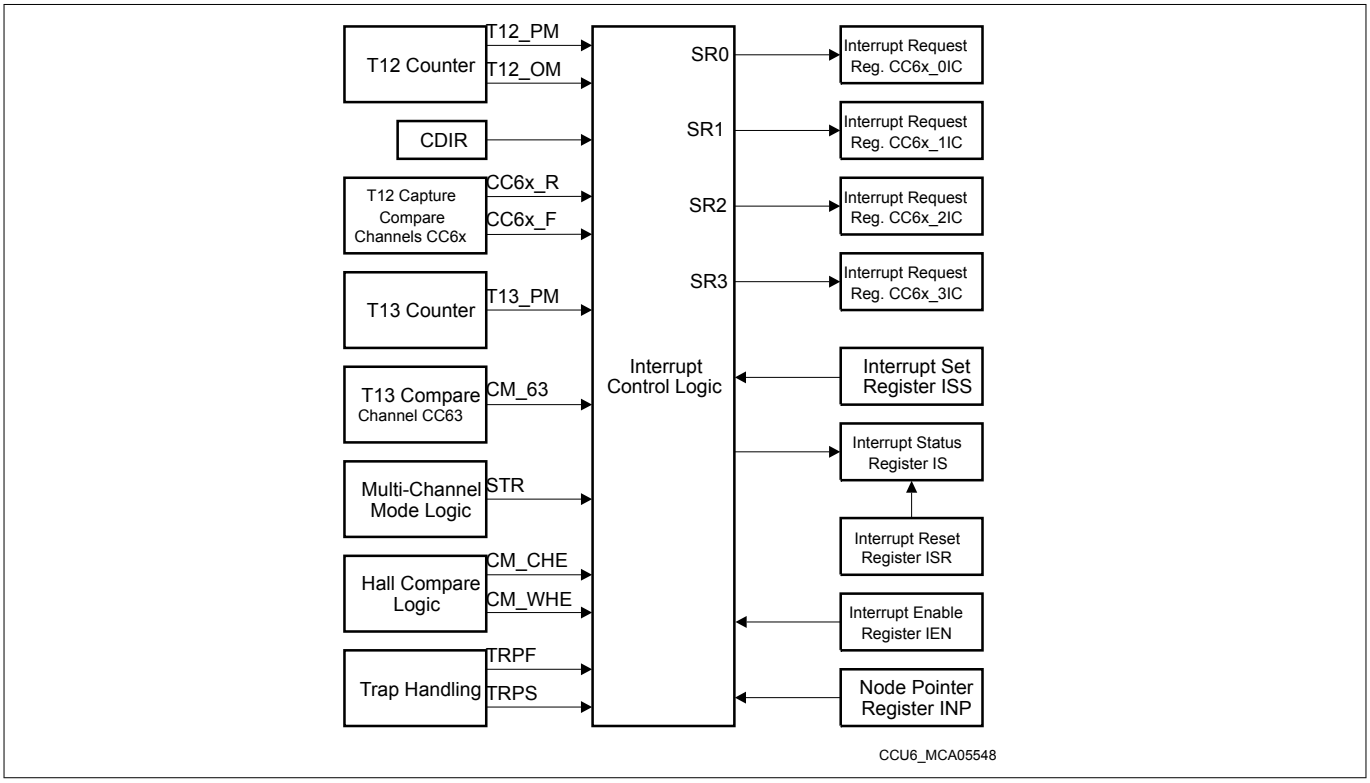


Figure 159 Interrupt sources and events

17.9 General module operation

This section provides information about the:

- Input selection (see Chapter 17.9.1)

17.9.1 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 $1/f_{CC6}$.

17.10 Capture/compare unit 6 (CCU6) registers

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.

The registers are addressed wordwise.

17.10.1 System registers

The registers CCU6_PISEL0 and CCU6_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.

17.10.1.1 Register overview - System registers (ascending offset address)

Table 118 Register overview - System registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_PISEL0	Port input select 0 register	$006C_H$	542
CCU6_PISEL2	Port input select 2 register	0074_H	544

17.10.2 Timer 12 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.

The register CCU6_T12MSEL contains control bits to select the capture/compare functionality of the three channels of timer T12. Table 119, Table 120 and Table 121 define and elaborate some of the capture/compare modes selectable. Refer to the register definition for the selection.

Table 119 Double-register capture modes

Description	
0100_B	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_B	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_B	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_B	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 120 Combined T12 modes

Description	
1000_B	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_B	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 121 Multi-input capture modes

Description	
1010_B	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_B	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_B	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_B	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_B	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_B	reserved (no capture or compare action)

17.10.2.1 Register overview - Timer 12 related registers (ascending offset address)

Table 122 Register overview - Timer 12 related registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_CC60SR	Capture/compare shadow register for channel CC60 register	0014_H	546
CCU6_CC61SR	Capture/compare shadow register for channel CC61 register	0018_H	547
CCU6_CC62SR	Capture/compare shadow register for channel CC62 register	$001C_H$	548
CCU6_T12PR	Timer T12 period register	0024_H	549
CCU6_T12DTC	Dead-time control register for timer T12 low register	$002C_H$	550
CCU6_CC60R	Capture/compare register for channel CC60 register	0034_H	552
CCU6_CC61R	Capture/compare register for channel CC61 register	0038_H	553
CCU6_CC62R	Capture/compare register for channel CC62 register	$003C_H$	554
CCU6_T12MSEL	T12 capture/compare mode select register	0040_H	555

(table continues...)

Table 122 (continued) Register overview - Timer 12 related registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_T12	Timer T12 counter register	0078_H	558

17.10.3 Timer 13 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.

The register CCU6_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 CCU6_T13 can always be read by software.

Timer T13 supports only edge-aligned mode (counting up).

17.10.3.1 Register overview - Timer 13 related registers (ascending offset address)

Table 123 Register overview - Timer 13 related registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_CC63R	Capture/compare for channel CC63 register	0000_H	559
CCU6_CC63SR	Capture/compare shadow for channel CC63 register	0020_H	560
CCU6_T13PR	Timer T13 period register	0028_H	561
CCU6_T13	Timer T13 counter register	$007C_H$	562

17.10.4 Capture/compare control registers

CCU6_CMPMODIF

Table 124 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	

17.10.4.1 Register overview - Capture and compare control registers (ascending offset address)

Table 125 Register overview - Capture and compare control registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_TCTR4	Timer control 4 register	0004_H	563
CCU6_CMPMODIF	Compare state modification register	0010_H	565
CCU6_TCTR0	Timer control 0 register	0030_H	567
CCU6_TCTR2	Timer control 2 register	0058_H	570
CCU6_CMPSTAT	Compare state register	0080_H	572

17.10.5 Global modulation control registers

CCU6_TRPCTR

Table 126 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

17.10.5.1 Register overview - Global modulation control registers (ascending offset address)

Table 127 Register overview - Global modulation control registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_PSLR	Passive state level register	0050_H	576
CCU6_MCMCTR	Multi-channel mode control register	0054_H	577
CCU6_TRPCTR	Trap control register	0060_H	579

17.10.6 Multi-channel modulation control registers

17.10.6.1 Register overview - Multi-channel modulation control registers (ascending offset address)

Table 128 Register overview - Multi-channel modulation control registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_MCMOUTS	Multi-channel mode output shadow register	0008_H	581
CCU6_MODCTR	Modulation control register	$005C_H$	583
CCU6_MCMOUT	Multi-channel mode output register	0064_H	585

17.10.7 Interrupt control registers

17.10.7.1 Register overview - Interrupt control registers (ascending offset address)

Table 129 Register overview - Interrupt control registers (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_ISR	Capture/compare interrupt status reset register	$000C_H$	587
CCU6_IEN	Capture/compare interrupt enable register	0044_H	589
CCU6_INP	Capture/compare interrupt node pointer register	0048_H	592
CCU6_ISS	Capture/compare interrupt status set register	$004C_H$	594
CCU6_IS	Capture/compare interrupt status register	0068_H	596

17.10.8 Capture/compare unit 6 (CCU6) register definition

17.10.8.1 Register address space - CCU6

Table 130 Registers address space - CCU6

Module	Base address	End address	Note
CCU6	$4000C000_H$	$4000FFFF_H$	Capture/Compare Unit 6 (CCU6) registers

17.10.8.2 Register overview - CCU6 (ascending offset address)

Table 131 Register overview - CCU6 (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_CC63R	Capture/compare for channel CC63 register	0000_H	559
CCU6_TCTR4	Timer control 4 register	0004_H	563
CCU6_MCMOUTS	Multi-channel mode output shadow register	0008_H	581
CCU6_ISR	Capture/compare interrupt status reset register	$000C_H$	587
CCU6_CMPMODIF	Compare state modification register	0010_H	565
CCU6_CC60SR	Capture/compare shadow register for channel CC60 register	0014_H	546
CCU6_CC61SR	Capture/compare shadow register for channel CC61 register	0018_H	547
CCU6_CC62SR	Capture/compare shadow register for channel CC62 register	$001C_H$	548
CCU6_CC63SR	Capture/compare shadow for channel CC63 register	0020_H	560
CCU6_T12PR	Timer T12 period register	0024_H	549
CCU6_T13PR	Timer T13 period register	0028_H	561
CCU6_T12DTC	Dead-time control register for timer T12 low register	$002C_H$	550
CCU6_TCTR0	Timer control 0 register	0030_H	567
CCU6_CC60R	Capture/compare register for channel CC60 register	0034_H	552
CCU6_CC61R	Capture/compare register for channel CC61 register	0038_H	553
CCU6_CC62R	Capture/compare register for channel CC62 register	$003C_H$	554
CCU6_T12MSEL	T12 capture/compare mode select register	0040_H	555
CCU6_IEN	Capture/compare interrupt enable register	0044_H	589
CCU6_INP	Capture/compare interrupt node pointer register	0048_H	592
CCU6_ISS	Capture/compare interrupt status set register	$004C_H$	594
CCU6_PSLR	Passive state level register	0050_H	576
CCU6_MCMCTR	Multi-channel mode control register	0054_H	577
CCU6_TCTR2	Timer control 2 register	0058_H	570
CCU6_MODCTR	Modulation control register	$005C_H$	583
CCU6_TRPCTR	Trap control register	0060_H	579
CCU6_MCMOUT	Multi-channel mode output register	0064_H	585
CCU6_IS	Capture/compare interrupt status register	0068_H	596

(table continues...)

Table 131 (continued) Register overview - CCU6 (ascending offset address)

Short name	Long name	Offset address	Page number
CCU6_PISEL0	Port input select 0 register	$006C_H$	542
CCU6_PISEL2	Port input select 2 register	0074_H	544
CCU6_T12	Timer T12 counter register	0078_H	558
CCU6_T13	Timer T13 counter register	$007C_H$	562
CCU6_CMPSTAT	Compare state register	0080_H	572

17.10.8.3 Port input select 0 register

CCU6_PISEL0 Offset address: $006C_H$

Port input select 0 register RESET_TYPE_3 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 **CC60_0**: The input pin for CC60_0 01_B **CC60_1**: The input pin for CC60_1 10_B **Reserved**: Reserved 11_B **Reserved**: 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 **CC61_0**: The input pin for CC61_0 01_B **CC61_1**: The input pin for CC61_1 10_B **Reserved**: Reserved 11_B **Reserved**: 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 **CC62_0**: The input pin for CC62_0 01_B **CC62_1**: The input pin for CC62_1 10_B **Reserved**: Reserved 11_B **Reserved**: 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 **CTRAP_0**: The input pin for CTRAP_0 01_B **CTRAP_1**: The input pin for CTRAP_1 10_B **CTRAP_2**: The input pin for CTRAP_2 11_B **CTRAP_3**: Signal from differential units
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 **CCPOS0_0**: The input pin for CCPOS0_0 01_B **CCPOS0_1**: The input pin for CCPOS0_1 10_B **CCPOS0_2**: The input pin for CCPOS0_2 11_B **CCPOS0_3**: The input pin for CCPOS0_3
ISPOS1	11:10	rw	**Input select for CCPOS1**

(table continues...)

(continued)

Field	Bits	Type	Description
			This bit field defines the port pin that is used for the CCPOS1 input signal. 00_B **CCPOS1_0**: The input pin for CCPOS1_0 01_B **CCPOS1_1**: The input pin for CCPOS1_1 10_B **CCPOS1_2**: The input pin for CCPOS1_2 11_B **CCPOS1_3**: The input pin for CCPOS1_3
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 **CCPOS2_0**: The input pin for CCPOS2_0 01_B **CCPOS2_1**: The input pin for CCPOS2_1 10_B **CCPOS2_2**: The input pin for CCPOS2_2 11_B **CCPOS2_3**: The input pin for CCPOS2_3
IST12HR	15:14	rw	**Input select for T12HR** This bit field defines the input signal used as T12HR input. 00_B **T12HRA**: Either signal T12HRA (if T12EXT = 0) or T12HRE (if T12EXT = 1) is selected 01_B **T12HRB**: Either signal T12HRB (if T12EXT = 0) or T12HRF (if T12EXT = 1) is selected 10_B **T12HRC**: Either signal T12HRC (if T12EXT = 0) or T12HRG (if T12EXT = 1) is selected 11_B **T12HRD**: Either signal T12HRD (if T12EXT = 0) or T12HRH (if T12EXT = 1) is selected

17.10.8.4 Port input select 2 register

CCU6_PISEL2 Offset address: 0074_H

Port input select 2 register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								T13EXT	T12EXT	ISCNT13		ISCNT12		IST13HR	
r								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 **T13HRA**: Either signal T13HRA (if T13EXT = 0) or T13HRE (if T13EXT = 1) is selected 01_B **T13HRB**: Either signal T13HRB (if T13EXT = 0) or T13HRF (if T13EXT = 1) is selected 10_B **T13HRC**: Either signal T13HRC (if T13EXT = 0) or T13HRG (if T13EXT = 1) is selected 11_B **T13HRD**: 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 **T12_prescaler**: The T12 prescaler generates the counting events. Bit TCTR4.T12CNT is not taken into account 01_B **TCTR4_T12CNT**: Bit TCTR4.T12CNT written with 1 is a counting event. The T12 prescaler is not taken into account 10_B **Rising_edge**: The timer T12 is counting each rising edge detected in the selected T12HR signal 11_B **Falling_edge**: 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 **T13_prescaler**: The T13 prescaler generates the counting events. Bit TCTR4.T13CNT is not taken into account 01_B **TCTR4_T13CNT**: Bit TCTR4.T13CNT written with 1 is a counting event. The T13 prescaler is not taken into account 10_B **Rising_edge**: The timer T13 is counting each rising edge detected in the selected T13HR signal 11_B **Falling_edge**: 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 **T12HR_D_A_**: T12HR[D:A], one of the signals T12HR[D:A] is selected 1_B **T12HR_H_E_**: T12HR[H:E], one of the signals T12HR[H:E] is selected

(table continues...)

(continued)

Field	Bits	Type	Description
T13EXT	7	rw	**Extension for T13HR inputs** This bit extends the 2-bit field IST13HR. 0_B **T13HR_D_A_**: T13HR[D:A], one of the signals T13HR[D:A] is selected 1_B **T13HR_H_E_**: T13HR[H:E], one of the signals T13HR[H:E] is selected
RES	15:8	r	**Reserved**

17.10.8.5 Capture/compare shadow register for channel CC60 register

The registers CC60R can only be read by software, the modification of the value is done by a shadow register transfer from register CC60SR. The corresponding shadow registers CC60SR can be read and written by software. In capture mode, the value of the T12 counter register can also be captured by registers CC60SR if the selected capture event is detected (depending on the selected mode).

CCU6_CC60SR Offset address: 0014_H

Capture/compare shadow register for channel CC60 register RESET_TYPE_3 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 0 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.

17.10.8.6 Capture/compare shadow register for channel CC61 register

The registers CC61R can only be read by software, the modification of the value is done by a shadow register transfer from register CC61SR. The corresponding shadow registers CC61SR can be read and written by software. In capture mode, the value of the T12 counter register can also be captured by registers CC61SR if the selected capture event is detected (depending on the selected mode).

CCU6_CC61SR Offset address: 0018_H

Capture/compare shadow register for channel CC61 register RESET_TYPE_3 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 1 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.

17.10.8.7 Capture/compare shadow register for channel CC62 register

The registers CC62R can only be read by software, the modification of the value is done by a shadow register transfer from register CC62SR. The corresponding shadow registers CC62SR can be read and written by software. In capture mode, the value of the T12 counter register can also be captured by registers CC62SR if the selected capture event is detected (depending on the selected mode).

CCU6_CC62SR Offset address: $001C_H$

Capture/compare shadow register for channel CC62 register RESET_TYPE_3 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 2 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.

17.10.8.8 Timer T12 period register

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.

CCU6_T12PR — Offset address: 0024_H

Timer T12 period register — RESET_TYPE_3 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).

17.10.8.9 Dead-time control register for timer T12 low register

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.

CCU6_T12DTC Offset address: $002C_H$

Dead-time control register for timer T12 low register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES0	**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	8	rw	**Dead-time enable bit 0** Bit DTE0 enables and disables the dead-time generation for compare channel 0 of timer T12. 0_B **DISABLED**: 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 **ENABLED**: 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
DTE1	9	rw	**Dead-time enable bit 1** Bit DTE1 enables and disables the dead-time generation for compare channel 1 of timer T12. 0_B **DISABLED**: 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 **ENABLED**: 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
DTE2	10	rw	**Dead-time enable bit 2** Bit DTE2 enables and disables the dead-time generation for compare channel 2 of timer T12. 0_B **DISABLED**: 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

(table continues...)

(continued)

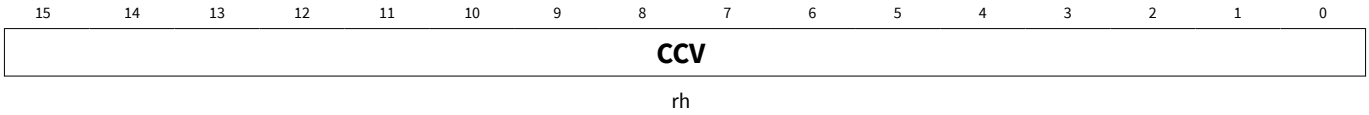
Field	Bits	Type	Description
			1_B **ENABLED**: Dead-time generation is enabled. The corresponding outputs switch from the passive state to the active state (according to the compare status) with the delay programmed in bit field DTM
RES	11	r	**Reserved**
DTR0	12	rh	**Dead-time run indication bit 0** Bit DTR0 indicate the status of the dead-time generation for compare channel 0 of timer T12. 0_B **Zero**: The value of the corresponding dead-time counter channel is 0 1_B **Not_zero**: The value of the corresponding dead-time counter channel is not 0
DTR1	13	rh	**Dead-time run indication bit 1** Bit DTR1 indicates the status of the dead-time generation for compare channel 1 of timer T12. 0_B **Zero**: The value of the corresponding dead-time counter channel is 0 1_B **Not_zero**: The value of the corresponding dead-time counter channel is not 0
DTR2	14	rh	**Dead-time run indication bit 2** Bit DTR2 indicates the status of the dead-time generation for compare channel 2 of timer T12. 0_B **Zero**: The value of the corresponding dead-time counter channel is 0 1_B **Not_zero**: The value of the corresponding dead-time counter channel is not 0
RES0	15	r	**Reserved** Returns 0 if read. Should be written with 0.

17.10.8.10 Capture/compare register for channel CC60 register

In compare mode, the registers CC60R is the actual compare registers for T12. The values stored in CC60R 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 CC60R if the corresponding capture event is detected.

CCU6_CC60R Offset address: 0034_H

Capture/compare register for channel CC60 register RESET_TYPE_3 value: 0000_H



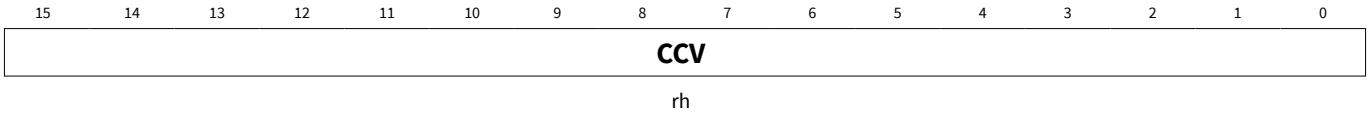
Field	Bits	Type	Description
CCV	15:0	rh	**Channel 0 capture/compare value** In compare mode, the bit fields CCV contain the values that are compared to the T12 counter value. In capture mode, the captured value of T12 can be read from these registers.

17.10.8.11 Capture/compare register for channel CC61 register

In compare mode, the registers CC61R is the actual compare registers for T12. The values stored in CC61R 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 CC61R if the corresponding capture event is detected.

CCU6_CC61R Offset address: 0038_H

Capture/compare register for channel CC61 register RESET_TYPE_3 value: 0000_H



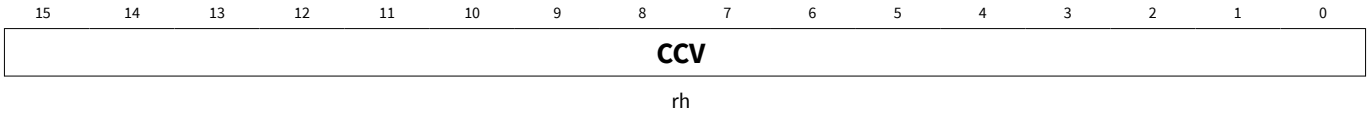
Field	Bits	Type	Description
CCV	15:0	rh	**Channel 1 capture/compare value** In compare mode, the bit fields CCV contain the values that are compared to the T12 counter value. In capture mode, the captured value of T12 can be read from these registers.

17.10.8.12 Capture/compare register for channel CC62 register

In compare mode, the registers CC62R is the actual compare registers for T12. The values stored in CC62R 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 CC62R if the corresponding capture event is detected.

CCU6_CC62R Offset address: $003C_H$

Capture/compare register for channel CC62 register RESET_TYPE_3 value: 0000_H



Field	Bits	Type	Description
CCV	15:0	rh	**Channel 2 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.

17.10.8.13 T12 capture/compare mode select register

CCU6_T12MSEL Offset address: 0040_H

T12 capture/compare mode select register RESET_TYPE_3 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	3:0	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: 0_H **Compare_outputs_disabled**: Compare outputs disabled, pins CC6n and COUT6n can be used for I/O. No capture action 1_H **Pin_CC6n_pin_COUT6n**: Compare output on pin CC6n, pin COUT6n can be used for I/O. No capture action 2_H **Pin_COUT6n_pin_CC6n**: Compare output on pin COUT6n, pin CC6n can be used for I/O. No capture action 3_H **Pins_COUT6n_and_CC6n**: Compare output on pins COUT6n and CC6n 4_H **Double_register_Capture_modes**: See Table "Register capture modes" … 7_H **Double_register_Capture_modes**: See Table "Register capture modes" 8_H **Hall_sensor_mode**: See Table "Register capture modes". In order to enable the hall edge detection, all three MSEL6x must be programmed to Hall sensor mode 9_H **Hysteresis_like_mode**: See Table "Combined T12 modes" A_H **Multi_input_Capture_modes**: See Table "Multi-input capture modes" … F_H **Multi_input_Capture_modes**: See Table "Multi-input capture modes"
MSEL61	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: 0_H **Compare_outputs_disabled**: Compare outputs disabled, compare outputs disabled, pins CC6n and COUT6n can be used for I/O. No capture action. 1_H **Pin_CC6n_pin_COUT6n**: Compare output on pin CC6n, pin COUT6n can be used for I/O; no capture action 2_H **Pin_COUT6n_pin_CC6n**: Pin CC6n, compare output on pin COUT6n, pin CC6n can be used for I/O. No capture action

(table continues...)

(continued)

Field	Bits	Type	Description
			3_H **Pins_COUT6n_and_CC6n**: Compare output on pins COUT6n and CC6n 4_H **Double_register_Capture_modes**: See Table "Double-register capture modes" … 7_H **Double_register_Capture_modes**: See Table "Double-register capture modes" 8_H **Hall_sensor_mode**: See Table "Combined T12 modes". In order to enable the hall edge detection, all three MSEL6x must be programmed to Hall sensor mode 9_H **Hysteresis_like_mode**: See Table "Combined T12 modes" A_H **Multi_input_Capture_modes**: See Table "Multi-input capture modes" … F_H **Multi_input_Capture_modes**: See Table "Multi-input capture modes"
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: 0_H **Compare_outputs_disabled**: Compare outputs disabled, pins CC6n and COUT6n can be used for I/O. No capture action 1_H **Pin_CC6n_pin_COUT6n**: Compare output on pin CC6n, pin COUT6n can be used for I/O. No capture action 2_H **Pin_COUT6n_pin_CC6n**: Compare output on pin COUT6n, pin CC6n can be used for I/O. No capture action 3_H **Pins_COUT6n_and_CC6n**: Compare output on pins COUT6n and CC6n 4_H **Double_register_Capture_modes**: See Table "Double-register capture modes" … 7_H **Double_register_Capture_modes**: See Table "Double-register capture modes" 8_H **Hall_sensor_mode**: See Table "Combined T12 modes". In order to enable the hall edge detection, all three MSEL6x must be programmed to Hall sensor mode 9_H **Hysteresis_like_mode**: See Table "Combined T12 modes" A_H **Multi_input_Capture_modes**: See Table "Multi-input capture modes" … F_H **Multi_input_Capture_modes**: See Table "Multi-input capture modes"
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.

(table continues...)

(continued)

Field	Bits	Type	Description
			000_B **Any**: Any edge at one of the inputs CCPOSx (x = 0, 1, 2) triggers the sampling 001_B **T13_compare_match**: A T13 compare-match triggers the sampling 010_B **T13_period_match**: A T13 period-match triggers the sampling 011_B **Hall**: The Hall sampling triggered by hardware sources is switched off 100_B **T12_period_match**: A T12 period-match (while counting up) triggers the sampling 101_B **T12_one_match**: A T12 one-match (while counting down) triggers the sampling 110_B **T12_compare_match_UP**: A T12 compare-match of channel 0 (while counting up) triggers the sampling 111_B **T12_compare_match_DOWN**: A T12 compare-match of channel 0 (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 **Not_active**: The delay bypass is not active. The dead-time counter DTC0 is generating a delay after the source signal becomes active 1_B **Active**: The delay bypass is active. The dead-time counter DTC0 is not used by the sampling of the Hall pattern

17.10.8.14 Timer T12 counter register

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.

CCU6_T12 Offset address: 0078_H

Timer T12 counter register RESET_TYPE_3 value: 0000_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 lower 8-bit counter value of timer T12.

17.10.8.15 Capture/compare for channel CC63 register

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.

CCU6_CC63R Offset address: 0000_H

Capture/compare for channel CC63 register RESET_TYPE_3 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 low byte** The bit field CCV contains the value that is compared to the T13 counter value.

17.10.8.16 Capture/compare shadow for channel CC63 register

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.

CCU6_CC63SR Offset address: 0020_H

Capture/compare shadow for channel CC63 register RESET_TYPE_3 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.

17.10.8.17 Timer T13 period register

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.

CCU6_T13PR — Offset address: 0028_H

Timer T13 period register — RESET_TYPE_3 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.

17.10.8.18 Timer T13 counter register

CCU6_T13 Offset address: $007C_H$

Timer T13 counter register RESET_TYPE_3 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 lower 8-bit counter value of timer T13.

17.10.8.19 Timer control 4 register

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.

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.

CCU6_TCTR4 Offset address: 0004_H

Timer control 4 register RESET_TYPE_3 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 **No_influence**: T12R is not influenced 1_B **T12R_cleared**: T12R is cleared, T12 stops counting
T12RS	1	w	**Timer T12 run set** Setting this bit sets the T12R bit. 0_B **No_influence**: T12R is not influenced 1_B **T12R_set**: T12R is set, T12 counts
T12RES	2	w	**Timer T12 reset** 0_B **No_effect**: No effect on T12 1_B **Zero**: 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**: No effect on the dead-time counters 1_B **Zero**: The three dead-time counter channels are reset to zero
RES	4, 12:11	r	**Reserved** Returns 0 if read.
T12CNT	5	w	**Timer T12 count event** 0_B **No_action**: No action 1_B **Count**: If enabled (PISEL2), timer T12 counts one step
T12STR	6	w	**Timer T12 shadow transfer request** 0_B **No_action** : No action 1_B **STE12_set**: STE12 is set, enabling the shadow transfer
T12STD	7	w	**Timer T12 shadow transfer disable** 0_B **No_action** : No action

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **STE12_reset**: STE12 is reset without triggering the shadow transfer
T13RR	8	w	**Timer T13 run reset** Setting this bit resets the T13R bit. 0_B **No_influence**: T13R is not influenced 1_B **T13R_cleared**: T13R is cleared, T13 stops counting
T13RS	9	w	**Timer T13 run set** Setting this bit sets the T13R bit. 0_B **No_influence**: T13R is not influenced 1_B **T13R_set**: T13R is set, T13 counts
T13RES	10	w	**Timer T13 reset** 0_B **No_effect**: No effect on T13 1_B **Zero**: 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** : No action 1_B **Count**: If enabled (PISEL2), timer T13 counts one step
T13STR	14	w	**Timer T13 shadow transfer request** 0_B **No_action** : No action 1_B **STE13_set**: STE13 is set, enabling the shadow transfer
T13STD	15	w	**Timer T13 shadow transfer disable** 0_B **No_action** : No action 1_B **STE13_reset**: STE13 is reset without triggering the shadow transfer

17.10.8.20 Compare state modification register

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.

CCU6_CMPMODIF Offset address: 0010_H

Compare state modification register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**MCC 63R**	**RES0**			**MCC 62R**	**MCC 61R**	**MCC 60R**	**RES**	**MCC 63S**	**RES**			**MCC 62S**	**MCC 61S**	**MCC 60S**
r	w	r			w	w	w	r	w	r			w	w	w

Field	Bits	Type	Description
MCC60S	0	w	**Capture/compare status modification bit 0 (set)** This bit is used to set the corresponding CC60ST 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 CC60ST-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".
MCC61S	1	w	**Capture/compare status modification bit 1 (set)** This bit is used to set the corresponding CC61ST 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 CC61ST-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".
MCC62S	2	w	**Capture/compare status modification bit 2 (set)** This bit is used to set the corresponding CC62ST 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 CC62ST-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".
RES	5:3, 7, 15	r	**Reserved**
MCC63S	6	w	**Capture/compare status modification bits (set)** This bit is used to set the corresponding CC63ST 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 CC63ST-bits by a single data write action.

(table continues...)

(continued)

Field	Bits	Type	Description
			Functionality see Table "Capture/compare status modification bits (set and reset)".
MCC60R	8	w	**Capture/compare status modification bit 0 (reset)** This bit is used to reset the corresponding CC60ST 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 CC620T-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".
MCC61R	9	w	**Capture/compare status modification bit 1 (reset)** This bit is used to reset the corresponding CC61ST 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 CC61ST-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".
MCC62R	10	w	**Capture/compare status modification bit 2 (reset)** This bit is used to reset the corresponding CC62ST 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 CC62ST-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".
RES0	13:11	r	**Reserved** Returns 0 if read.
MCC63R	14	w	**Capture/compare status modification bits (reset)** These bits are used to reset the corresponding CC63ST 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 CC63ST-bits by a single data write action. Functionality see Table "Capture/compare status modification bits (set and reset)".

17.10.8.21 Timer control 0 register

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).

CCU6_TCTR0 Offset address: 0030_H

Timer control 0 register RESET_TYPE_3 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 **1**: fT12 = fCCU 001_B **2**: fT12 = fCCU / 2 010_B **4**: fT12 = fCCU / 4 011_B **8**: fT12 = fCCU / 8 100_B **16**: fT12 = fCCU / 16 101_B **32**: fT12 = fCCU / 32 110_B **64**: fT12 = fCCU / 64 111_B **128**: fT12 = fCCU / 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 **DISABLED**: The additional prescaler for T12 is disabled 1_B **ENABLED**: 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 **Stop**: Timer T12 is stopped 1_B **Run**: Timer T12 is running
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.

(table continues...)

(continued)

Field	Bits	Type	Description
			A T12 shadow transfer event is a period-match while counting up or a one-match while counting down. 0_B **DISABLED**: The shadow register transfer is disabled 1_B **ENABLED**: 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 **UP**: T12 counts up 1_B **DOWN**: 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 **1**: fT13 = fCCU 001_B **2**: fT13 = fCCU / 2 010_B **4**: fT13 = fCCU / 4 011_B **8**: fT13 = fCCU / 8 100_B **16**: fT13 = fCCU / 16 101_B **32**: fT13 = fCCU/ 32 110_B **64**: fT13 = fCCU / 64 111_B **128**: fT13 = fCCU / 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 **DISABLED**: The additional prescaler for T13 is disabled 1_B **ENABLED**: 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 **Stop**: Timer T13 is stopped 1_B **Run**: Timer T13 is running
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.

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **DISABLED**: The shadow register transfer is disabled 1_B **ENABLED**: The shadow register transfer is enabled
RES	15:14	r	**Reserved** Returns 0 if read.

17.10.8.22 Timer control 2 register

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.

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.

CCU6_TCTR2 Offset address: 0058_H

Timer control 2 register RESET_TYPE_3 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 **DISABLED**: The single-shot mode is disabled, no hardware action on T12R 1_B **ENABLED**: 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_action**: No hardware action on T13R 1_B **ENABLED**: 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** : No action 001_B **Channel_0**: Set T13R on a T12 compare event on channel 0 010_B **Channel_1**: Set T13R on a T12 compare event on channel 1

(table continues...)

(continued)

Field	Bits	Type	Description
			011_B **Channel_2**: Set T13R on a T12 compare event on channel 2 100_B **Channel_0_1_2**: Set T13R on any T12 compare event on the channels 0, 1, or 2 101_B **Period_match**: Set T13R upon a period-match of T12 110_B **Zero_match**: Set T13R upon a zero-match of T12 (while counting up) 111_B **CCPOSx**: Set T13R on any edge of inputs CCPOSx
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** : No action 01_B **Up**: While T12 is counting up 10_B **Down**: While T12 is counting down 11_B **Independent**: Independent on the count direction of T12
RES	7, 15:12	r	**Reserved** Returns 0 if read.
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 **DISABLED**: The external setting of T12R is disabled 01_B **Rising_edge**: Bit T12R is set if a rising edge of signal T12HR is detected 10_B **Falling_edge**: Bit T12R is set if a falling edge of signal T12HR is detected 11_B **Edge**: 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 **DISABLED**: The external setting of T13R is disabled 01_B **Rising_edge**: Bit T13R is set if a rising edge of signal T13HR is detected 10_B **Falling_edge**: Bit T13R is set if a falling edge of signal T13HR is detected 11_B **Edge**: Bit T13R is set if an edge of signal T13HR is detected

17.10.8.23 Compare state register

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.

CCU6_CMPSTAT Offset address: 0080_H

Compare state register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T13IM	**COUT63PS**	**COUT62PS**	**CC62PS**	**COUT61PS**	**CC61PS**	**COUT60PS**	**CC60PS**	**RES**	**CC63ST**	**CCPOS2**	**CCPOS1**	**CCPOS0**	**CC62ST**	**CC61ST**	**CC60ST**
rwh	rwh	rwh	rwh	rwh	rwh	rwh	rwh	r	rh	rh	rh	rh	rh	rh	rh

Field	Bits	Type	Description
CC60ST	0	rh	**Capture/compare state bits** 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 **Less**: 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 **Greater**: In compare mode, the counter value is greater than or equal to the compare value. In capture mode, the selected edge has been detected
CC61ST	1	rh	**Capture/compare state bits** 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 **Less**: 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 **Greater**: In compare mode, the counter value is greater than or equal to the compare value; In capture mode, the selected edge has been detected
CC62ST	2	rh	**Capture/compare state bits** 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 **Less**: 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 **Greater**: 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	3	rh	**Sampled Hall pattern bit 0**

(table continues...)

(continued)

Field	Bits	Type	Description
			Bit CCPOS0 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 **Zero**: The input CCPOS0 has been sampled as 0 1_B **One**: The input CCPOS0 has been sampled as 1
CCPOS1	4	rh	**Sampled Hall pattern bit 1** Bit CCPOS1 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 **Zero**: The input CCPOS1 has been sampled as 0 1_B **One**: The input CCPOS1 has been sampled as 1
CCPOS2	5	rh	**Sampled Hall pattern bit 2** Bit CCPOS2 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 **Zero**: The input CCPOS2 has been sampled as 0 1_B **One**: The input CCPOS2 has been sampled as 1
CC63ST	6	rh	**Capture/compare state bits** Bit CC63ST is related to T13. These bits are set and reset according to the T12 and T13 switching rules. 0_B **Less**: 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 **Greater**: In compare mode, the counter value is greater than or equal to the compare value. In capture mode, the selected edge has been detected
RES	7	r	**Reserved** Returns 0 if read.
CC60PS	8	rwh	**Passive state select for compare outputs** Bits CC6xPS 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 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0 1_B **One**: The corresponding compare output drives passive level while CC6xST is 1
COUT60PS	9	rwh	**Passive state select for compare outputs** Bits COUT6xPS select the state of the corresponding compare channel, which is considered to be the passive state. During the passive state, the

(table continues...)

(continued)

Field	Bits	Type	Description
			passive level (defined in register PSLR) is driven by the output pin. Bits 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0 1_B **One**: The corresponding compare output drives passive level while CC6xST is 1
CC61PS	10	rwh	**Passive state select for compare outputs** Bits CC6xPS 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 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0 1_B **One**: The corresponding compare output drives passive level while CC6xST is 1
COUT61PS	11	rwh	**Passive state select for compare outputs** Bits COUT6xPS 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 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0 1_B **One**: The corresponding compare output drives passive level while CC6xST is 1
CC62PS	12	rwh	**Passive state select for compare outputs** Bits CC6xPS 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 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **One**: The corresponding compare output drives passive level while CC6xST is 1
COUT62PS	13	rwh	**Passive state select for compare outputs** COUT6xPS 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 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0 1_B **One**: The corresponding compare output drives passive level while CC6xST is 1
COUT63PS	14	rwh	**Passive state select for compare outputs** Bits COUT6xPS 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 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 **Zero**: The corresponding compare output drives passive level while CC6xST is 0 1_B **One**: 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 **Not_inverted**: T13 output is not inverted 1_B **Inverted**: T13 output is inverted for further modulation

17.10.8.24 Passive state level register

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”).

Note: 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.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.

CCU6_PSLR Offset address: 0050_H

Passive state level register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								PSL63	RES0	PSL					
r								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: 00_H **Level_0**: The passive level is 0 01_H **Level_1**: The passive level is 1
RES0	6	r	**Reserved** Returns 0 if read.
PSL63	7	rwh	**Passive state level of output COUT63** This bit field defines the passive level of the output pin COUT63. 0_B **Level_0**: The passive level is 0 1_B **Level_1**: The passive level is 1
RES	15:8	r	**Reserved**

17.10.8.25 Multi-channel mode control register

Register MCMCTR contains control bits for the multi-channel functionality.

CCU6_MCMCTR Offset address: 0054_H

Multi-channel mode control register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES					STE13U	STE12D	STE12U	RES0		SWSYN		RES	SWSEL		
r					rw	rw	rw	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_request**: No trigger request will be generated 001_B **Correct_pattern**: Correct hall pattern on CCPOSx detected 010_B **T13_period_match**: T13 period-match detected (while counting up) 011_B **T12_one_match**: T12 one-match (while counting down) 100_B **T12_channel_1_compare_match**: T12 channel 1 compare-match detected (phase delay function) 101_B **T12_period_match**: T12 period match detected (while counting up) else reserved, no trigger request will be generated
RES	3, 15:11	r	**Reserved**
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**: T13 zero-match triggers the shadow transfer 10_B **T12_zero_match**: A T12 zero-match (while counting up) triggers the shadow transfer 11_B **Reserved**: Reserved. No action
RES0	7:6	r	**Reserved** Returns 0 if read.
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**: No action

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLED**: The T12_ST shadow transfer mechanism is enabled if MCMEN = 1
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** : No action 1_B **ENABLED**: 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** : No action 1_B **ENABLED**: The T13_ST shadow transfer mechanism is enabled if MCMEN = 1

17.10.8.26 Trap control register

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.

CCU6_TRPCTR Offset address: 0060_H

Trap control register RESET_TYPE_3 value: 0000_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRPPEN	**TRPEN13**	**TRPEN**						**RES**					**TRPM2**	**TRPM10**	
rw	rw	rw						r					rw	rw	

Field	Bits	Type	Description
TRPM10	1:0	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. The combination (TRPM1, TRPM0) leads to: 00_B **T12_zero_match**: 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 **T13_zero_match**: 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**: Reserved 11_B **Immediately**: The trap state is left (return to normal operation according to TRPM2) immediately without any synchronization to T12 or T13
TRPM2	2	rw	**Trap mode control bit 2** 0_B **Hardware_reset**: The trap state can be left (return to normal operation = bit TRPS = 0) as soon as the input 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 TRPM10) is detected. 1_B **Software_reset**: 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 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 TRPM10) is detected.
RES	7:3	r	**Reserved** Returns 0 if read.
TRPEN	13:8	rw	**Trap enable control** Setting these bits enables the trap functionality for the following corresponding output signals:

(table continues...)

(continued)

Field	Bits	Type	Description
			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: 00_H **DISABLED**: The trap functionality of the corresponding output signal is disabled; the output state is independent from bit TRPS 01_H **ENABLED**: 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 **DISABLED**: The trap functionality for T13 is disabled; timer T13 (if selected and enabled) provides PWM functionality even while TRPS = 1 1_B **ENABLED**: 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 **DISABLED**: The trap functionality based on the input pin CTRAP is disabled. A trap can only be generated by software by setting bit TRPF 1_B **ENABLED**: The trap functionality based on the input pin CTRAP is enabled. A trap can be generated by software by setting bit TRPF or by CTRAP = 0

17.10.8.27 Multi-channel mode output shadow register

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.

CCU6_MCMOUTS Offset address: 0008_H

Multi-channel mode output shadow register RESET_TYPE_3 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.
RES	6, 14	r	**Reserved** Returns 0 if read.
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 **By_hardware**: 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 **By_software**: Bit field MCMP is updated by the value written to bit field MCMPS
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.
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 **By_hardware**: 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

(table continues...)

(continued)

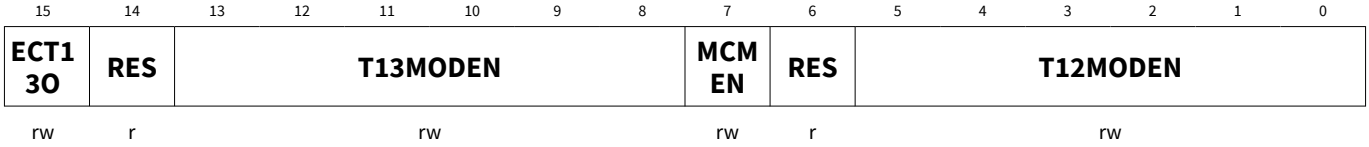
Field	Bits	Type	Description
			1_B **By_software**: The bit fields CURH and EXPH are updated by the value written to the bit fields CURHS and EXPHS

17.10.8.28 Modulation control register

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.

CCU6_MODCTR Offset address: $005C_H$

Modulation control register RESET_TYPE_3 value: 0000_H



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: 00_H **DISABLED**: The modulation of the corresponding output signal by a T12 PWM pattern is disabled 01_H **ENABLED**: The modulation of the corresponding output signal by a T12 PWM pattern is enabled
RES	6, 14	r	**Reserved** Returns 0 if read.
MCMEN	7	rw	**Multi-channel mode enable** 0_B **DISABLED**: The modulation of the corresponding output signal by a multi-channel pattern according to bit field MCMOUT is disabled 1_B **ENABLED**: The modulation of the corresponding output signal by a multi-channel pattern according to bit field MCMOUT is enabled
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

(table continues...)

(continued)

Field	Bits	Type	Description
			The enable feature of the modulation is defined as follows: 00_H **DISABLED**: The modulation of the corresponding output signal by a T13 PWM pattern is disabled 01_H **ENABLED**: The modulation of the corresponding output signal by a T13 PWM pattern is enabled
ECT13O	15	rw	**Enable compare timer T13 output** 0_B **DISABLED**: The alternate output function COUT63 is disabled 1_B **ENABLED**: The alternate output function COUT63 is enabled for the PWM signal generated by T13

17.10.8.29 Multi-channel mode output register

Register MCMOUT shows the multi-channel control bits that are currently used. Register MCMOUT is defined as follows:

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).

CCU6_MCMOUT Offset address: 0064_H

Multi-channel mode output register RESET_TYPE_3 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. 00_H **Passive**: The output is set to the passive state. The PWM generated by T12 or T13 is not taken into account 01_H **PWM**: The output can deliver the PWM generated by T12 or T13 (according to register MODCTR)
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 **No_shadow_transfer**: No shadow transfer, currently, no shadow transfer from MCMPS to MCMP is requested 1_B **Shadow_transfer**: 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, 15:14	r	**Reserved** Returns 0 if read.

(table continues...)

(continued)

Field	Bits	Type	Description
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).

17.10.8.30 Capture/compare interrupt status reset register

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.

CCU6_ISR Offset address: $000C_H$

Capture/compare interrupt status reset register RESET_TYPE_3 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**: No action 1_B **Reset**: Bit CC60R in register IS will be reset
RCC60F	1	w	**Reset capture, compare-match falling edge flag** 0_B **No_action**: No action 1_B **Reset**: Bit CC60F in register IS will be reset
RCC61R	2	w	**Reset capture, compare-match rising edge Flag** 0_B **No_action**: No action 1_B **Reset**: Bit CC61R in register IS will be reset
RCC61F	3	w	**Reset capture, compare-match falling edge flag** 0_B **No_action**: No action 1_B **Reset**: Bit CC61F in register IS will be reset
RCC62R	4	w	**Reset capture, compare-match rising edge flag** 0_B **No_action** : No action 1_B **Reset**: Bit CC62R in register IS will be reset
RCC62F	5	w	**Reset capture, compare-match falling edge flag** 0_B **No_action** : No action 1_B **Reset**: Bit CC62F in register IS will be reset
RT12OM	6	w	**Reset timer T12 one-match flag** 0_B **No_action**: No action 1_B **Reset**: Bit T12OM in register IS will be reset
RT12PM	7	w	**Reset timer T12 period-match flag** 0_B **No_action**: No action 1_B **Reset**: Bit T12PM in register IS will be reset
RT13CM	8	w	**Reset timer T13 compare-match flag** 0_B **No_action**: No action 1_B **Reset**: Bit T13CM in register IS will be reset
RT13PM	9	w	**Reset timer T13 period-Match flag** 0_B **No_action**: No action 1_B **Reset**: Bit T13PM in register IS will be reset

(table continues...)

(continued)

Field	Bits	Type	Description
RTRPF	10	w	**Reset trap flag** 0_B **No_action**: No action 1_B **Reset**: Bit TRPF in register IS will be reset (not taken into account while input CTRAP= 0 and TRPPEN = 1)
RES	11	r	**Reserved** Returns 0 if read.
RCHE	12	w	**Reset correct Hall event flag** 0_B **No_action**: No action 1_B **Reset**: Bit CHE in register IS will be reset
RWHE	13	w	**Reset wrong Hall event flag** 0_B **No_action**: No action 1_B **Reset**: Bit WHE in register IS will be reset
RIDLE	14	w	**Reset IDLE flag** 0_B **No_action**: No action 1_B **Reset**: Bit IDLE in register IS will be reset
RSTR	15	w	**Reset STR flag** 0_B **No_action**: No action 1_B **Reset**: Bit STR in register IS will be reset

17.10.8.31 Capture/compare interrupt enable register

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.

CCU6_IEN Offset address: 0044_H

Capture/compare interrupt enable register RESET_TYPE_3 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**: No interrupt will be generated if the set condition for bit CC60R in register IS occurs 1_B **Interrupt**: 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
ENCC60F	1	rw	**Capture, compare-match falling edge interrupt enable for channel 0** 0_B **No_interrupt**: No interrupt will be generated if the set condition for bit CC60F in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit CC61R in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit CC61F in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit CC62R in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit CC62F in register IS occurs

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit T12OM in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit T12PM in register IS occurs 1_B **Interrupt**: 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
ENT13CM	8	rw	**Enable interrupt for T13 compare-match** 0_B **No_interrupt**: No interrupt will be generated if the set condition for bit T13CM in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit T13PM in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit TRPF in register IS occurs 1_B **Interrupt**: 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.
ENCHE	12	rw	**Enable interrupt for correct Hall Event** 0_B **No_interrupt**: No interrupt will be generated if the set condition for bit CHE in register IS occurs 1_B **Interrupt**: 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**: No interrupt will be generated if the set condition for bit WHE in register IS occurs

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **Interrupt**: 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 **IDLE_not_set**: The bit IDLE is not automatically set when a wrong hall event is detected 1_B **IDLE_set**: 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**: No interrupt will be generated if the set condition for bit STR in register IS occurs 1_B **Interrupt**: 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

17.10.8.32 Capture/compare interrupt node pointer register

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.

CCU6_INP Offset address: 0048_H

Capture/compare interrupt node pointer register RESET_TYPE_3 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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: 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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: 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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: 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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: Interrupt output line SR3 is selected
INPERR	9:8	rw	**Interrupt node pointer for error interrupts**

(table continues...)

(continued)

Field	Bits	Type	Description
			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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: 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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: 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 **SR0**: Interrupt output line SR0 is selected 01_B **SR1**: Interrupt output line SR1 is selected 10_B **SR2**: Interrupt output line SR2 is selected 11_B **SR3**: Interrupt output line SR3 is selected
RES	15:14	r	**Reserved** Returns 0 if read.

17.10.8.33 Capture/compare interrupt status set register

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.

CCU6_ISS Offset address: $004C_H$

Capture/compare interrupt status set register RESET_TYPE_3 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** : No action 1_B **Set**: Bit CC60R in register IS will be set
SCC60F	1	w	**Set capture, compare-match falling edge flag** 0_B **No_action** : No action 1_B **Set**: Bit CC60F in register IS will be set
SCC61R	2	w	**Set capture, compare-match rising edge flag** 0_B **No_action** : No action 1_B **Set**: Bit CC61R in register IS will be set
SCC61F	3	w	**Set capture, compare-match falling edge flag** 0_B **No_action** : No action 1_B **Set**: Bit CC61F in register IS will be set
SCC62R	4	w	**Set capture, compare-match rising edge flag** 0_B **No_action** : No action 1_B **Set**: Bit CC62R in register IS will be set
SCC62F	5	w	**Set capture, compare-match falling edge flag** 0_B **No_action** : No action 1_B **Set**: Bit CC62F in register IS will be set
ST12OM	6	w	**Set timer T12 one-match flag** 0_B **No_action** : No action 1_B **Set**: Bit T12OM in register IS will be set
ST12PM	7	w	**Set timer T12 period-match flag** 0_B **No_action** : No action 1_B **Set**: Bit T12PM in register IS will be set
ST13CM	8	w	**Set timer T13 compare-match flag** 0_B **No_action** : No action 1_B **Set**: Bit T13CM in register IS will be set
ST13PM	9	w	**Set timer T13 period-match flag** 0_B **No_action** : No action

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **Set**: Bit T13PM in register IS will be set
STRPF	10	w	**Set trap flag** 0_B **No_action** : No action 1_B **Set**: Bits TRPF and TRPS in register IS will be set
SWHC	11	w	**Software Hall compare** 0_B **No_action** : No action 1_B **Set**: The Hall compare action is triggered
SCHE	12	w	**Set correct Hall event flag** 0_B **No_action** : No action 1_B **Set**: Bit CHE in register IS will be set
SWHE	13	w	**Set wrong Hall event flag** 0_B **No_action** : No action 1_B **Set**: Bit WHE in register IS will be set
SIDLE	14	w	**Set IDLE flag** 0_B **No_action** : No action 1_B **Set**: Bit IDLE in register IS will be set
SSTR	15	w	**Set STR flag** 0_B **No_action**: No action 1_B **Set**: Bit STR in register IS will be set

17.10.8.34 Capture/compare interrupt status register

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).

CCU6_IS Offset address: 0068_H

Capture/compare interrupt status register RESET_TYPE_3 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	0	rh	**Capture, compare-match rising edge flag** 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 CC60. 0_B **Not_occurred**: The event has not yet occurred since this bit has been reset for the last time 1_B **Detected**: The event described above has been detected
ICC60F	1	rh	**Capture, compare-match falling edge flag** 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 CC60. 0_B **Not_occurred**: The event has not yet occurred since this bit has been reset for the last time 1_B **Detected**: The event described above has been detected
ICC61R	2	rh	**Capture, compare-match rising edge flag** 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 CC61. 0_B **Not_occurred**: The event has not yet occurred since this bit has been reset for the last time 1_B **Detected**: The event described above has been detected
ICC61F	3	rh	**Capture, compare-match falling edge flag**

(table continues...)

(continued)

Field	Bits	Type	Description
			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 CC61. 0_B **Not_occurred**: The event has not yet occurred since this bit has been reset for the last time 1_B **Detected**: The event described above has been detected
ICC62R	4	rh	**Capture, compare-match rising edge flag** 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 CC62. 0_B **Not_occurred**: The event has not yet occurred since this bit has been reset for the last time 1_B **Detected**: The event described above has been detected
ICC62F	5	rh	**Capture, compare-match falling edge flag** 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 CC62. 0_B **Not_occurred**: The event has not yet occurred since this bit has been reset for the last time 1_B **Detected**: The event described above has been detected
T12OM	6	rh	**Timer T12 one-match flag** 0_B **Not_detected**: 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 **Detected**: A timer T12 one-match (while counting down) has been detected
T12PM	7	rh	**Timer T12 period-match flag** 0_B **Not_detected**: 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 **Detected**: A timer T12 period-match (while counting up) has been detected
T13CM	8	rh	**Timer T13 compare-match flag** 0_B **Not_detected**: A timer T13 compare-match has not yet been detected since this bit has been reset for the last time 1_B **Detected**: A timer T13 compare-match has been detected
T13PM	9	rh	**Timer T13 period-match flag** 0_B **Not_detected**: A timer T13 period-match has not yet been detected since this bit has been reset for the last time 1_B **Detected**: 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 **Not_detected**: The trap condition has not been detected

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **Detected**: The trap condition has been detected (input 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 **Not_active**: The trap state is not active 1_B **Active**: 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 **Not_detected**: 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 **Detected**: 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 **Not_detected**: 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 **Detected**: 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**: No action 1_B **Idle**: Bit field MCMP is cleared and held to 0, the selected outputs are set to passive state
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 **No**: The shadow transfer has not yet taken place 1_B **Yes**: The shadow transfer has taken place

17.11 MOTIX™ TLE984xQX module implementation details

This section describes the CCU6 module interfaces with the clock control, port connections, interrupt control, and address decoding.

17.11.1 Interfaces of the CCU6 module

An overview of the CCU6 kernel I/O interface is shown in Figure 160.

The bus peripheral interface (BPI) enables the CCU6 kernel to be attached to the 8-bit bus. The BPI consists of a clock control logic which gates the clock input to the kernel, and an address decoder for special function registers (SFRs) in the CCU6 kernel.

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 OCDS 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 (CCU6_TRPCTR.TRPENx = 1) are set to respective passive levels (similar to TRAP state). In addition, all CCU6 inputs are frozen. Refer to SCU Chapter 6.10 and OCDS chapter.

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 SCU_PMCON.CCU6_DIS request bit is set, both T12 and T13 are immediately stopped and PWM outputs enabled for the trap condition (CCU6_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 6.9.

Table 132 CCU6/T21CCU interconnection

CCU6 input	T21CCU output
T12HRD	T21CCU_CCTCON.CCTST
T13HRD	T21CCU_CCTCON.CCTST

Figure 160 shows all interrupt and interface signals and GPIO interface associated with the CCU6 module kernel.

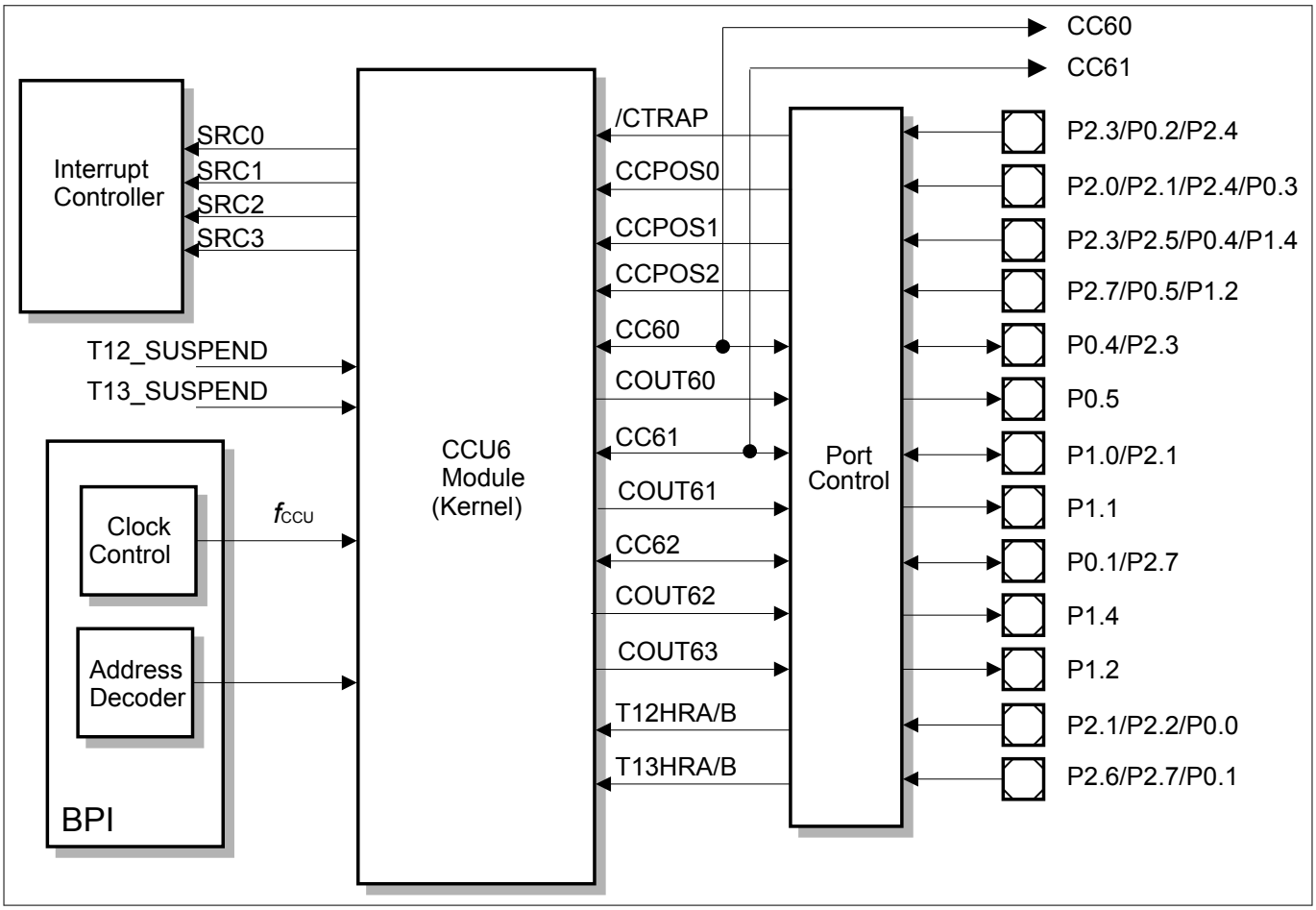


Figure 160 Interconnections of the CCU6 module

18 UART1/UART2

18.1 Features

- Full-duplex asynchronous modes
  - 8-bit or 9-bit data frames, LSB first
  - Fixed or variable baud-rate
- Receive buffered (1 Byte)
- Multiprocessor communication
- Interrupt generation on the completion of a data transmission or reception
- Baud-rate generator with fractional divider for generating a wide range of baud-rates, e.g. 9.6 kBaud, 19.2 kBaud, 115.2 kBaud, 125 kBaud, 250 kBaud, 500 kBaud
- Hardware logic for break and sync byte detection
- for UART1: LIN support: connected to timer channel for synchronization to LIN baud-rate

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 (that means serial interrupt not enabled), TI and RI can also be used for polling the serial interface.

18.2 Introduction

The UART1/UART2 provide a full-duplex asynchronous receiver/transmitter, that is it can transmit and receive simultaneously. They are also receive-buffered, that is, they 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 reception of the second byte is complete, the previous byte will be lost. The serial port receive and transmit registers are both accessed at special function register (SFR) SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

18.2.1 Block diagram

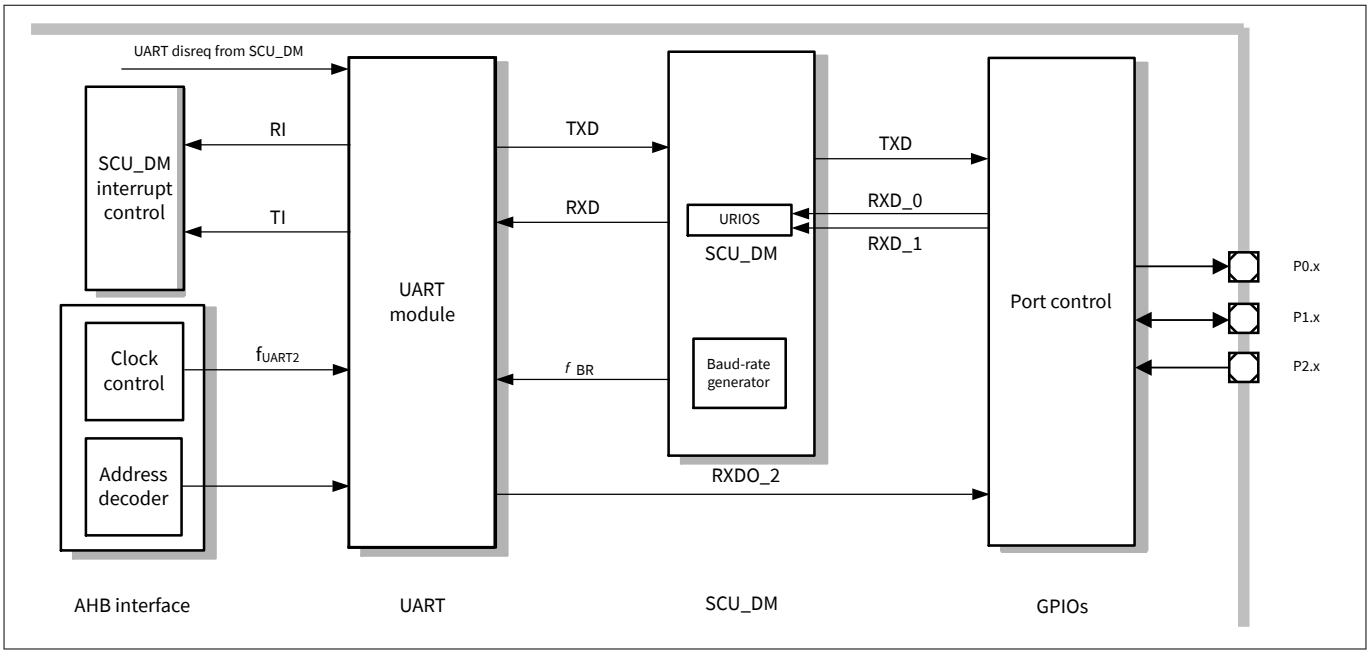


Figure 161 UART block diagram

18.3 UART modes

The UART1/UART2 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 their corresponding values, as shown in the following table.

Mode 1 example: 8 data bits, 1 start bit, 1 stop bit, no parity selection, 16 times oversampled (majority decision of bits 6, 7, 8), receive and transmit register double buffered, Tx/Rx IRQ(s).

Table 133 UART modes

SM0	SM1	Operating mode	baud-rate
0	0	Mode 0: 8-bit shift register	$f_{sys}/2$
0	1	Mode 1: 8-bit shift UART	Variable
1	0	Mode 2: 9-bit shift UART	$f_{sys}/64$ or $f_{sys}/32$
1	1	Mode 3: 9-bit shift UART	Variable

18.3.1 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. The data will be written to the transmit shift register with a 1 at the 9th bit position. 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 $f_{sys}/2$ where f_{sys} is the input clock frequency, that is one bit per machine cycle.

18.3.2 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 are 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 (SCON.0), and either SM2 = 0 (SCON.5) (see Chapter 18.4) 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 162.

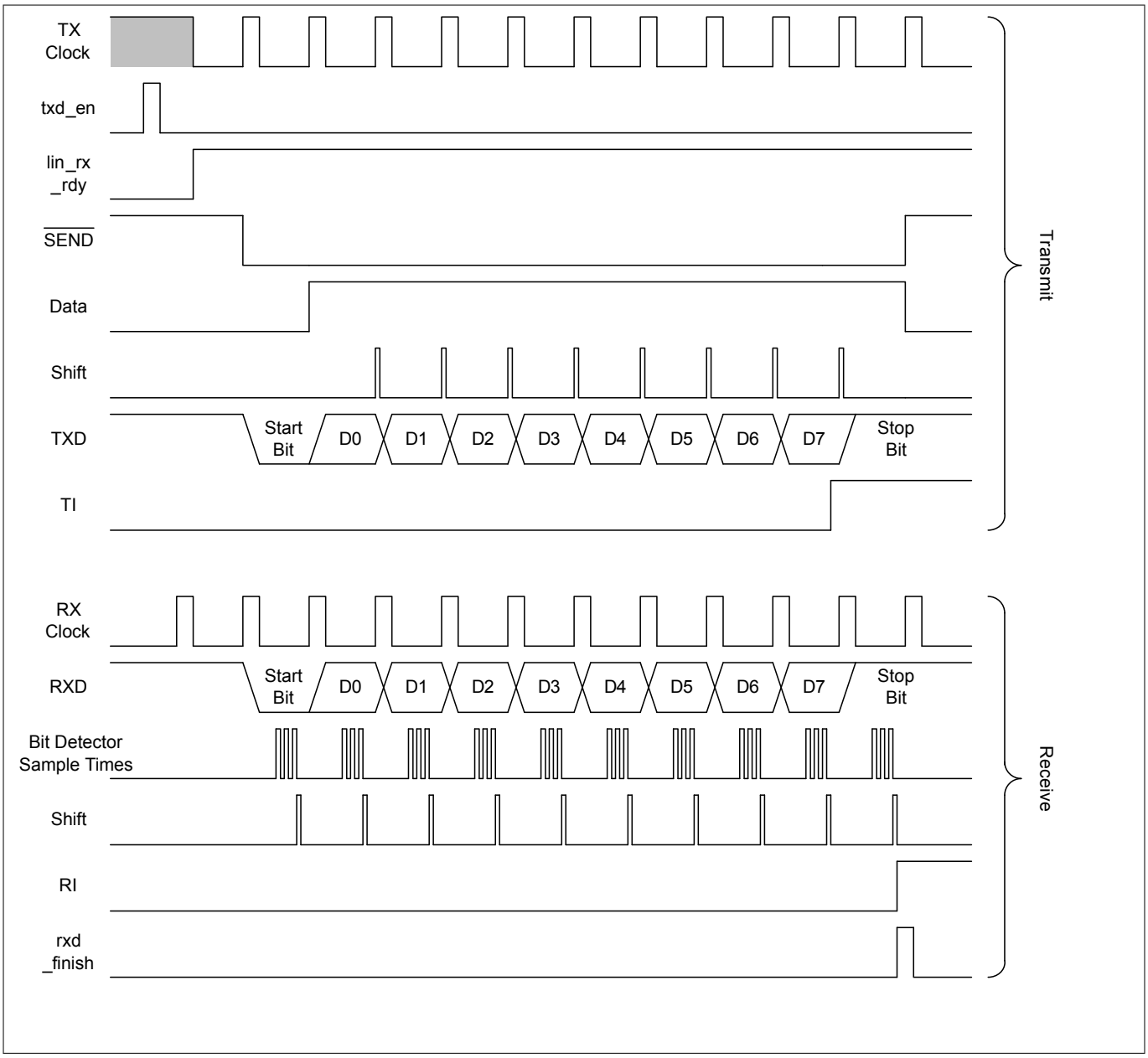


Figure 162 Serial interface, mode 1, timing diagram

18.3.3 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 of 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 (SCON.0), and either SM2 = 0 (SCON.5) (see Chapter 18.4) 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 $f_{sys}/64$ or $f_{sys}/32$.

18.3.4 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.

The associated timings for transmit/receive in modes 2 and 3 are illustrated in the following figure.

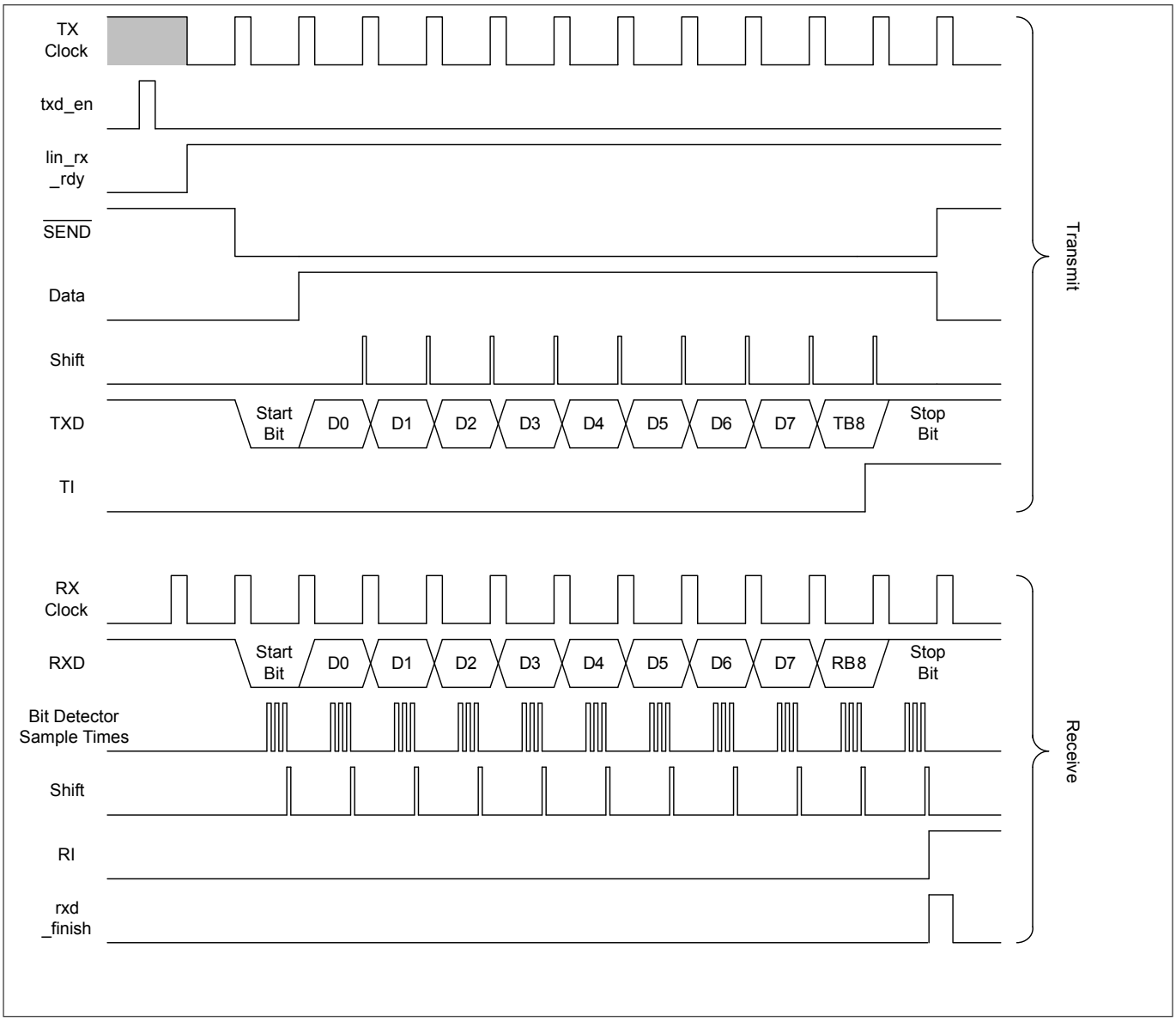


Figure 163 Serial interface, modes 2 and 3, timing diagram

18.4 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 (SCON.2). 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 register 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 the 9th bit. The 9th bit in an address byte is 1 and in a data byte the 9th bit is 0. 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 SM2 bits as set and ignore the incoming data bytes.

Note: 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.

18.5 Interrupts

The two UART interrupts can be separately enabled or disabled by setting or clearing their corresponding enable bits in SCU SFR MODIEN2. An overview of the UART interrupt sources is shown in the following table.

Table 134 UART interrupt sources

Interrupt	Flag	Interrupt enable bit
Reception completed	SCON.RI	SCU_MODIEN2.RIEN1/2
Transmission completed	SCON.TI	SCU_MODIEN2.TIEN1/2

18.6 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 $f_{sys}/2$ and $f_{sys}/64$ 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 Chapter 18.6.1).

“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_{sys} refers to the input clock frequency.

18.6.1 Baud-rate generator

Note: The register names used here refer to UART1. For UART2, the register names need to be adapted accordingly.

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 clock is derived through a prescaler (f_{DIV}) from the input clock f_{sys}. The baud-rate timer counts downwards and can be started or stopped through the baud-rate control run bit BCON1.BR1_R. Each underflow of the timer provides one clock pulse to the serial channel. The timer is reloaded with the 11-bit BG1.BG1_BR_VALUE stored in its reload register BG1 each time it underflows. The duration between underflows depends on the ‘n’ value in the fractional divider, which can be selected by the bits BGL1.BG1_FD_SEL. ‘n’ times out of 32, the timer counts one cycle more than specified by BG1.BG1_BR_VALUE. The prescaler is selected by the bits BCON1.BR1_PRE.

Register BG1 is the dual-function baud-rate generator/reload register. Reading BG1 returns the contents of the timer (not the configured reload value!), while writing to BG1 updates the reload register.

The register BG1 should be written only when BCON1.BR1_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 BCON1.BR1_R is set. Any write to BG1, while BCON1.BR1_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_{sys}
- Value of bit field BCON1.BR1_PRE
- Value of bit field BGL1.BG1_FD_SEL
- Value of the 11-bit reload value BG1.BG1_BR_VALUE

The following figure shows a simplified block diagram of the baud-rate generator.

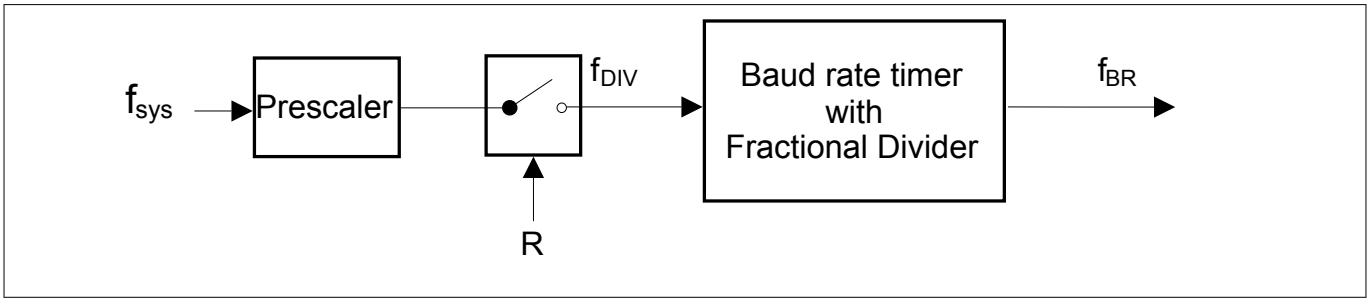


Figure 164 Simplified baud-rate generator block diagram

The following formula calculate the final baud-rate:

$$\text{Baud-rate} = \frac{f_{sys}}{16 \times PRE \times \left(BR\_VALUE + \frac{n}{32}\right)} \quad (11)$$

The value of PRE (prescaler) is chosen by the bit field BCON1.BR1_PRE. BR_VALUE represents the contents of the reload value, taken as unsigned 11-bit integer from the bit field BG1.BG1_BR_VALUE. n/32 is defined by the fractional divider selection in bit field BGL1.BG1_FD_SEL.

The maximum baud-rate that can be generated is limited to $f_{sys}/32$. Hence, for module clocks of 40 MHz and 25 MHz, the maximum achievable baud-rate is 1.25 MBaud and 0.78 MBaud respectively.

The following tables list various commonly used baud-rates together with their corresponding parameter settings and the deviation errors compared to the intended baud-rate.

Table 135 Typical baud-rates of UART (f_{sys} = 40 MHz)

Baud-rate (f_{sys} = 40 MHz)	PRE	Reload value (BR_VALUE)	Numerator of fractional value (FD_SEL)	Deviation error
250.4 kBaud	1 (BR1_PRE = 000)	9 (9_H)	31 ($1F_H$)	+0.15%
115.2 kBaud	1 (BR1_PRE = 000)	21 (15_H)	22 (16_H)	+0.06%
20 kBaud	1 (BR1_PRE = 000)	125 ($7D_H$)	0 (0_H)	0.00%
19.2 kBaud	1 (BR1_PRE = 000)	130 (82_H)	7 (7_H)	-0.01%
9600 Baud	2 (BR1_PRE = 001)	130 (82_H)	7 (7_H)	-0.01%
4800 Baud	4 (BR1_PRE = 010)	130 (82_H)	7 (7_H)	-0.01%
2400 Baud	8 (BR1_PRE = 011)	130 (82_H)	7 (7_H)	-0.01%

Table 136 Typical baud-rates of UART (f_{sys} = 25 MHz)

Baud-rate (f_{sys} = 25 MHz)	PRE	Reload value (BR_VALUE)	Numerator of fractional value (FD_SEL)	Deviation error
115.2 kBaud	1 (BR1_PRE = 000)	13 (D_H)	18 (12_H)	+0.01%
20 kBaud	1 (BR1_PRE = 000)	78 ($4E_H$)	4 (4_H)	0.00%
19.2 kBaud	1 (BR1_PRE = 000)	81 (51_H)	12 (C_H)	+0.01%
9600 Baud	2 (BR1_PRE= 001)	81 (51_H)	12 (C_H)	+0.01%
4800 Baud	4 (BR1_PRE = 010)	81 (51_H)	12 (C_H)	+0.01%
2400 Baud	8 (BR1_PRE = 011)	81 (51_H)	12 (C_H)	+0.01%

18.7 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 sync byte detection, provides the capability to detect the baud-rate within LIN protocol using timer 2/timer 21. This allows the UART module to be synchronized to the LIN baud-rate for data transmission and reception.

18.7.1 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 the 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 the following figure. The frame consists of:

- Header, which comprises a sync break (13-bit time low), sync byte (55_H), and ID field
- Response time
- Data bytes (according to UART protocol)
- Checksum

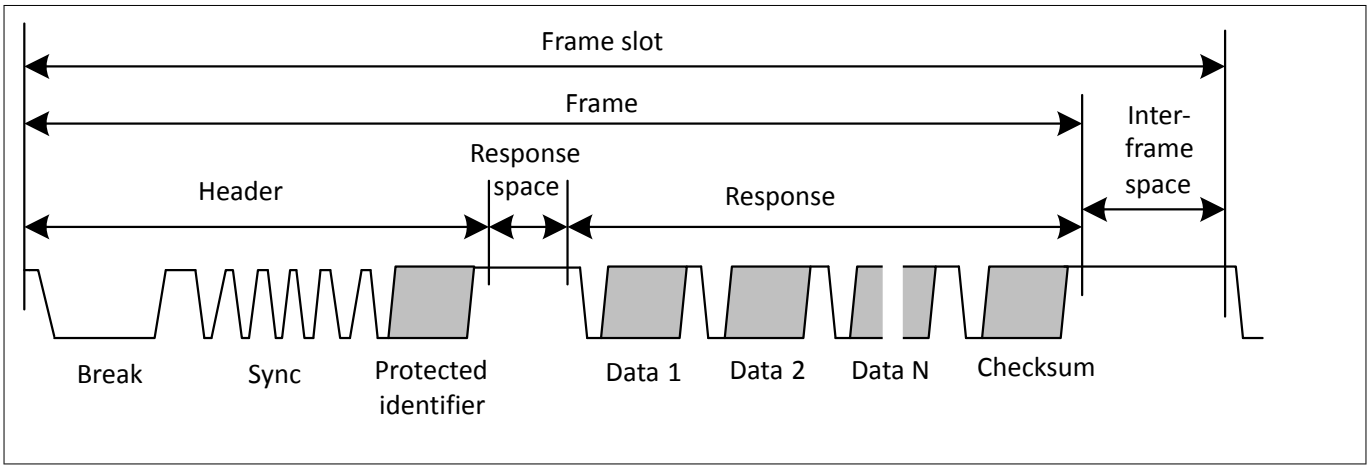


Figure 165 Structure of LIN frame

Each byte field is transmitted as a serial byte, as shown in the following figure. 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).

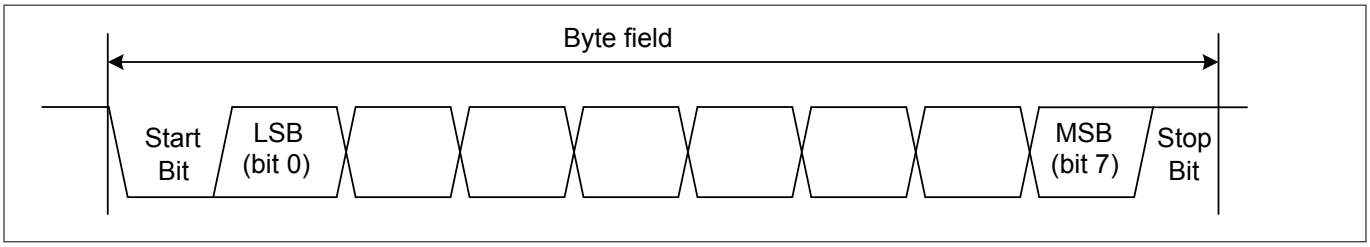


Figure 166 Structure of byte field

The sync break is used to signal the beginning of a new frame. It is the only field that does not comply with the previous figure. A sync 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 sync break delimiter, as shown in the following figure. The sync break delimiter will be at least one nominal bit time long.

A slave node will use a sync break detection threshold of 11 nominal bit times.

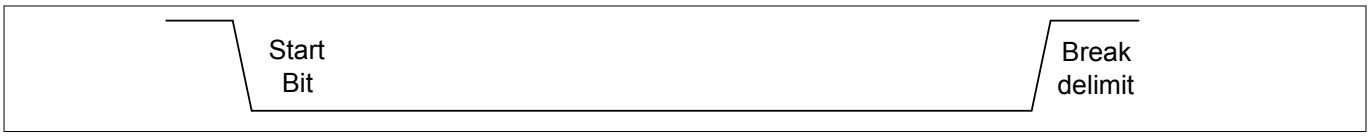


Figure 167 Sync break field

The sync byte is a specific pattern for the determination of the time base. The sync byte field consists of the data value 55_H, as shown in the following figure.

A slave task is always able to detect the sync break/sync sequence, even if it expects a byte field (assuming the byte fields are separated from each other). If this happens, detection of the sync break/sync sequence will abort the transfer in progress and processing of the new frame will commence.

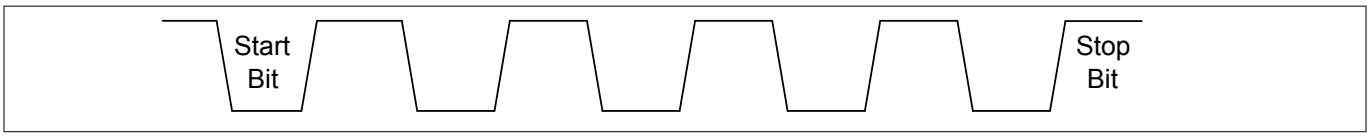


Figure 168 Sync byte field

The slave task will receive and transmit data when an appropriate ID is sent by the master:

1. The slave waits for the sync break
2. The slave synchronizes on the sync byte
3. The slave snoops for the ID
4. According to the ID, the 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 a check byte

18.7.2 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 part of the frame.

The header consists of a sync break and sync byte pattern followed by an identifier. Among these three fields, only the sync break pattern cannot be transmitted as a normal 8-bit UART data. The sync 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 the frame. For this purpose, every frame starts with a sequence consisting of a sync break followed by a sync byte field. This sequence is unique and provides enough information for any slave task to detect the beginning of a new frame and to be synchronized at the start of the identifier field.

18.7.3 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 the user software:

1. Initialize interface for reception and timer for baud-rate measurement
2. Wait for an incoming LIN frame from host
3. Synchronize the baud-rate to the host
4. Enter for master request frame or for slave response frame

The next sections, Chapter 18.7.4, Chapter 18.7.5 and Chapter 18.7.6, provide some hints on setting up the microcontroller for baud-rate detection of LIN.

Note: Re-synchronization and setup of the baud-rate has always to be done for every master request header or slave response header LIN frame by user software.

18.7.4 Initialization of break/sync field detection logic

The LIN baud-rate detection feature provides the capability to detect the baud-rate within the LIN protocol using timer 2/timer 21. Initialization consists of:

- Setting of the serial port of the microcontroller to Mode 1 (8-bit UART, variable baud-rate) for communication
- Providing the baud-rate range via bit field LINST.BGSEL
- Toggling of the LINST.BRDIS bit (set the bit to 1 before clearing it back to 0) to initialize the sync break/sync detection logic
- Clearing all status flags LINST.BRK, LINST.EOFSYN and LINST.ERRSYN to 0
- Setting of timer 2/timer 21 to capture mode with falling edge trigger at pin T2EX. Setting of the bits T2MOD.EDGESEL to 0 by default and T2CON.CP/$\overline{RL2}$ to 1
- Enabling timer 2 external events. T2CON. EXEN2 is set to 1. (EXF2 flag is set when a negative transition occurs at pin T2EX)
- *Configuring of* f_{T2} by bit field T2MOD.T2PRE.

18.7.5 Baud-rate range selection

The sync break/sync field detection logic supports a maximum number of bits in the sync break field as defined by the following equation.

$$\text{Maximum number of bits} = \text{Baud-rate} \times \frac{4095}{\text{Sample frequency}} \tag{12}$$

The sample frequency is given by the following equation.

$$\text{Sample frequency} = \frac{f_{sys}}{8 \times 2^{BGSEL}} \tag{13}$$

If the maximum number of bits in the break field is exceeded, the internal counter will overflow, which results in a baud-rate detection error. Therefore, an appropriate BGSEL value has to be selected for the required baud-rate detection range.

The baud-rate range defined by different BGSEL settings is shown in the following table.

Table 137 BGSEL bit field definition for different input frequencies

f_{sys}	BGSEL	Baud-rate select for detection f_{sys} / (2184 × 2^{BGSEL}) to f_{sys} / (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
25 MHz	00_B	11.2 kHz to 347.2 kHz
	01_B	5.7 kHz to 173.6 kHz
	10_B	2.9 kHz to 86.8 kHz
	11_B	1.4 kHz to 43.4 kHz

Each BGSEL setting supports a range of baud-rate for detection. If the baud-rate used is outside the defined range, the baud-rate may not be detected correctly.

When f_{sys} = 40 MHz, the baud rate range between 18.3 kHz to 555.6 kHz can be detected. The following examples serve as a guide to select the BGSEL value:

- If the baud-rate falls in the range of 2.3 kHz to 4.6 kHz, selected BGSEL value is "11_B"
- If the baud-rate falls in the range of 4.6 kHz to 9.2 kHz, selected BGSEL value is "10_B"
- If the baud-rate falls in the range of 9.2 kHz to 18.3 kHz, selected BGSEL value is "01_B"
- If the baud-rate falls in the range of 18.3 kHz to 555.6 kHz, selected BGSEL value is "00_B"
- If the baud-rate is 20 kHz, the possible values of BGSEL that can be selected are "00_B", "01_B", "10_B", and "11_B". However, it is advisable to select "00_B" for better detection accuracy

The baud-rate can also be detected when f_{sys} = 25 MHz, for which the baud-rate range that can be detected is between 1.4 kHz to 347.2 kHz.

In general, BGSEL should be selected as small as possible, to achieve the best possible accuracy.

18.7.6 LIN baud-rate detection

The baud-rate detection for LIN is shown in the following figure. The header LIN frame consists of the:

- Sync break (13-bit times low)
- Sync byte (55_H)
- Protected ID field

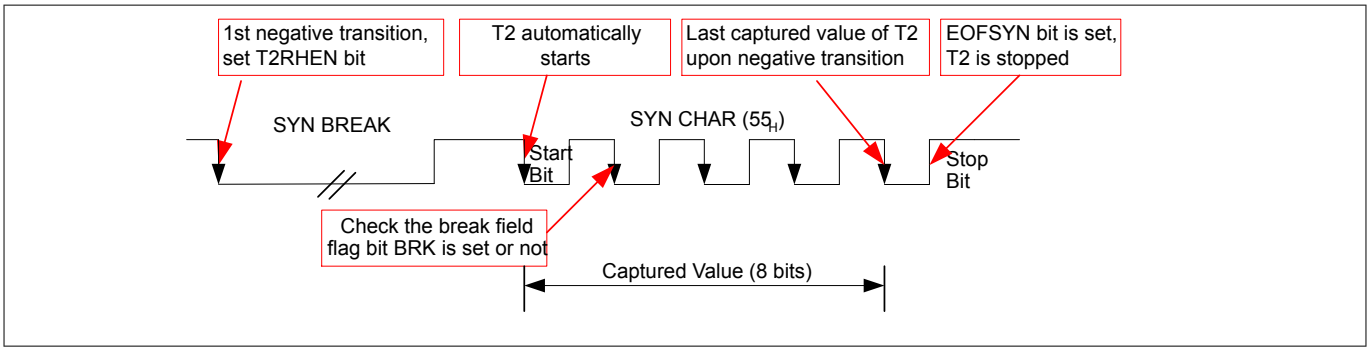


Figure 169 LIN auto baud-rate detection

With the first falling edge:

- The timer 2 external start enable bit (T2MOD.T2RHEN) is set. The falling edge at pin T2EX is selected by default for timer 2 external start (bit T2MOD.T2REGS is 0)

With the second falling edge:

- Start timer 2 by the hardware

With the third falling edge:

- Timer 2 captures the timing of 2 bits of SYN byte
- Check the break field flag bit LINST.BRK

If the sync break field flag LINST.BRK is set, software may continue to capture 4/6/8 bits of sync byte. Finally, the end of sync byte flag (LINST.EOFSYN) is set, timer 2 is stopped. T2 reload/capture register (RC) is the time taken for 2/4/6/8 bits according to the implementation. Then the LIN routine calculates the actual baud-rate, sets the BCON1.BR1_PRE and BG1 values if the UART module uses the baud-rate generator for baud rate generation.

After the third falling edge, the software may discard the current operation and continue to detect the next header LIN frame if the following conditions were detected:

- The sync break field flag LINST.BRK is not set, or
- The sync byte error flag LINST.ERRSYN is set

18.8 UART1/UART2 (UART) register definition

Note: The register names used here refer to UART1. For UART2, the register names need to be adapted accordingly.

UART uses the special function registers (SFRs) UART_SCON, UART_SBUF, SCU_BCON1, SCU_LINST and SCU_BG1.

UART_SCON is the control register and UART_SBUF is the data register.

The serial port control and status register is the UART_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).

UART_SBUF is the receive and transmit buffer of the serial interface. Writing to UART_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 UART_SBUF accesses a physically separate receive register.

The registers are addressed wordwise.

18.8.1 Register address space - UART

Table 138 Registers address space - UART

Module	Base address	End address	Note
UART1	48020000_H	$48021FFF_H$	UART1 registers
UART2	48022000_H	$48023FFF_H$	UART2 registers

18.8.2 Register overview - UART (ascending offset address)

Table 139 Register overview - UART (ascending offset address)

Short name	Long name	Offset address	Page number
UART_SCON	Serial channel control register	0000_H	616
UART_SBUF	Serial data buffer register	0004_H	618
UART_SCONCLR	Serial channel control clear register	0008_H	619

18.8.3 Serial channel control register

UART_SCON Offset address: 0000_H

Serial channel control register RESET_TYPE_3 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								SM0	SM1	SM2	REN	TB8	RB8	TI	RI
r								rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
RI	0	rw	**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. Must be cleared by flag SCONCLR.RICLR. This flag can also be set by software.
TI	1	rw	**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. Must be cleared by flag SCONCLR.TICLR. This flag can also be set by software.
RB8	2	rw	**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. Must be cleared by flag SCONCLR.RB8CLR. This flag can also be set by software.
TB8	3	rw	**Serial port transmitter bit 9** In modes 2 and 3, this is the 9th data bit sent. In mode 1, this bit is set to 1 In mode 0, this bit is set to 1
REN	4	rw	**Enable receiver of serial port** 0_B **DISABLE**: Serial reception is disabled 1_B **ENABLE**: Serial reception is enabled
SM2	5	rw	**Enable serial port multiprocessor communication in modes 2 and 3** Mode 2 or 3: - if SM2 = 1: RI will not be activated if the received 9th data bit (RB8) is 0. Mode 1: - if SM2 = 1: RI will not be activated if no valid stop bit (RB8) was received. Mode 0: - SM2 should be 0

(table continues...)

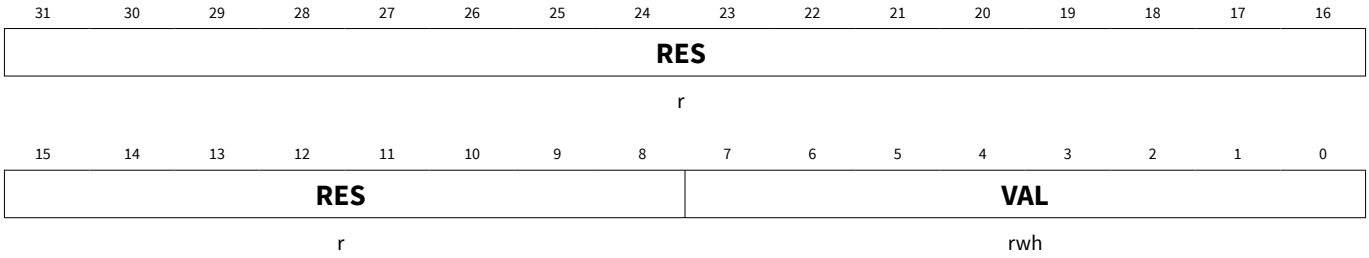
(continued)

Field	Bits	Type	Description
SM1	6	rw	**Serial port operating mode selection** See Table "UART modes".
SM0	7	rw	**Serial port operating mode selection** See Table "UART modes".
RES	31:8	r	**Reserved** Returns 0 if read. Should be written with 0.

18.8.4 Serial data buffer register

UART_SBUF Offset address: 0004_H

Serial data buffer register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
VAL	7:0	rwh	**Serial interface buffer register**
RES	31:8	r	**Reserved** Returns 0 if read. Should be written with 0.

18.8.5 Serial channel control clear register

UART_SCONCLR

Offset address: 0008_H

Serial channel control clear register

RESET_TYPE_3 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													**RB8CLR**	**TICLR**	**RICLR**
r													w	w	w

Field	Bits	Type	Description
RICLR	0	w	**SCON.RI clear flag** Flag is always read as 0. 0_B **Not_cleared**: RI flag is not cleared 1_B **Cleared**: RI flag is cleared
TICLR	1	w	**SCON.TI clear flag** Flag is always read as 0. 0_B **Not_cleared**: TI flag is not cleared 1_B **Cleared**: TI flag is cleared
RB8CLR	2	w	**SCON.RB8 clear flag** Flag is always read as 0. 0_B **Not_cleared**: RB8 flag is not cleared 1_B **Cleared**: RB8 flag is cleared
RES	31:3	r	**Reserved** Returns 0 if read. Should be written with 0.

18.8.6 Baud-rate generator control and status registers

The UART module is also used to support LIN communication. For this purpose the UART is equipped with a LIN break recognition, sync byte detector and special baud-rate generator including a fractional divider. The control registers for this support hardware are located in the System control unit - digital modules (SCU-DM) module.

18.9 Interfaces of the UART module

Figure 170 and Figure 171 are showing an overview of the UART I/O interfaces.

In mode 0 the serial port behaves as an 8-bit shift register (UART2 only). Data are shifted in through RXD and out through RXDO. The TXD line is used to provide a shift clock that can be used by external devices to clock data.

In mode 1, 2 and 3 the port behaves as a standard UART. Data are shifted in/out via RXD/TXD.

The UART1 module offers the possibility to process data coming from different sources. The RXD input selection is performed by the SCU via SFR bit MODPISEL.URIOS1.

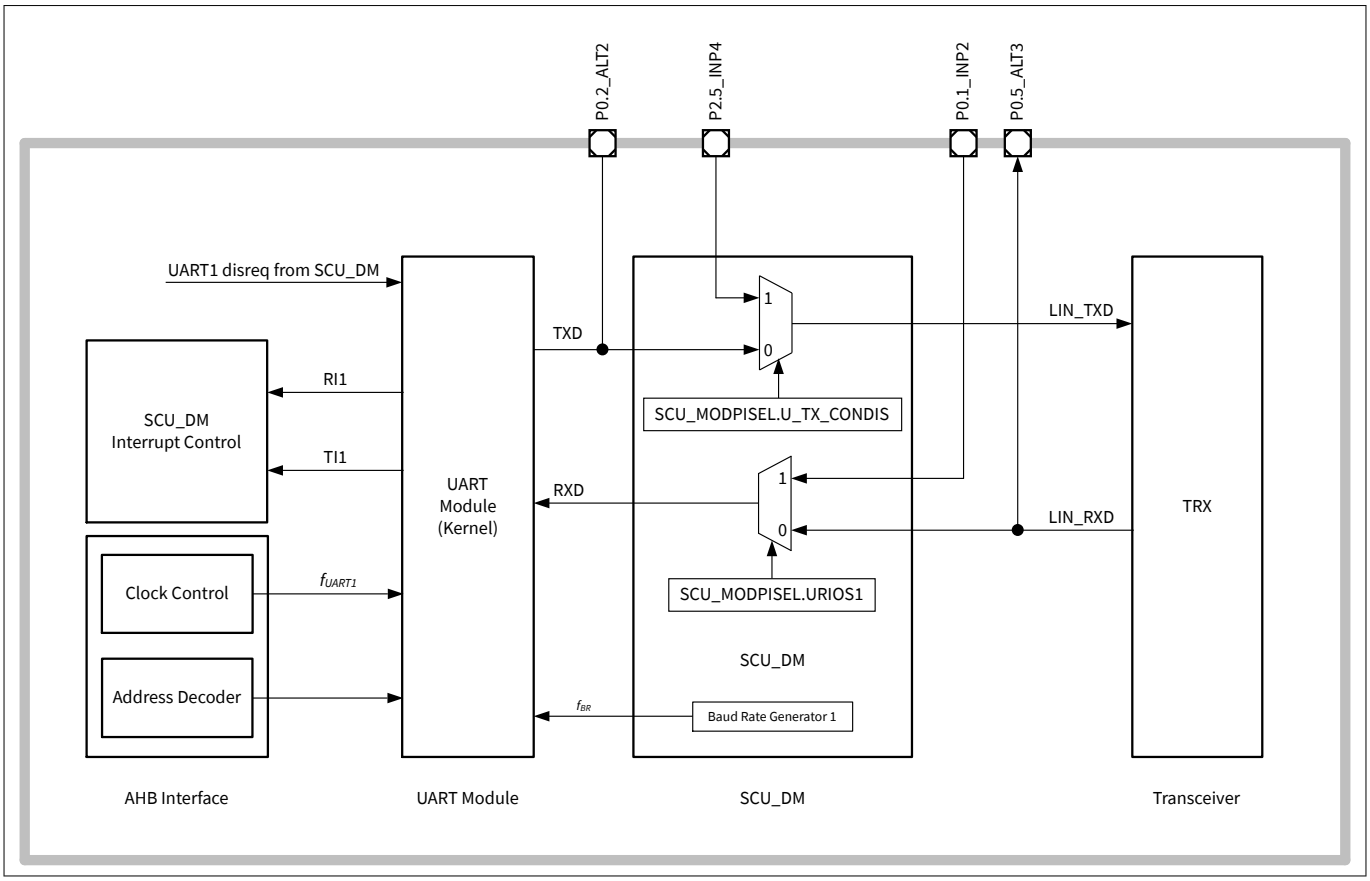


Figure 170 UART1 module I/O interface

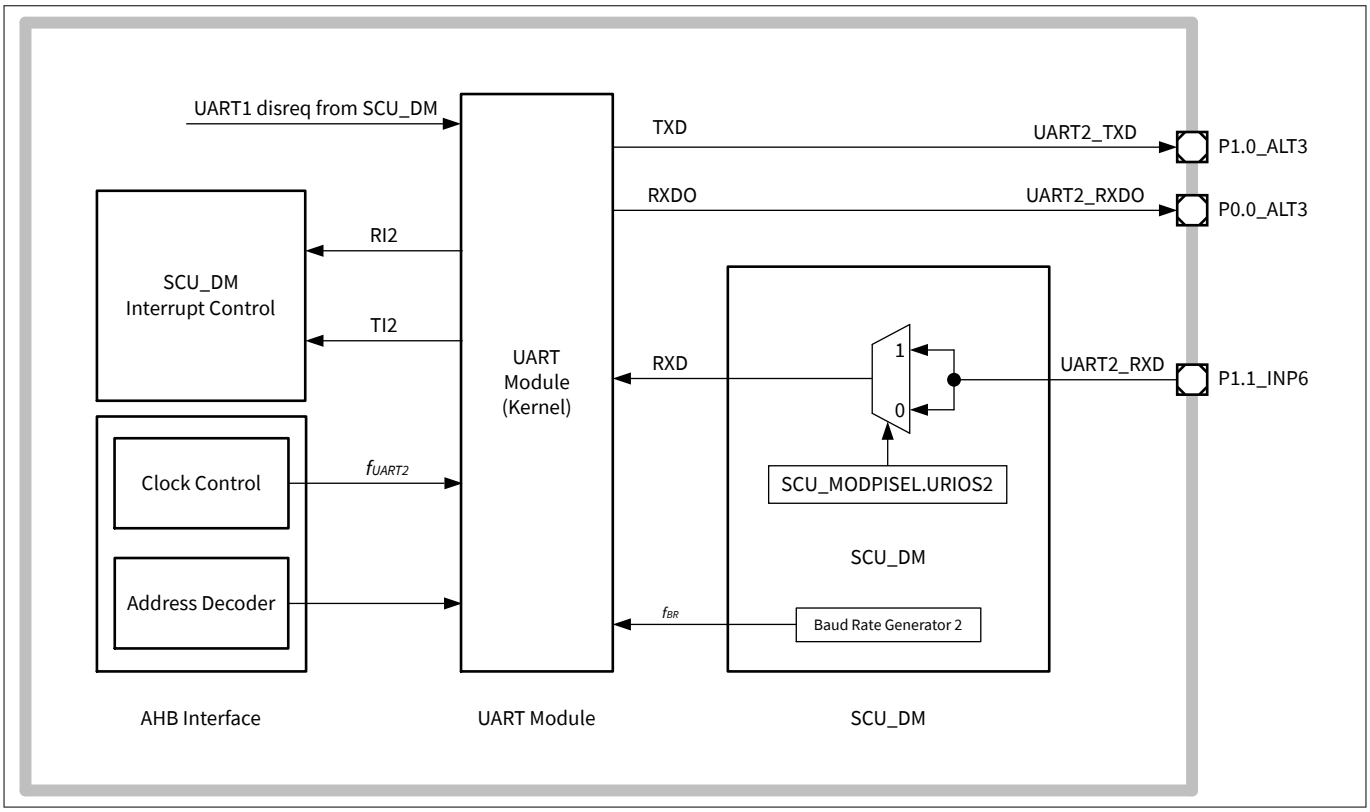


Figure 171 UART2 module I/O interface

19 LIN transceiver

19.1 Features

19.1.1 General functional features

- LIN transceiver compliant to LIN2.2 standard, backward compatible to LIN2.1, LIN2.0 and LIN1.3
- LIN transceiver compliant to SAE-J2602 (slew-rate, receiver hysteresis)
- LIN communication supported via UART1 interface

19.1.2 Mode of operation

- Transceiver Normal mode
- Transceiver Receive-Only mode
- Transceiver Sleep (wake-capable) mode
- Transceiver Off mode

19.1.3 Special features

- LIN baud-rate measurement via Timer2
- Dominant TXD timeout feature.
- Transceiver port can be configured as standard HV I/O (LHVIO) via SFR
- Transceiver port overcurrent limitation and overtemperature protection
- Transceiver fully resettable via enable bit

19.1.4 Slope mode features

- LIN normal slope mode (up to 20 kbits/s)
- LIN low slope mode (up to 10.4 kbits/s)
- LIN fast slope mode (up to 62.5 kbits/s)
- LIN flash mode (up to 115 kbits/s or 250 kbits/s)

19.1.5 Wake-up features

- LIN network wake-up

19.2 Introduction

The transceiver supports the Local Interconnect Network (LIN) compliant to the LIN2.2 standard, backwards compatible to LIN1.3, LIN2.0 and LIN2.1. The transceiver operates as a bus driver located in between the protocol controller and the physical network. The LIN network is a single wire, bi-directional bus featuring baud-rates ranging from 2.4 kBaud to 20 kBaud. Additional baud-rates up to 62.5 kBaud are implemented.

The integrated slope control allows to use several data transmission rates with optimized EMC performance. For data transfer at the end-of-line, a Flash mode featuring up to 115 kBaud is implemented.

The transceiver converts the data stream on the TXD input into a bus signal using a current-limited wave-shaping driver which reduces electromagnetic emissions (EME). The receiver converts the data stream to logic level signals that are sent out via the RXD output.

Ultra-low power consumption is possible using the Sleep (wake-capable) mode which allows wake-up via the communication network.

Furthermore, the transceiver can be used as a high voltage input/output (LHVIO) controlled by SFR bits (see LIN_CTRL.TXD and LIN_CTRL.RXD bit description).

19.2.1 Block diagram

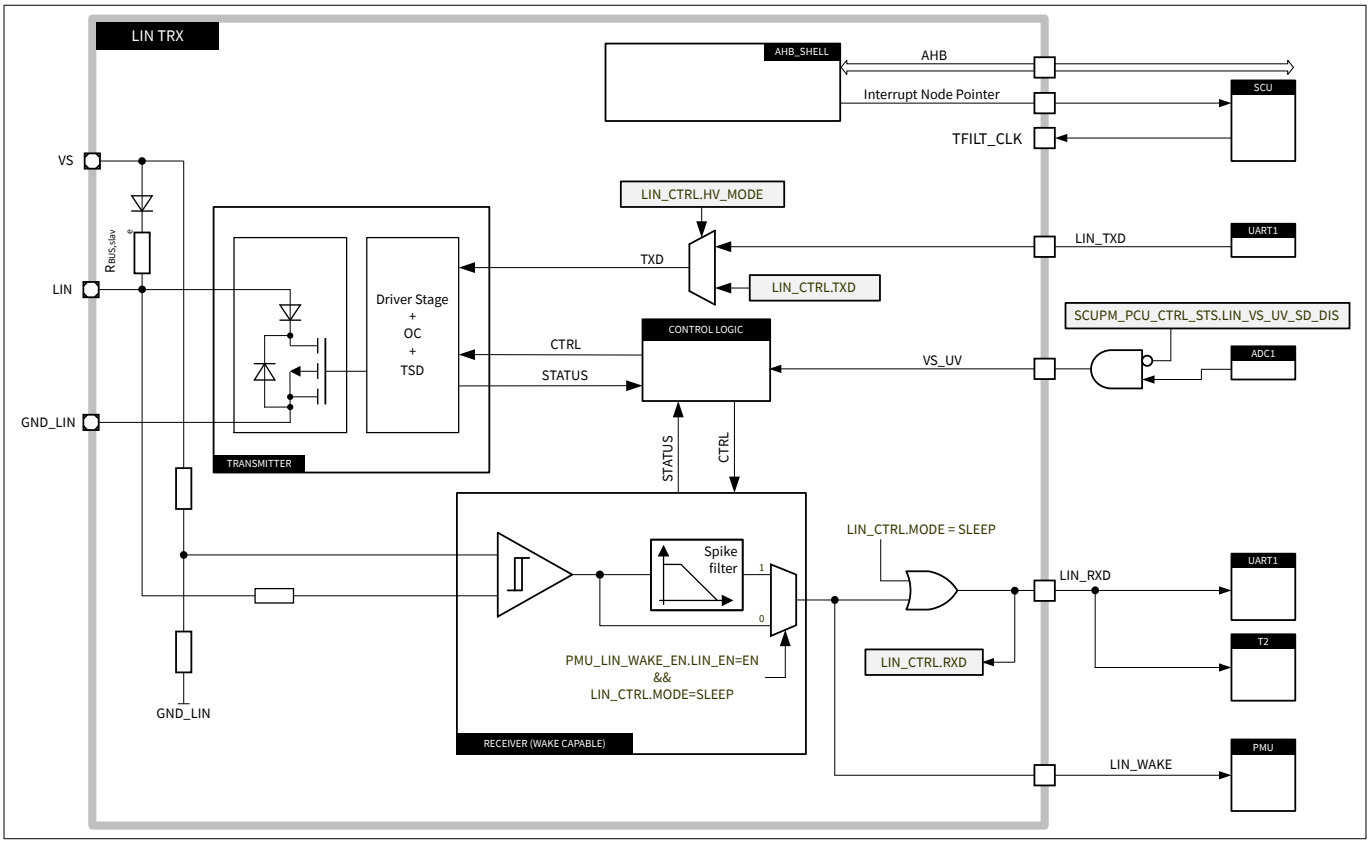


Figure 172 LIN transceiver block diagram

19.3 Functional description

The integrated transceiver supports the following operating modes:

- Normal mode
- Receive-Only mode
- Sleep (wake-capable) mode
- Off mode

The transceiver module is controlled by an internal state machine which determines the actual state of the transceiver. This state machine is controlled via SFRs. The following diagram shows the different operating modes:

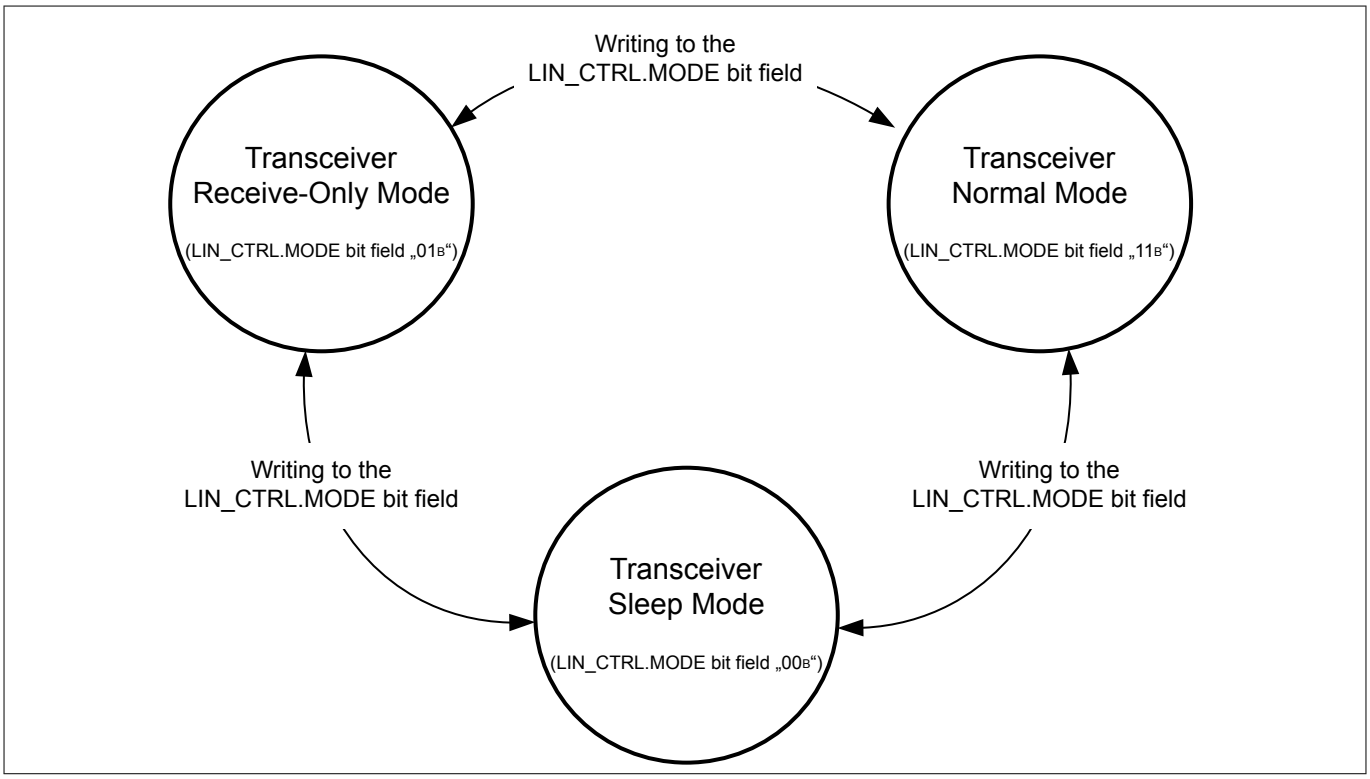


Figure 173 Transceiver state-machine

Table 140 Transceiver operating modes

LIN_CTRL.MODE	PMU_LIN_WAKE_EN.LIN_EN	Description
00	0	Transceiver Sleep mode
00	1	Transceiver Sleep mode (wake-capable)
01	X	Transceiver Receive-Only mode
11	X	Transceiver Normal mode

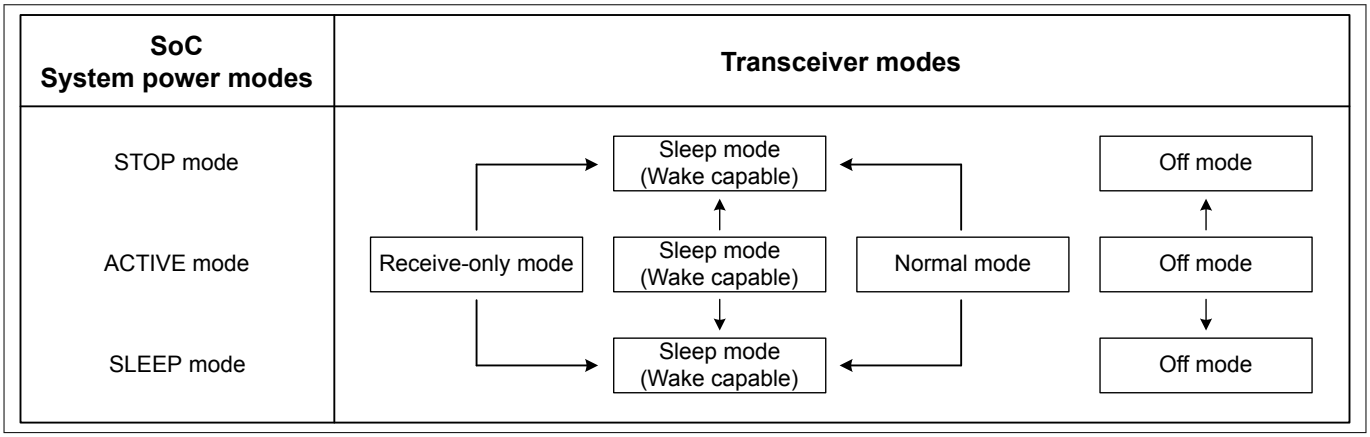


Figure 174 Transceiver operating modes vs. SoC system power states (PMU_LIN_WAKE_EN.LIN_EN bit = 1)

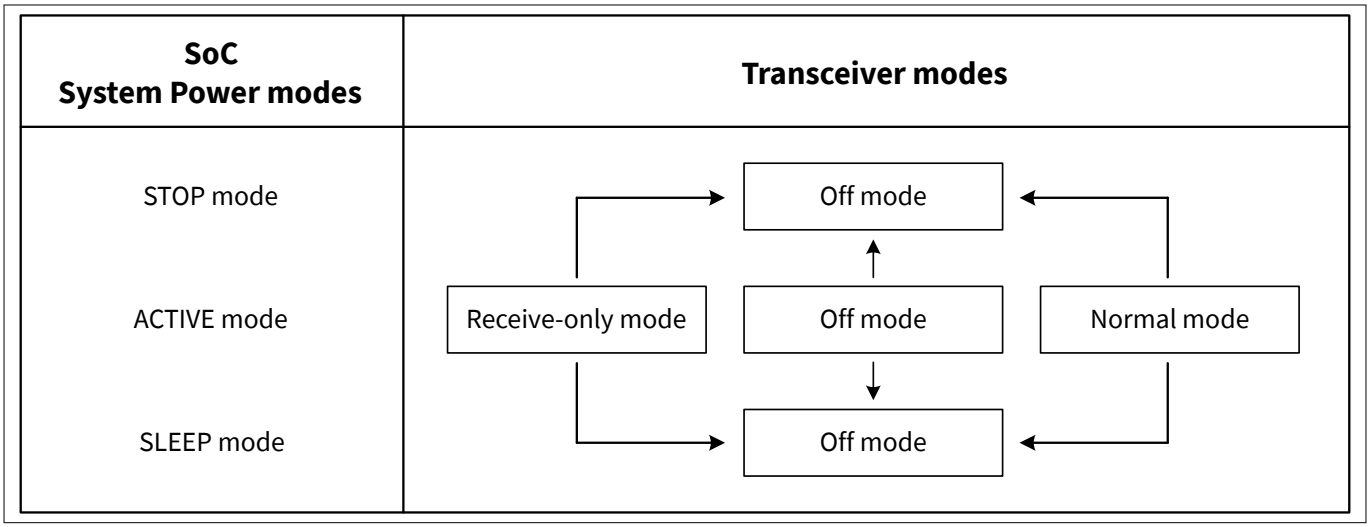


Figure 175 Transceiver operating modes vs. SoC system power states (PMU_LIN_WAKE_EN.LIN_EN bit = 0)

19.3.1 Modes of operation

19.3.1.1 Normal mode

In Normal mode, the device is fully functional and all functions of the transceiver are available. Data can be received/transmitted from/to the network. The following functions are available in the Normal mode:

- The transmitter is active. Data on the TXD line are driven to the network.
- The receiver is active. Data on the network are transferred to the RXD output
- LIN communication is supported with normal (up to 20 kBaud), low (up to 10.4 kBaud), fast slopes (up to 40 kBaud) and flash mode (up to 115 kBaud). The transmitter slope settings are locked in Normal mode
- The transmit path (TXD) provides a dominant timeout function
- The local overtemperature protection is enabled
- The V_S undervoltage detection (via ADC1) is enabled

The following conditions need to be satisfied to enter Normal mode.

- Normal mode is selected (LIN_CTRL.MODE bit field "11_B")
- LHVIO operation is disabled (LIN_CTRL.HV_MODE bit field "0_B")

- Mode feedback status bits indicates a proper operation (LIN_CTRL.MODE_FB)
- Transceiver slope status bits indicates a proper operation (LIN_CTRL.FB_SM[3:1])
- No V_S undervoltage event (SCUPM_SYS_SUPPLY_IRQ_STS.VS_UV_STS bit is reset)
- No overtemperature event occured (LIN_IRQS.OT_STS bit is reset)
- No overcurrent event occured (LIN_IRQS.OC_IS bit is reset)
- No TXD dominant timeout occured (LIN_IRQS.TXD_TMOUT_STS bit is reset)

19.3.1.2 Receive-Only mode

In Receive-Only mode, data can only be received from the communication network. The following functions are available in Receive-Only mode:

- The transmitter is deactivated. Data on the TXD line are not driven to the network
- The receiver is active. Data on the network are transferred to the RXD output
- The local overtemperature protection is disabled
- The V_S undervoltage detection (via ADC1) is enabled

Receive-Only mode can be entered:

- Via a software command (refer to LIN_CTRL register description)
- Automatically upon error detection in Normal mode

The following conditions need to be satisfied to transition from Sleep mode into Receive-Only mode:

- Receive-Only mode is selected (LIN_CTRL.MODE bit field "01_B")
- LHVIO operation is disabled (LIN_CTRL.HV_MODE bit field "0_B")
- Mode feedback status bits indicates a proper operation (LIN_CTRL.MODE_FB)
- Transceiver slope status bits indicates a proper operation (LIN_CTRL.FB_SM[3:1])

The transceiver transitions from Normal mode into Receive-Only mode should one or more of the following conditions be satisfied:

- Receive-Only mode is selected (LIN_CTRL.MODE bit field "01_B")
- Mode feedback status bits indicates an error (LIN_CTRL.MODE_FB)
- Transceiver slope status bits indicates an error (LIN_CTRL.FB_SM[3:1])
- V_S undervoltage event occured (SCUPM_SYS_SUPPLY_IRQ_STS.VS_UV_STS bit is set)
- Local overtemperature event occured (LIN_IRQS.OT_STS bit is set)
- Overcurrent event occured (LIN_IRQS.OC_IS bit is set)
- TXD dominant timeout error (LIN_IRQS.TXD_TMOUT_STS bit is set)

19.3.1.3 Sleep (wake-capable) mode

Sleep (wake-capable) mode is a low power mode with reduced quiescent current consumption. In this mode the transceiver can detect a wake-up pattern on the communication network and to wake-up the MCU subsystem. The following functions are available in Sleep (wake-capable) mode:

- The transmitter is disabled
- The receiver is disabled
- The wake receiver is enabled
- The communication network is continuously monitored for a valid wake-up event
- The TXD timeout function is disabled
- The local overtemperature protection is disabled
- The V_S undervoltage detection (via ADC1) is disabled

The following conditions need to be satisfied to enter Sleep (wake-capable) mode:

- Sleep (wake-capable) is selected (LIN_CTRL.MODE bit field "00_B")

The MCU subsystem wake-up scheme can be configured in the PMU module. SoC wake-ups triggered by the communication network can be enabled/disabled in the PMU module.

19.3.1.4 High voltage input/output (LHVIO)

The transceiver can be configured as a high voltage input/output port controlled by the LIN_CTRL.TXD and LIN_CTRL.RXD bit fields.

Programming sequence to enter High Voltage I/O mode:

- Transceiver must be disabled and in Sleep mode (refer to LIN_CTRL.MODE bit field description)
- High Voltage I/O mode must be enabled (LIN_CTRL.HV_MODE bit field set to "1_B")
- Transceiver must be enabled in Normal mode (refer to LIN_CTRL.MODE and LIN_CTRL.EN bit field description)

Additionaly the following conditions need to be satisfied to enter High Voltage I/O mode:

- Mode feedback status bits indicates a proper operation (LIN_CTRL.MODE_FB)
- Transceiver slope status bits indicates a proper operation (LIN_CTRL.FB_SM[3:1])
- No overtemperature event occured (LIN_IRQS.OT_STS bit is reset)
- No overcurrent event occured (LIN_IRQS.OC_IS bit is reset)

Programming sequence to exit High Voltage I/O mode:

- Transceiver must be disabled and in Sleep mode (refer to LIN_CTRL.MODE bit field description)
- High Voltage I/O mode must be disabled (LIN_CTRL.HV_MODE bit field set to "0_B")

High-Voltage IO mode provides error handling for the following cases. If a malfunction is detected, the module transitions automatically in Sleep mode. The LIN_CTRL.MODE_FB bits reflect the error source(s).

- Mode feedback status bits indicates an error (LIN_CTRL.MODE_FB)
- Transceiver slope status bits indicates an error (LIN_CTRL.FB_SM[3:1])
- An overtemperature event occured (LIN_IRQS.OT_STS bit is set)
- An overcurrent event occured (LIN_IRQS.OC_IS bit is set)

19.3.1.5 Wake-up from network

There are two ways to wake-up the transceiver from Sleep (wake-capable) mode:

- Operating mode change via LIN_CTRL.MODE bits.
- Remote wake up initiated by the falling edge of a recessive (high) to dominant (low) state transition on the communication network, where the dominant state is to be held for t_{BUS} filter time. After this t_{BUS} filter time has been met, the transceiver indicates the detection of a wake-up event by issuing a signal (LIN_WAKE) to the PMU. For more details, please refer to the SLEEP/STOP mode and Wake-up Management Unit (WMU) description in the PMU section. At this stage, the MCU subsystem (supplemented by the Timer2 module) is responsible for validating the wake-up pattern.

Transceiver network configuration	**t_{BUS}**
LIN	refer to $t_{WK,bus}$

19.3.2 Fail-safe functions

The transceiver provides error handling for three different cases.

19.3.2.1 TXD dominant timeout

While operating in normal mode, a permanent dominant level (i.e. $t > t_{timeout}$) on the TXD line would block the communication because the network would be held in dominant state.

In order to prevent this scenario the transmitter disconnects the TXD input from the bus driver after a certain time and keeps the transmitter recessive by entering the Receive-Only mode. The other nodes on the network would still be able to communicate, and the receiver could continue to listen to the bus traffic.

The transmitter stage is activated again after the dominant timeout condition is removed and the LIN_IRQS.TXD_TMOUT_STS status flag cleared.

Status information can be gathered via the LIN_IRQS.TXD_TMOUT_STS status flag.

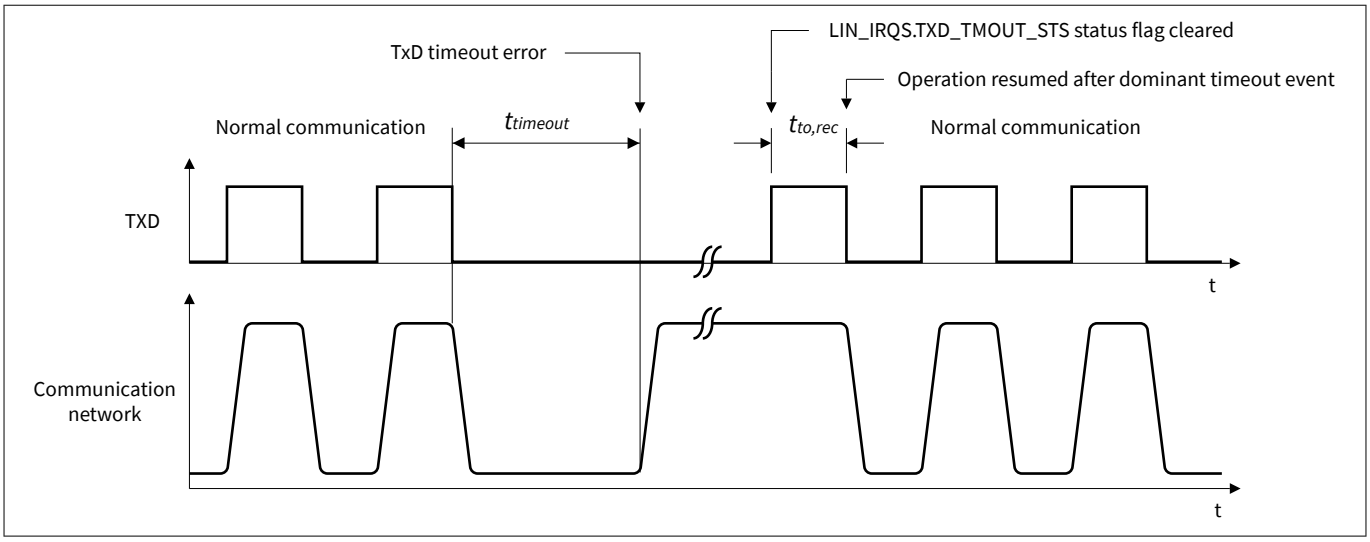


Figure 176 TXD dominant timeout behavior (LIN communication example)

19.3.2.2 Overcurrent protection

If the transmitter detects an overcurrent condition ($I > I_{BUS,sc}$), the transceiver enters Receive-Only mode and the LIN_IRQS.OC_IS flag is set. The short-circuit current is limited to $I_{BUS,sc}$. The LIN_IRQS.OC_IS flag can be cleared by software and will be set again as long as the overcurrent condition remains.

To register the occurrence of an overcurrent event, the transceiver sets an Interrupt Status bit (LIN_IRQS.OC_IS). To enable an interrupt source, set high the Interrupt Enable bit (LIN_IRQEN.OC_IEN). The interrupt events are propagated to the SCU for further processing and distribution. This interrupt is routed to INTISR_10.

19.3.2.3 Overtemperature protection

The transceiver is protected against overtemperature conditions.

In case of an overtemperature event ($T_J > T_{jSD}$) the transceiver enters Receive-Only mode (transmitter is disabled, i.e. recessive state) and the LIN_IRQS.OT_STS status flag is set. This operating mode is reflected in the LIN_CTRL.MODE_FB bit field. To resume data transmission the transmitter needs to cool down and the LIN_IRQS.OT_STS status bit must be reset.

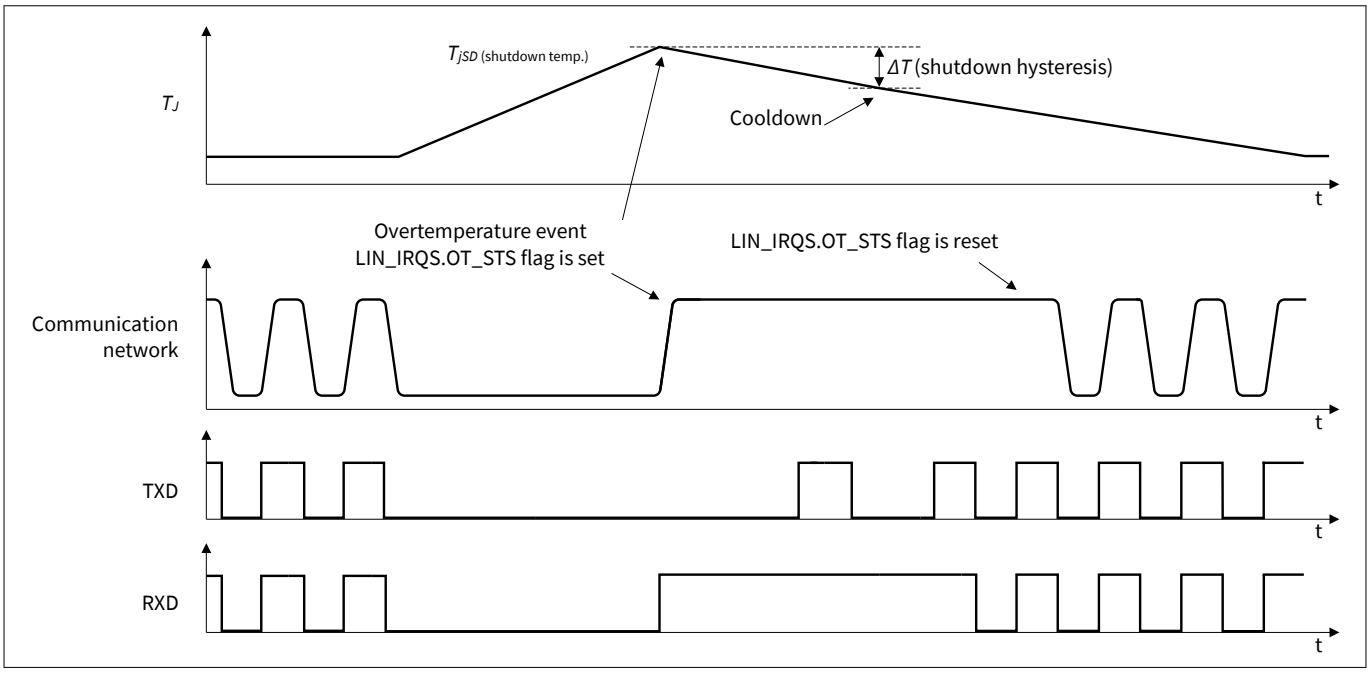


Figure 177 Overtemperature behavior (LIN communication example)

To register the occurrence of an overtemperature event, the transceiver sets an Interrupt Status bit (LIN_IRQS.OT_IS). To enable an interrupt source, set high the Interrupt Enable bit (LIN_IRQEN.OT_IEN). The interrupt events are propagated to the SCU for further processing and distribution. This interrupt is routed to INTISR_10.

19.3.3 Transceiver slope modes

The transceiver module provides additional slope mode control features intended to be used for EoL (End of Line) programming or to reduce emissions in case of low baud-rate transmission. The configurable slope modes are:

- Low slope for up to 10.4 kBaud transmission
- Normal slope for up to 20 kBaud transmission
- Fast slope for up to 40 kBaud transmission
- Flash mode for up to 115 kBaud transmission

Notice that slope mode change is prohibited in Normal Slope mode operation.

Normal Slope mode

This mode is usually used to transmit and receive LIN messages on the network. The selected slew-rate setting allows a transmission rate of up to 20 kBaud.

Low Slope mode

The usage of this mode is linked to a LIN communication at lower baud-rates. 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 kBaud.

Fast Slope mode

In this mode it is also possible to transmit and receive messages on the LIN network. The selected slew-rate setting allows a transmission rate of up to 40 kBaud.

Flash mode

In this mode it is possible to transmit and receive messages on LIN network. Transmission rates of up to 115 kBaud are possible. This mode can be used for EoL programming.

Slope mode change

Slope mode change is controlled via the LIN_CTRL.SM bits is prohibited while the transceiver in operating in Normal mode. This avoids possible transmission errors that may occur while performing an '*on-the-fly*' slope mode change. To change slope mode (e.g. from Normal Slope mode to Flash mode), it is necessary to transition the transceiver into Receive-Only mode or Sleep mode prior to configuring the desired slope mode. Once the slope mode is set, it is necessary to transition into Normal mode in order to communicate with the network.

19.3.4 Transceiver error handling

The LIN_MODE_FB bits supervise the correct combination of LIN_MODE according to the below table.

LIN_CTRL.MODE_FB[2:0]	Comments
000	Mode error
001	Sleep mode
010	Mode error
011	Mode error
100	Mode error
101	Receive-Only mode
110	Mode error
111	Normal mode

A mode error indicates a problem in the transceiver configuration. Check the transceiver configuration bit settings in case of a mode error. Should the error persists it is recommended to enter Sleep mode.

The transceiver provides the possibility to monitor the on-chip status of the Slope mode control through internally generated signals. The LIN_CTRL.FB_SM[3:1] bits indicate the correct combination of the slope modes according to the below table.

LIN_CTRL.FB_SM[3:1]	Comments
000	Transceiver not enabled
001	Low slope mode
010	Normal slope mode
011	Fast slope mode
100	Flash mode
101	Slope mode error
110	Slope mode error
111	Slope mode error

19.3.5 Interrupts

The transceiver module provides one interrupt node (NVIC node 10) to signal the following events (see LIN_IRQS register description):

- Overcurrent in the output stage
- Overtemperature in the transceiver
- TXD dominant timeout
- Slope mode status error

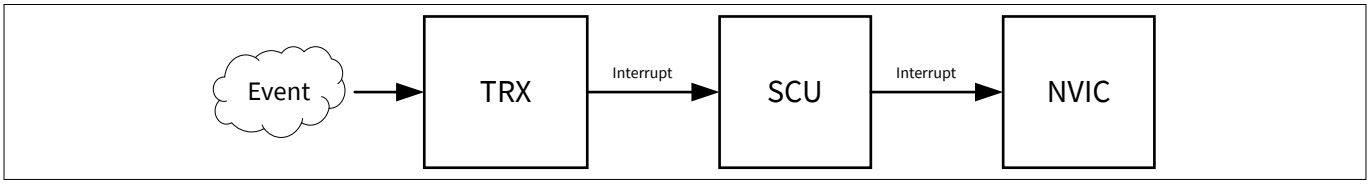


Figure 178 Event handling

To register the occurrence of an event, the transceiver module sets an Interrupt Status bit (xxx_IS). To enable an interrupt source, set high the corresponding Interrupt Enable bit (xxx_IEN). The interrupt events are propagated to the SCU for further processing and distribution. Interrupts will be generated (according to the event source) regardless of the Interrupt Status bit (xxx_IS). Conversely, an interrupt source can be masked by setting low the corresponding Interrupt Enable bit (xxx_IEN).

To clear an Interrupt Status bit (xxx_IS), set high the corresponding Interrupt Status Clear bit (xxx_ISC).

In addition to the interrupt scheme, the transceiver module also provides event status bits (xxx_STS) indicating the occurrence of a trigger event (e.g. transceiver module overtemperature).

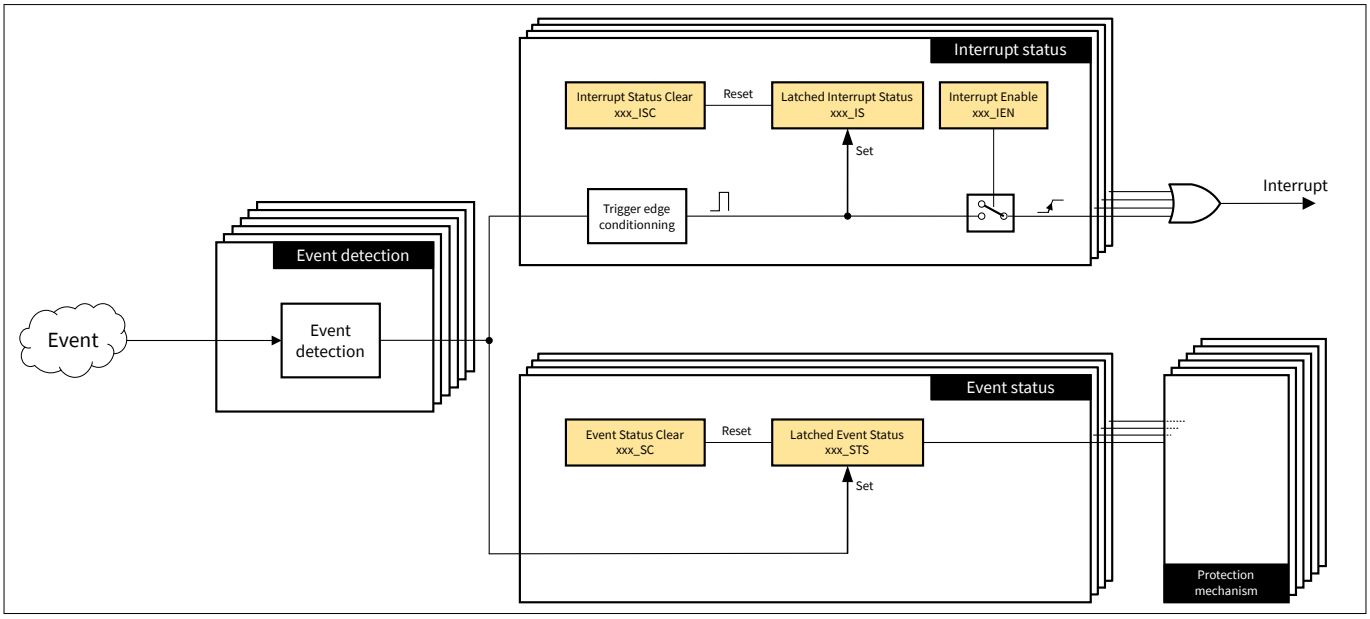


Figure 179 Interrupt and status registers

19.3.6 Interconnect TRX, UART1, TIMER2, GPIO, CCU6, SCU, PMU

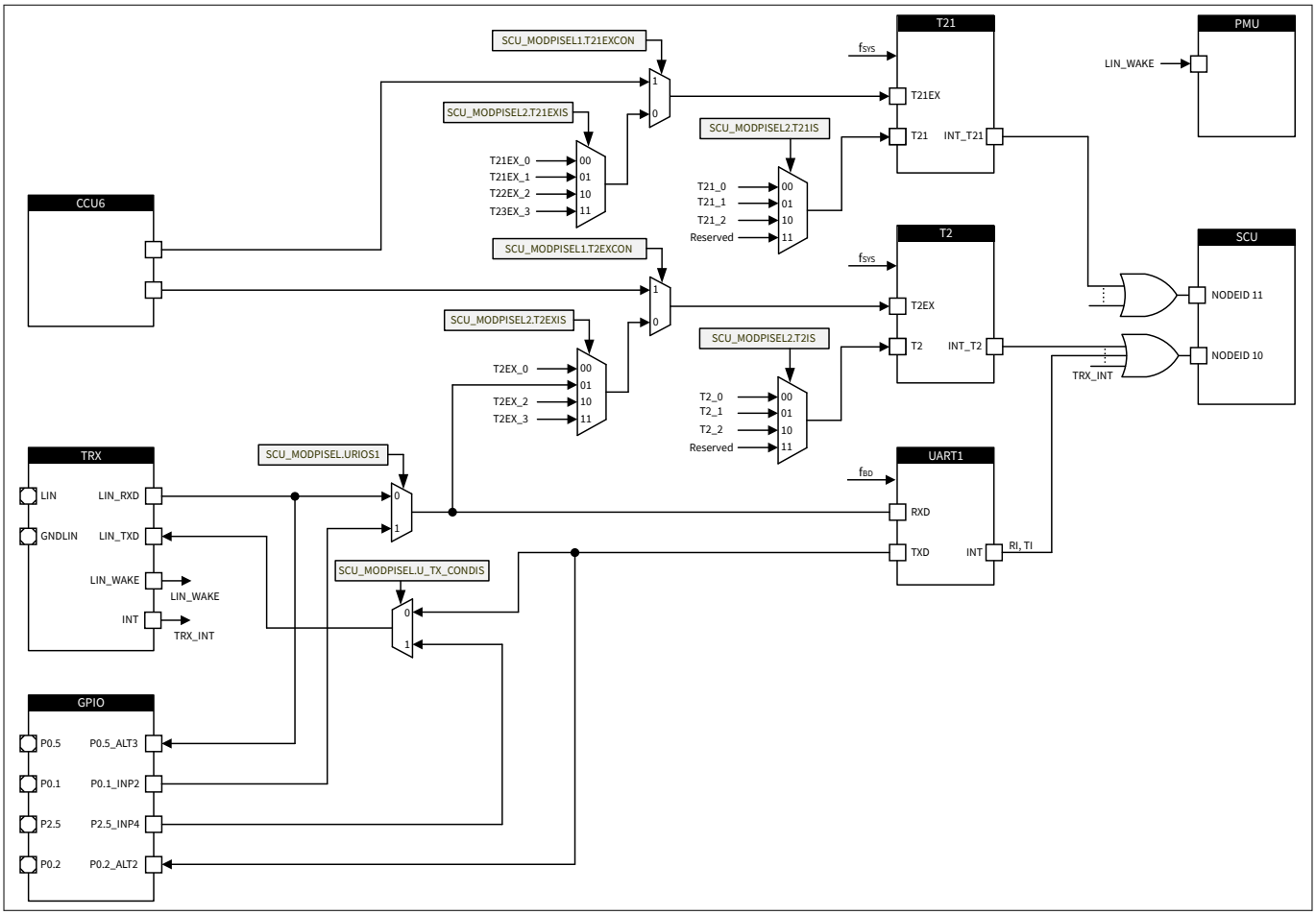


Figure 180 Interconnect TRX, UART1, TIMER2, GPIO, CCU6, SCU, PMU

19.4 LIN transceiver (TRX) register definition

19.4.1 Register address space - TRX

Table 141 Registers address space - TRX

Module	Base address	End address	Note
TRX	$4801E000_H$	$4801FFFF_H$	Transceiver registers

19.4.2 Register overview - TRX (ascending offset address)

Table 142 Register overview - TRX (ascending offset address)

Short name	Long name	Offset address	Page number
LIN_CTRL	Transceiver control register	0000_H	636
LIN_IRQS	Transceiver interrupt status register	0004_H	638
LIN_IRQCLR	Transceiver interrupt status rclear register	0008_H	639
LIN_IRQEN	Transceiver interrupt enable register	$000C_H$	640

19.4.3 Transceiver control register

LIN_CTRL Offset address: 0000_H

Transceiver control register RESET_TYPE_3 value: $0018\ XX07_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FB_SM			SM		RXD	TXD	RES		MODE_FB			RES	MODE		RES
r			rw		r	rw	r		r			r	rw		r

Field	Bits	Type	Description
RES	0, 3, 8:7, 18:16, 20:19, 22, 31:23	r	**Reserved** Always read as 1
MODE	2:1	rw	**Transceiver power mode control** 00_B **SLEEP**: Tranceiver module switched to Sleep mode 01_B **RECEIVE_ONLY**: Tranceiver module switched to Receive-Only mode 10_B **NU**: Not used 11_B **NORMAL**: Tranceiver module switched to Normal mode
MODE_FB	6:4	r	**Transmitter feedback signals settings [2:1]** 000_B **ERROR**: Mode error 001_B **SLEEP**: Transceiver Sleep mode 010_B **ERROR**: Mode error … 100_B **ERROR**: Mode error 101_B **RECEIVE_ONLY**: Transceiver Receive-Only mode 110_B **ERROR**: Mode error 111_B **NORMAL**: Transceiver Normal mode
TXD	9	rw	**Transmitter state (only used when HV_MODE is set)** 0_B **DOMINANT_STATE**: Transmitter switched on 1_B **RECESSIVE_STATE**: Transmitter switched off
RXD	10	r	**Reveiver output signal** Can be used to monitor the receiver output
SM	12:11	rw	**Transmitter slope mode control**

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: Slope mode can not be changed in normal mode. 00_B **NORMAL**: Normal slope mode 01_B **FAST**: Fast slope mode 10_B **LOW**: Low slope mode 11_B **FLASH**: Flash mode
FB_SM	15:13	r	**Feedback signal for slope mode setting [3:1]** 000_B **DISABLED**: Transceiver module not enabled 001_B **LOW**: Low slope mode 010_B **NORMAL**: Normal slope mode 011_B **FAST**: Fast slope mode 100_B **FLASH**: Flash mode 101_B **ERROR**: Slope mode error … 111_B **ERROR**: Slope mode error
HV_MODE	21	rw	**Transceiver high-voltage I/O mode** *Note: Switching to HVIO-mode (this configuration bit gets effective) is only possible when transceiver is in sleep mode.* 0_B **DISABLED**: High-voltage mode entry disabled 1_B **ENABLED**: High-voltage mode entry enabled

19.4.4 Transceiver interrupt status register

LIN_IRQS Offset address: 0004_H

Transceiver interrupt status register RESET_TYPE_3 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				TXD_TMOUT_STS	RES	OT_STS	M_SM_ERR_STS	RES	TXD_TMOUT_IS	OC_IS	OT_IS	M_SM_ERR_IS	RES		
r				rwhxr	r	rwhxr	rwhxr	r	rwhxre	rwhxr	rwhxre	rwhxre	r		

Field	Bits	Type	Description
RES	2:0, 7, 10, 31:12	r	**Reserved** Always read as 1
M_SM_ERR_IS	3	rwhxre	**Transceiver mode error - slope mode error interrupt status** 0_B **NO_ERROR**: No mode error slope mode status occurred 1_B **ERROR**: Mode error status occurred
OT_IS	4	rwhxre	**Transceiver overtemperature interrupt status** 0_B **NO_OT**: No overtemperature occurred 1_B **OT**: Overtemperature occurred
OC_IS	5	rwhxr	**Transceiver overcurrent interrupt status** 0_B **NO_OC**: No overcurrent status occurred 1_B **OC**: Overcurrent status occurred
TXD_TMOUT_IS	6	rwhxre	**Transceiver TXD timeout interrupt status** 0_B **NO_TMOUT**: No timeout occurred 1_B **TMOUT**: Timeout occurred
M_SM_ERR_STS	8	rwhxr	**Transceiver mode error - slope mode error status** 0_B **NO_ERROR**: No mode error slope mode status occurred 1_B **ERROR**: Mode error status occurred
OT_STS	9	rwhxr	**Transceiver overtemperature status** 0_B **NO_OT**: No overtemperature occurred 1_B **OT**: Overtemperature occurred
TXD_TMOUT_STS	11	rwhxr	**Transceiver TXD timeout status** 0_B **NO_TMOUT**: No timeout occurred 1_B **TMOUT**: Timeout occurred

19.4.5 Transceiver interrupt status rclear register

LIN_IRQCLR Offset address: 0008_H

Transceiver interrupt status rclear register RESET_TYPE_3 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				TXD_TMOUT_SC	RES	OT_SC	M_SM_ERR_SC	RES	TXD_TMOUT_ISC	OC_ISC	OT_ISC	M_SM_ERR_ISC	RES		
r				w	r	w	w	r	w	w	w	w	r		

Field	Bits	Type	Description
RES	2:0, 7, 10, 31:12	r	**Reserved** Always read as 1
M_SM_ERR_ISC	3	w	**Transceiver mode error - slope mode error interrupt status clear** 0_B **NO_CLR**: Overtemperature not cleared 1_B **CLR**: Overtemperature cleared
OT_ISC	4	w	**Tranceiver overtemperature interrupt status / status clear** 0_B **NO_CLR**: Overtemperature not cleared 1_B **CLR**: Overtemperature cleared
OC_ISC	5	w	**Tranceiver overcurrent interrupt status clear** 0_B **NO_CLR**: Overcurrent status not cleared 1_B **CLR**: Overcurrent status cleared
TXD_TMOUT_ISC	6	w	**Transceiver TXD timeout interrupt status clear** 0_B **NO_CLR**: No timeout cleared 1_B **CLR**: Timeout cleared
M_SM_ERR_SC	8	w	**Transceiver mode error - slope mode error status clear** 0_B **NO_CLR**: Overtemperature not cleared 1_B **CLR**: Overtemperature cleared
OT_SC	9	w	**Tranceiver overtemperature status clear** 0_B **NO_CLR**: Overtemperature not cleared 1_B **CLR**: Overtemperature cleared
TXD_TMOUT_SC	11	w	**Transceiver TXD timeout status clear** 0_B **NO_CLR**: No timeout cleared 1_B **CLR**: Timeout cleared

19.4.6 Transceiver interrupt enable register

LIN_IRQEN Offset address: $000C_H$

Transceiver interrupt enable register RESET_TYPE_3 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									**TXD_TMOUT_IEN**	**OC_IEN**	**OT_IEN**	**M_SM_ERR_IEN**	**RES**		
r									rw	rw	rw	rw	r		

Field	Bits	Type	Description
RES	2:0, 31:7	r	**Reserved** Always read as 1
M_SM_ERR_IEN	3	rw	**Transceiver mode error - slope mode error interrupt** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
OT_IEN	4	rw	**Transceiver overtemperature interrupt** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
OC_IEN	5	rw	**Transceiver overcurrent interrupt** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled
TXD_TMOUT_IEN	6	rw	**Transceiver TxD-timeout interrupt** 0_B **DISABLED**: Disabled 1_B **ENABLED**: Enabled

20 High-speed synchronous serial interface SSC1/SSC2

20.1 Features

- Master and slave mode operation
  - Full-duplex or half-duplex operation
- Transmit and receive double 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, e.g. 250 kBaud to 8 MBaud
- 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, transmit error)
  - On a transfer complete condition
- Port direction selection, see Chapter 14

20.2 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 or devices using other synchronous serial interfaces.

Data is transmitted or received on lines TXD and RXD, which are normally connected to the pins MTSR (master transmit/slave receive) and MRST (master receive/slave transmit). The clock signal is output through line MS_CLK (Master serial shift clock) or input through line SS_CLK (slave serial shift clock). Both lines are normally connected to the pin SCLK. Transmission and reception of data are double-buffered.

20.2.1 Block diagram

The following figure shows all functional relevant interfaces associated with the SSC kernel.

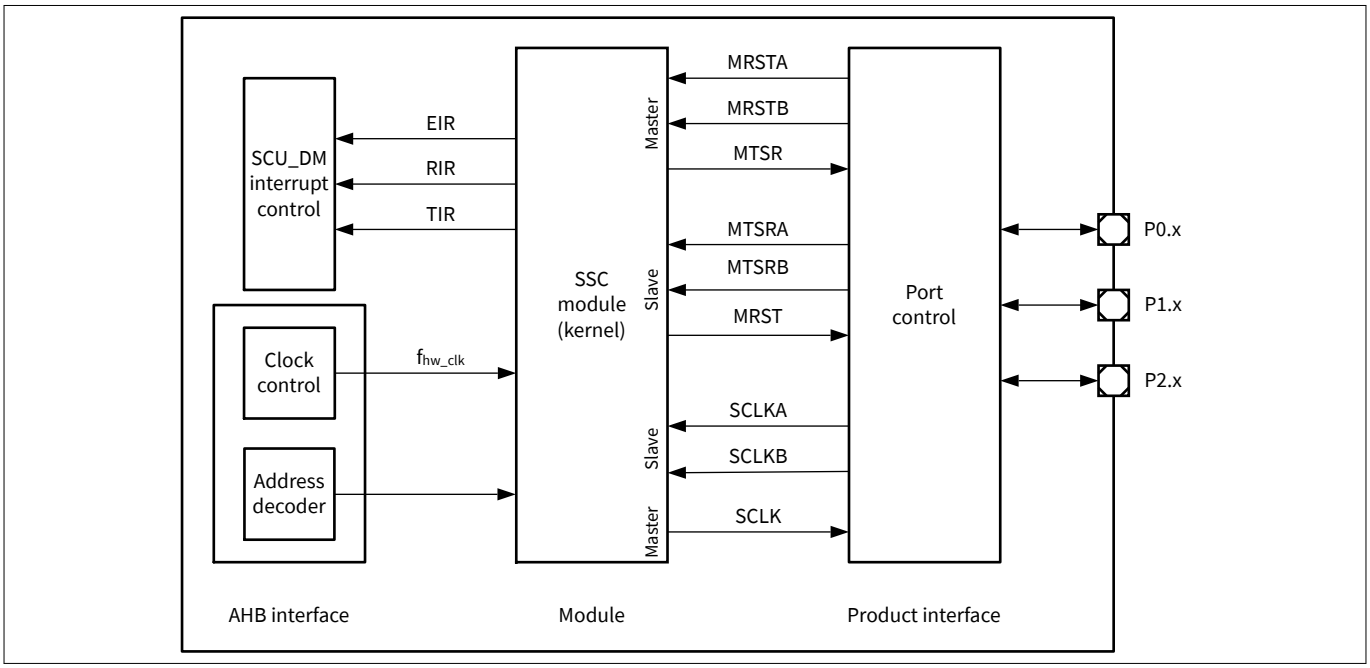


Figure 181 SSC interface diagram

20.3 Functional description

20.3.1 SSC1 and SSC2 mode overview

The SSC supports full-duplex and half-duplex synchronous communication up to 20 MBaud (at 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 multi master 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 (for example 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 through line MS_CLK (master serial shift clock) or input through line SS_CLK (slave serial Shift clock). Both lines are normally connected to pin SCLK.

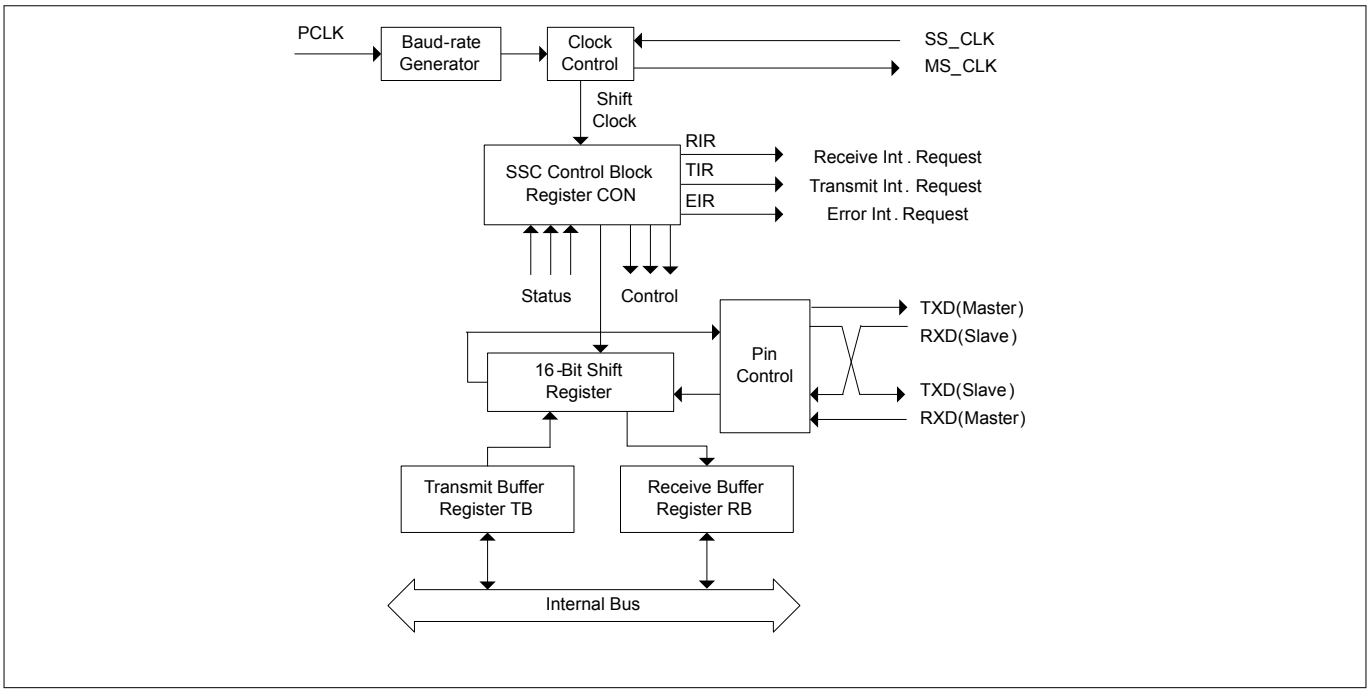


Figure 182 Synchronous serial channel SSC block diagram

20.3.2 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 through the pin control logic (see block diagram in Figure 182). Transmission and reception of serial data are synchronized and take place at the same time, that is 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 (at 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 8-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 the following figure).

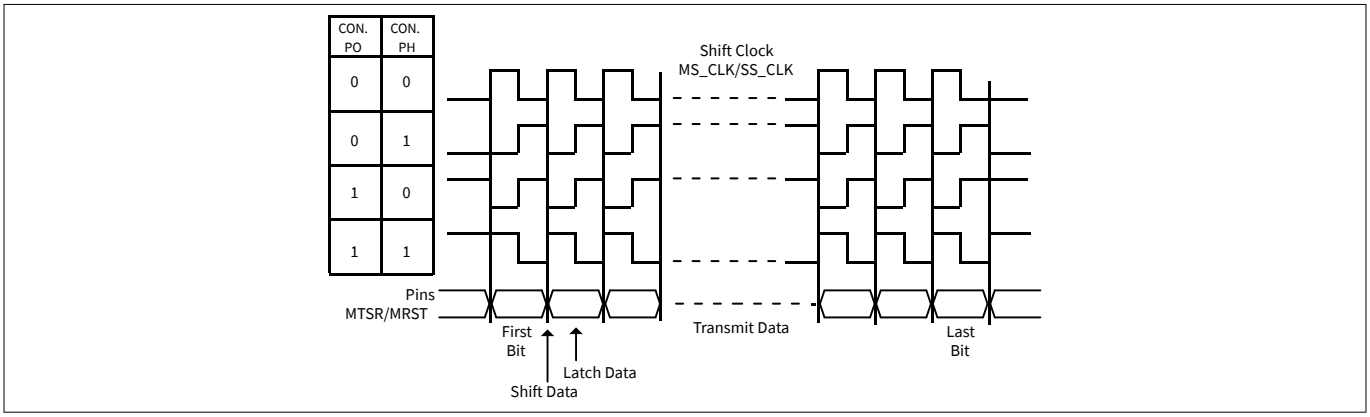


Figure 183 Serial clock phase and polarity options

20.3.3 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.

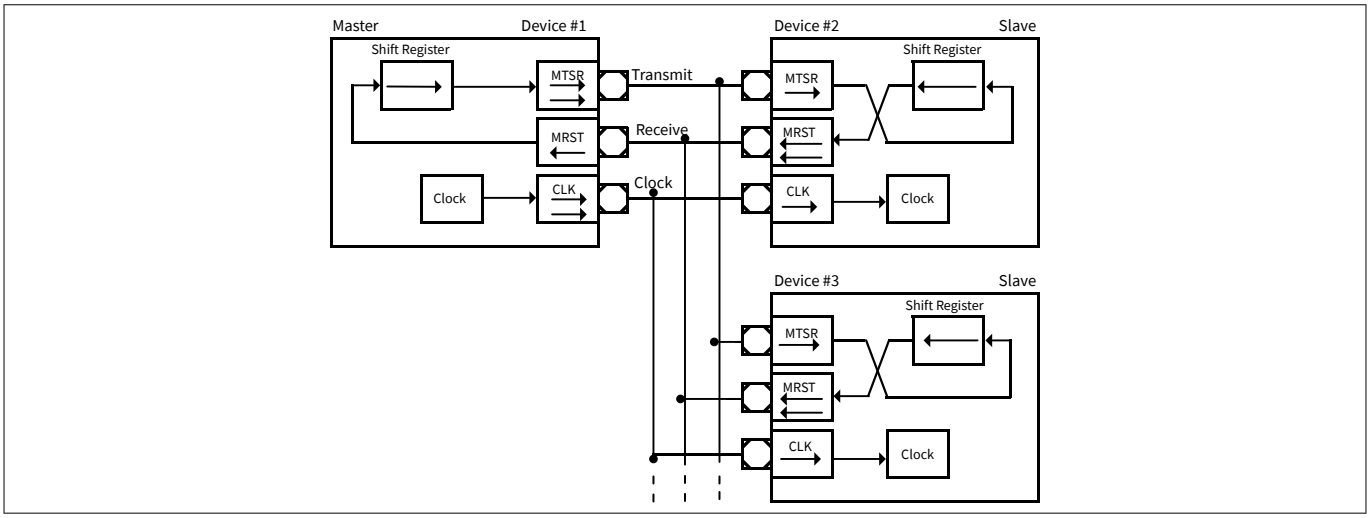


Figure 184 SSC full-duplex configuration

The data output pins MRST of all slave devices are connected together onto the one receive line in the configuration shown in the previous figure. 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, that is enabling 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 deselection 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 (1 s). 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 pre-programmed number of clock pulses (through 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 through the related direction control register, the clock output level will be selected in the control register CON and the alternate output be prepared through the related ALTSEL register, or the output latch must be loaded with the clock idle level.

20.3.4 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.

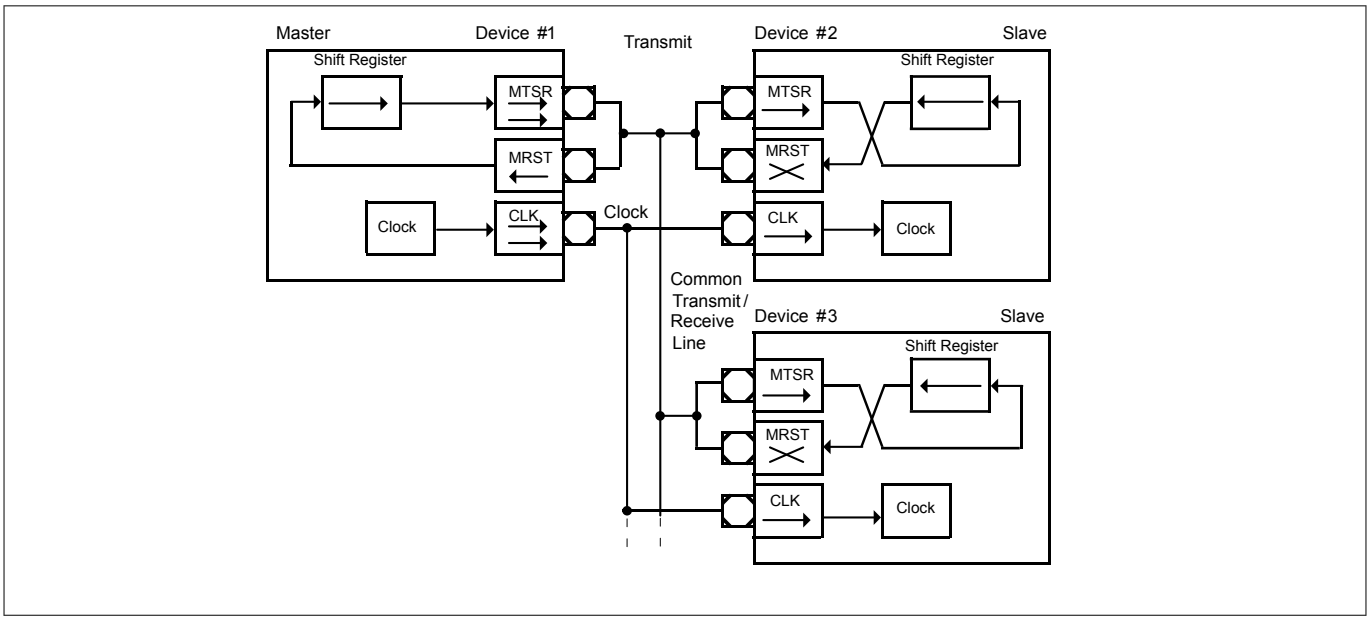


Figure 185 SSC half-duplex configuration

20.3.5 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.

20.3.5.1 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 181 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 either by hardware or 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.

20.3.6 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. Figure 182 shows the baud-rate generator. The following figure shows the baud-rate generator of the SSC in more detail.

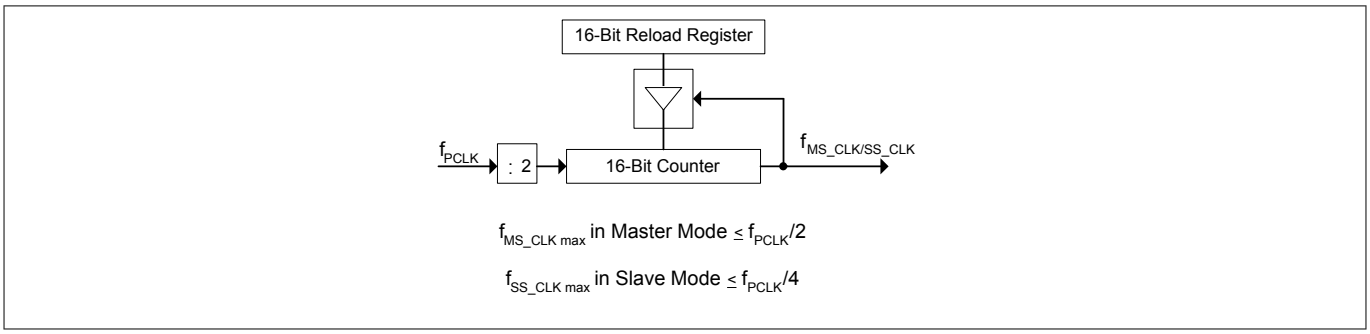


Figure 186 SSC baud-rate generator

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:

$$\text{Baus-rate} = \frac{f_{\text{hw\_clk}}}{2 \times (<\text{BR}> + 1)} \tag{14}$$

$$\text{BR} = \frac{f_{\text{hw\_clk}}}{2 \times \text{Baud-rate}} - 1 \tag{15}$$

 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 the previous figure.

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).

the following table lists some possible baud-rates together with the required reload values and the resulting bit times, assuming a module clock of 40 MHz.

Table 143 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%

(table continues...)

Table 143 (continued) Typical baud-rates of the SSC (f_{hw_clk} = 40 MHz)

Reload value	baud-rate (= f_{MS_CLK/SS_CLK})	Deviation
$07CF_H$	10 KBaud	0.0%
$4E1F_H$	1 KBaud	0.0%
$FFFF_H$	305.18 Baud	0.0%

20.3.7 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 the following figure) 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 rather must be cleared by software after servicing. This allows servicing of some error conditions through 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.

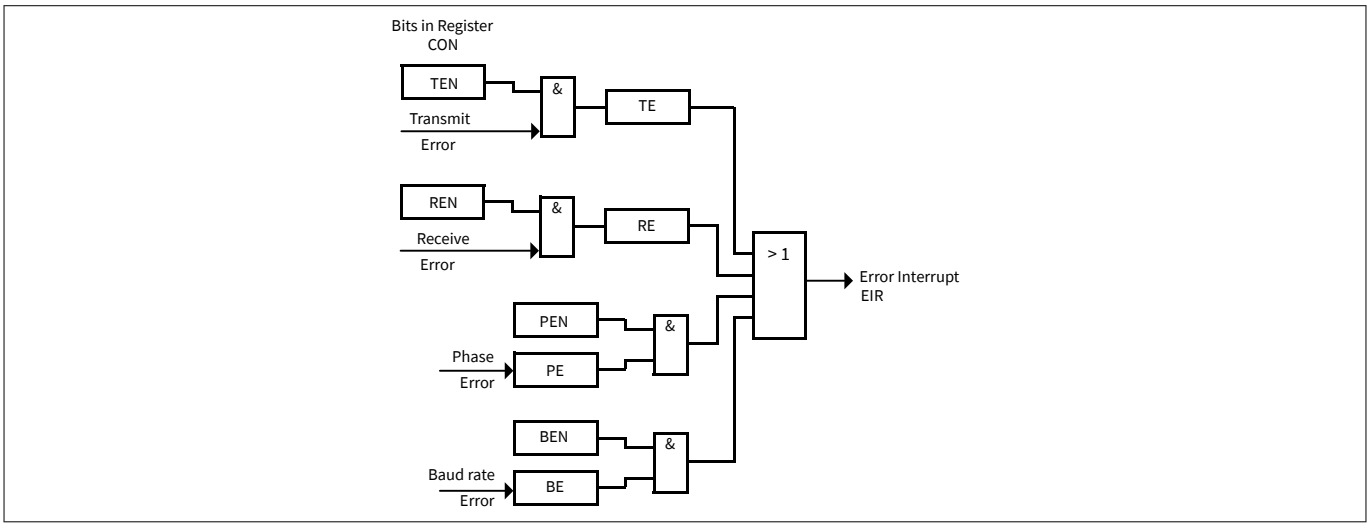


Figure 187 SSC error interrupt control

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 through 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 through 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 $f_{hw\_clk}/2$.

A baud-rate error (slave mode) is detected when the incoming clock signal deviates from the programmed baud-rate by more than 100%, meaning 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 through 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 through 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.

20.4 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 Chapter 20.3. An overview is given in the following table.

Table 144 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 the transmit buffer was not updated since the last transfer if a transfer is initiated by a master

20.5 SSC kernel registers

There are two SSC kernels in the MOTIX™ TLE984xQX, namely SSC1 and SSC2.

The registers are addressed wordwise.

Port input select register

The SSC_PISEL register controls the receiver input selection of the SSC module. In the implementation of MOTIX™ TLE984xQX, the SSC_PISEL register is not used.

Configuration register

The operating mode of the serial channel SSC is controlled by the control register SSC_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.

Baud-rate timer relead register

The SSC-BR register contains the 16-bit reload value for the baud-rate timer.

Transmitter buffer register

The SSC_TB register contains the transmit data value.

Reveiver buffer register

The SSC_RB register contains the receive data value.

20.5.1 High-speed synchronous serial interface (SSC) register definition

20.5.1.1 Register address space - SSC

Table 145 Registers address space - SSC

Module	Base address	End address	Note
SSC1	48024000_H	$48025FFF_H$	Synchronous serial interface 1 registers
SSC2	48026000_H	$48027FFF_H$	Synchronous serial interface 2 registers

20.5.1.2 Register overview - SSC (ascending offset address)

Table 146 Register overview - SSC (ascending offset address)

Short name	Long name	Offset address	Page number
SSC_PISEL	Port input select register	0000_H	655
SSC_CON	Control register	0004_H	656
SSC_TB	Transmitter buffer register	0008_H	661
SSC_RB	Receiver buffer register	$000C_H$	662
SSC_BR	Baud-rate timer reload register	0010_H	660
SSC_ISRCLR	Interrupt status register clear	0014_H	659

20.5.1.3 Port input select register

The PISEL register controls the receiver input selection of the SSC module. In the implementation of TLE984xQX, the PISEL register is not used.

SSC_PISEL Offset address: 0000_H

Port input select register RESET_TYPE_3 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												MIS_1	CIS	SIS	MIS_0
r												rw	rw	rw	rw

Field	Bits	Type	Description
MIS_0	0	rw	**Master mode input select bit 0 (master mode only)** 0_B **SSCx_M_MRST**: (x = 1 or 2, dependent form current SSC) 1_B **SSC12_M_MRST**: For both SSCs
SIS	1	rw	**Slave mode input select (slave mode only)** 0_B **SSCx_S_MTSR**: (x = 1 or 2, dependent form current SSC) 1_B **SSC12_S_MTSR**: For both SSCs
CIS	2	rw	**Clock input select (slave mode only)** 0_B **SSCx_S_SCK**: (x = 1 or 2, dependent form current SSC) 1_B **SSC12_S_SCK**: For both SSCs
MIS_1	3	rw	**Master mode input select bit 1 (master mode only)** 0_B **Default**: Inputs selected according to MIS_0 1_B **Do_not_use**: Connects to unused pins
RES	31:4	r	**Reserved** Always read as 0. Should be written with 0.

20.5.1.4 Control register

SSC_CON Offset address: 0004_H

Control register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			BSY	BE	PE	RE	TE	RES				BC			
r			r	r	r	r	r	r				r			

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** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_H **Reserved**: Do not use this combination 1_H F_H **Transfer_data**: Transfer datawidth is 2 … 16 bits (BM + 1)
HB	4	rw	**Heading control** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **LSB**: Transmit/Receive LSB first 1_B **MSB**: Transmit/Receive MSB first
PH	5	rw	**Clock phase control** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 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** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **LOW**: Idle clock line is low, leading clock edge is low-to-high transition 1_B **HIGH**: Idle clock line is high, leading clock edge is high-to-low transition
LB	7	rw	**Loop back control** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **NORMAL**: Output 1_B **LB**: Receive input is connected with transmit output (half-duplex mode)
TEN	8	rw	**Transmit error enable**

(table continues...)

(continued)

Field	Bits	Type	Description
			Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **IGNORE**: Transmit errors 1_B **CHECK**: Transmit errors
REN	9	rw	**Receive error enable** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **IGNORE**: Receive errors 1_B **CHECK**: Receive errors
PEN	10	rw	**Phase error enable** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **IGNORE**: Phase errors 1_B **CHECK**: Phase errors
BEN	11	rw	**Baud rate error enable** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). CHECK, baud rate errors. 0_B **IGNORE**: Baud rate errors 1_B **CHECK**: Baud rate errors
AREN	12	rw	**Automatic reset enable** Can only be accessed when EN = 0 (programming mode). Invalid data when EN = 1 (operating mode). 0_B **N_A**: No additional action upon a baud rate error 1_B **RESET**: The SSC is automatically reset upon a baud rate error
RES	13, 23:20, 31:29	r	**Reserved** Returns 0 if read; should be written with 0.
MS	14	rw	**Master select** 0_B **SLAVE**: Slave mode. Operate on shift clock received through SCLK 1_B **MASTER**: Master mode. Generate shift clock and output it through SCLK
EN	15	rw	**Enable bit** *Note: The effect of EN bit becomes visible on the next write to the CON register.* 0_B **Programming_mode**: Transmission and reception disabled. Access to control bits 1_B **Operating_mode**: Transmission and reception enabled. Access to status flags and M/S control
BC	19:16	r	**Bit count field** Can only be read when EN = 1 (operating mode). Invalid data when EN = 0 (programming mode). Shift counter is updated with every shift bit.

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: This bit-field is not to be written to.
TE	24	r	**Transmit error flag** Can only be read when EN = 1 (operating mode). Invalid data when EN = 0 (programming mode). 0_B **NO**: Error 1_B **ERROR**: Transfer starts with the slave's transmit buffer not being updated
RE	25	r	**Receive error flag** Can only be read when EN = 1 (operating mode). Invalid data when EN = 0 (programming mode). 0_B **NO**: Error 1_B **ERROR**: Reception completed before the receive buffer was read
PE	26	r	**Phase error flag** Can only be read when EN = 1 (operating mode). Invalid data when EN = 0 (programming mode). 0_B **NO**: Error 1_B **ERROR**: Received data changes around sampling clock edge
BE	27	r	**Baud rate error flag** Can only be read when EN = 1 (operating mode). Invalid data when EN = 0 (programming mode). 0_B **NO**: Error 1_B **ERROR**: More than factor 2 or 0.5 between slave's actual and expected baud rate
BSY	28	r	**Busy flag** Can only be read when EN = 1 (operating mode). Invalid data when EN = 0 (programming mode). Set while a transfer is in progress. *Note: This bit is not to be written to.*

20.5.1.5 Interrupt status register clear

SSC_ISRCLR Offset address: 0014_H

Interrupt status register clear RESET_TYPE_3 value: $0000\ 0000_H$

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

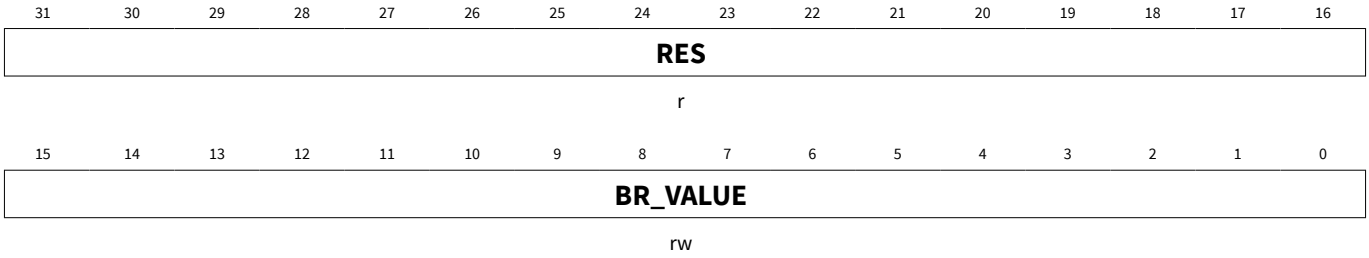
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				BECLR	PECLR	RECLR	TECLR	RES							
r				w	w	w	w	r							

Field	Bits	Type	Description
RES	7:0, 31:12	r	**Reserved** Always read as 0
TECLR	8	w	**Transmit error flag clear** 0_B **NO**: No error clear 1_B **CLEAR**: Error clear
RECLR	9	w	**Receive error flag clear** 0_B **NO**: No error clear 1_B **CLEAR**: Error clear
PECLR	10	w	**Phase error flag clear** 0_B **NO**: No error clear 1_B **CLEAR**: Error clear
BECLR	11	w	**Baud rate error flag clear** 0_B **NO**: No error clear 1_B **CLEAR**: Error clear

20.5.1.6 Baud-rate timer reload register

SSC_BR Offset address: 0010_H

Baud-rate timer reload register RESET_TYPE_3 value: $0000\ 0000_H$



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.
RES	31:16	r	**Reserved** Returns 0 if read; should be written with 0.

20.5.1.7 Transmitter buffer register

The SSC transmitter buffer register TB contains the transmit data value.

SSC_TB Offset address: 0008_H

Transmitter buffer register RESET_TYPE_3 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
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.
RES	31:16	r	**Reserved** Returns 0 if read. Should be written with 0.

20.5.1.8 Receiver buffer register

The SSC receiver buffer register RB contains the receive data value.

SSC_RB Offset address: $000C_H$

Receiver buffer register RESET_TYPE_3 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
RB_VALUE															
r															

Field	Bits	Type	Description
RB_VALUE	15:0	r	**Receive data register value** RB contains the received data value RB_VALUE. Unselected bits of RB will be not valid and should be ignored.
RES	31:16	r	**Reserved** Returns 0 if read; should be written with 0.

20.6 Output multiplexing

In case the multiplexed SSC-Port (SSC12_*) should be used, the outputs can be selected (from SSC1 or from SSC2). Please use the bits SSC_* in register SCU_MODPISEL for this purpose.

21 Measurement unit

21.1 Features

- 1 × 10-bit ADC with 13 inputs including attenuator allowing measurement of high voltage input signals
- Supply voltage attenuators with attenuation of VBAT_SENSE, VS, MONx, P2.x
- 1 × 8-bit ADC with 7 inputs including attenuator allowing measurement of high voltage input signals
- Supply voltage attenuators with attenuation of VS, VDDEXT, VDDP, VBG, VDDC, TSENSE_LS, TSENSE_CENTRAL
- VBG monitoring of 8-bit ADC to support functional safety requirements
- Temperature sensor for monitoring the chip temperature and low-side module temperature
- Supplement block with reference voltage generation, bias current generation, voltage buffer for NVM reference voltage, voltage buffer for analog module reference voltage and test interface

21.2 Introduction

The measurement unit is a functional unit that comprises the following associated submodules:

Table 147 Measurement functions and associated modules

Module Name	Modules	Functions
Central functions unit	Bandgap reference circuit + current reference circuit	The bandgap-reference submodule provides two reference voltages: **1.** An accurate reference voltage for the 10-bit and 8-bit ADCs. A local dedicated bandgap circuit is implemented to avoid deterioration of the reference voltage arising e.g. from crosstalk or ground voltage shift **2.** The reference voltage for the NVM module
10-bit ADC (ADC1)	10-bit ADC module with 13 multiplexed analog inputs	VBAT_SENSE, VS and MONx measurement Six (5 V) analog inputs from port 2.x
8-bit ADC (ADC2)	8-bit ADC module with 7 multiplexed inputs	VS/VDDEXT//VDDP/VBG/VDDC/TSENSE_LS and TSENSE_CENTRAL measurement
Temperature sensor	Temperature sensor readout amplifier with two multiplexed ΔVbe-sensing elements	Generates outputs voltage which is a linear function of the local chip (T_j) temperature
Measurement core module	Digital signal processing and ADC control unit	**1.** Generates the control signal for the 8-bit ADC2 and the synchronous clock for the switched capacitor circuits (temperature sensor) **2.** Performs digital signal processing functions and provides status outputs for interrupt generation

21.2.1 Block diagram

The structure of the measurement functions module is shown in the following figures.

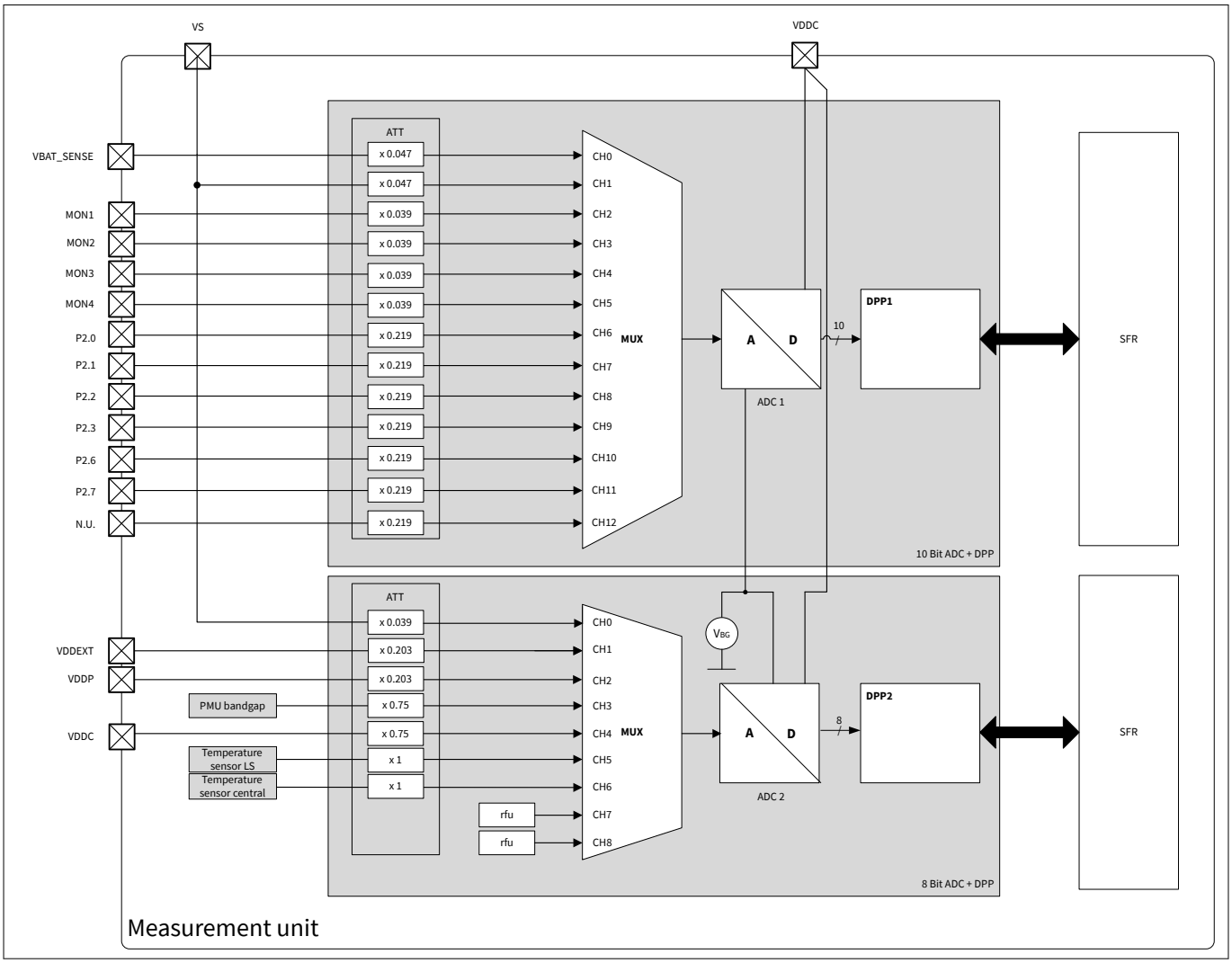


Figure 188 MOTIX™ TLE984xQX measurement unit overview (4MON)

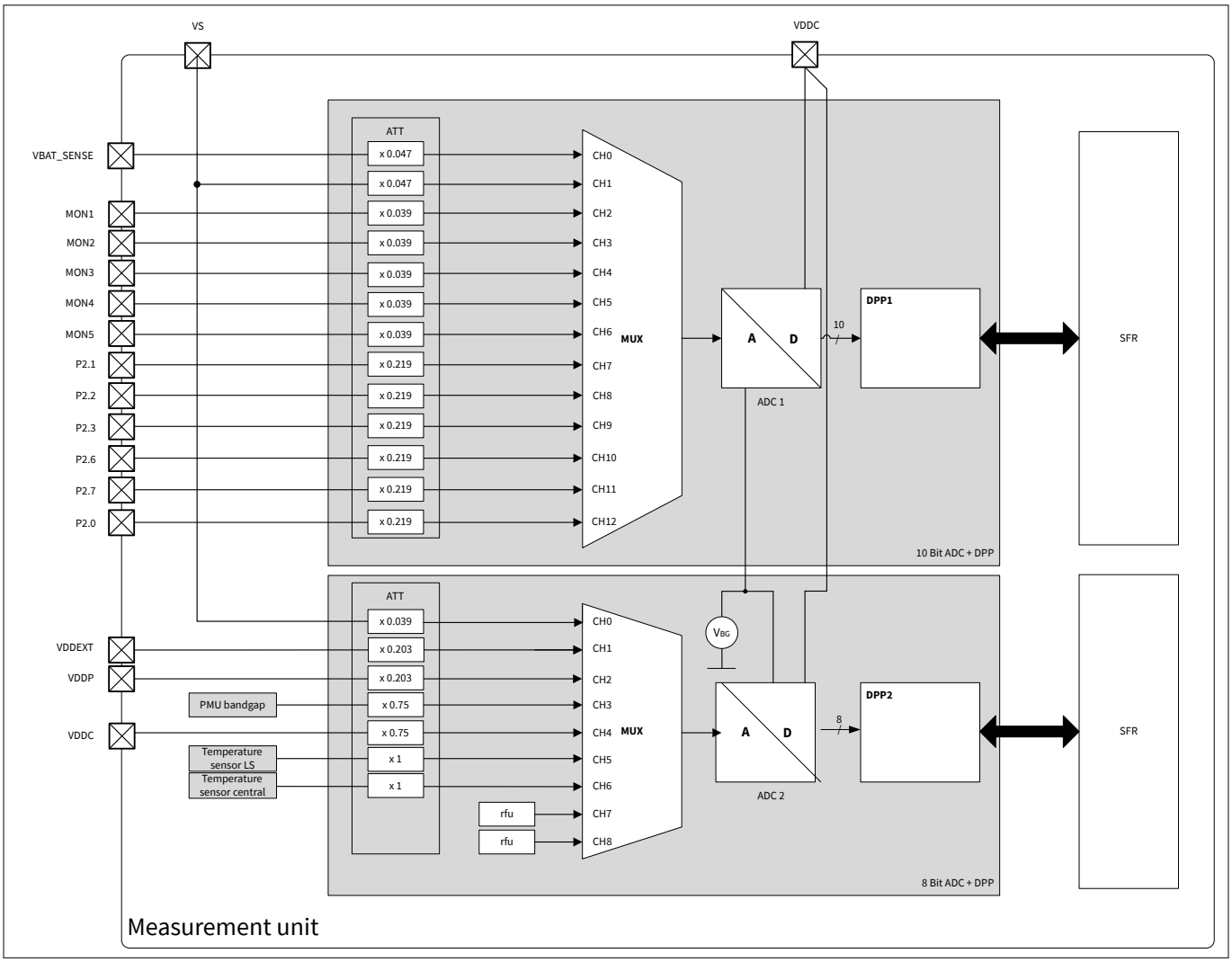


Figure 189 MOTIX™ TLE984xQX measurement unit overview (5MON)

21.3 8-bit 10 channel ADC core

The 8-bit ADC core operates at the VDDC supply voltage. This enables the user to operate the measurement system down to reset threshold. The ADC can also be operated independently from the DPP 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: 12)
- Scalable clock frequency from 10 - 30 MHz

The next chapter shows the channel allocation of the 8-bit ADC core

21.3.1 8-bit ADC channel allocation

The allocation of the 7 channels of ADC2 is sketched below:

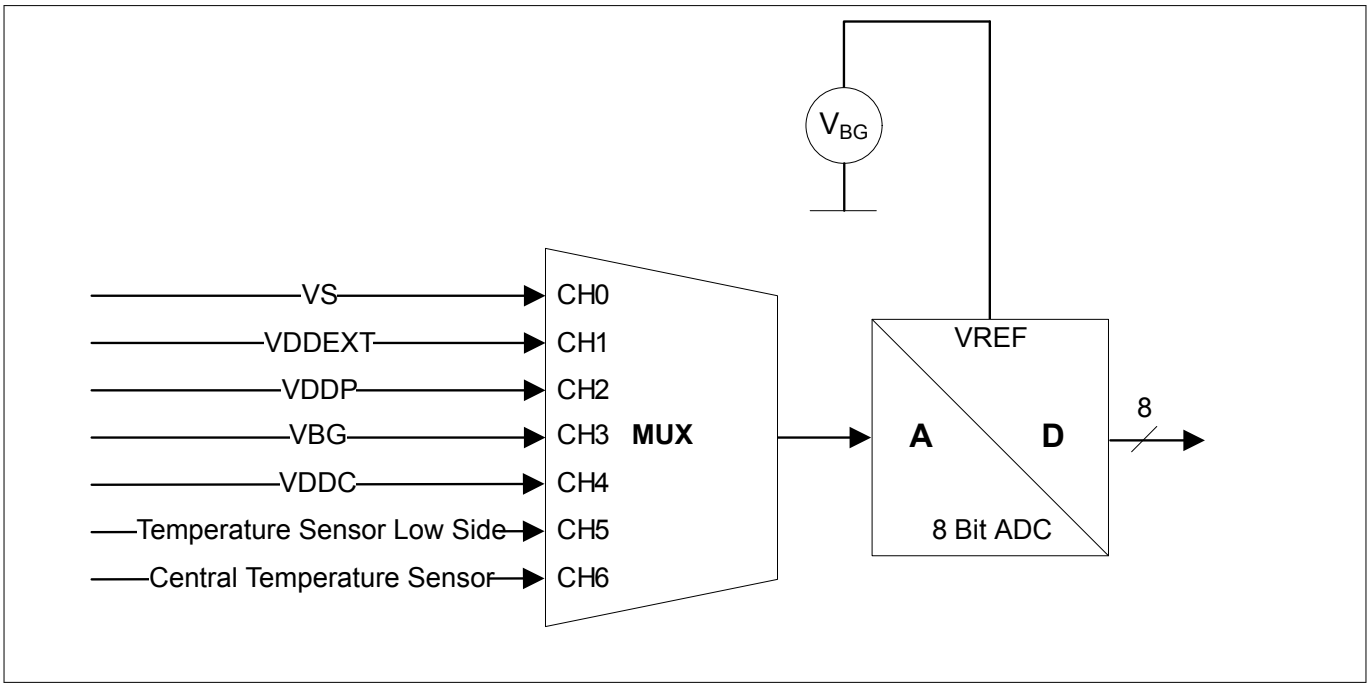


Figure 190 8-bit ADC (ADC2) channel allocation

- *VS* pin voltage measurement
- *VDDEXT* pin voltage measurement
- *VDDP* pin voltage measurement
- *VDDC* pin voltage measurement
- ADC2 reference voltage check (VBG)
- Low-side temperature measurement (T_j)
- Power management unit temperature measurement (T_j)

21.3.2 Transfer characteristics of ADC2

The transfer function of ADC2 can be expressed by the equation below:

$$ADC2out = floor\left(\frac{Vin * Gain_{CHx}}{Vlsb} + 1\right) \quad (16)$$

where Vin is the input voltage and $Gain_{CHx}$ the individual channel gain. The LSB voltage is calculated:

$$Vlsb = \frac{Vref}{256} \quad (17)$$

where Vref is V_{BG} (P_9.1.10).

A detailed specification of both A/D-converters is given in chapter . The Gain for each channel can be found in the table included in the following chapter.

21.3.3 Detailed ADC2 measurement channel description

Table 148 ADC2 channel selection and voltage ranges

Channel #	Measurement input pin	Gain of channel	Vin_FS [V] at V_{BG}
0	VS	5/128	31.05
1	VDDEXT	26/128	5.97
2	VDDP	26/128	5.97
3	Vbg	96/128	1.61
4	VDDC	96/128	1.61
5, 6	Temperature sensor	1	1.21

21.3.4 8-bit 10 channel control register

The ADC2 control register is located in the Measurement core module (incl. ADC2) block.

21.4 10-bit channel ADC core

The 10-bit ADC is using port 2.x as inputs. The configuration possibilities of the input channels are shown below.

21.4.1 10-bit ADC channel allocation

The allocation of the 12 channels of ADC1 is sketched below:

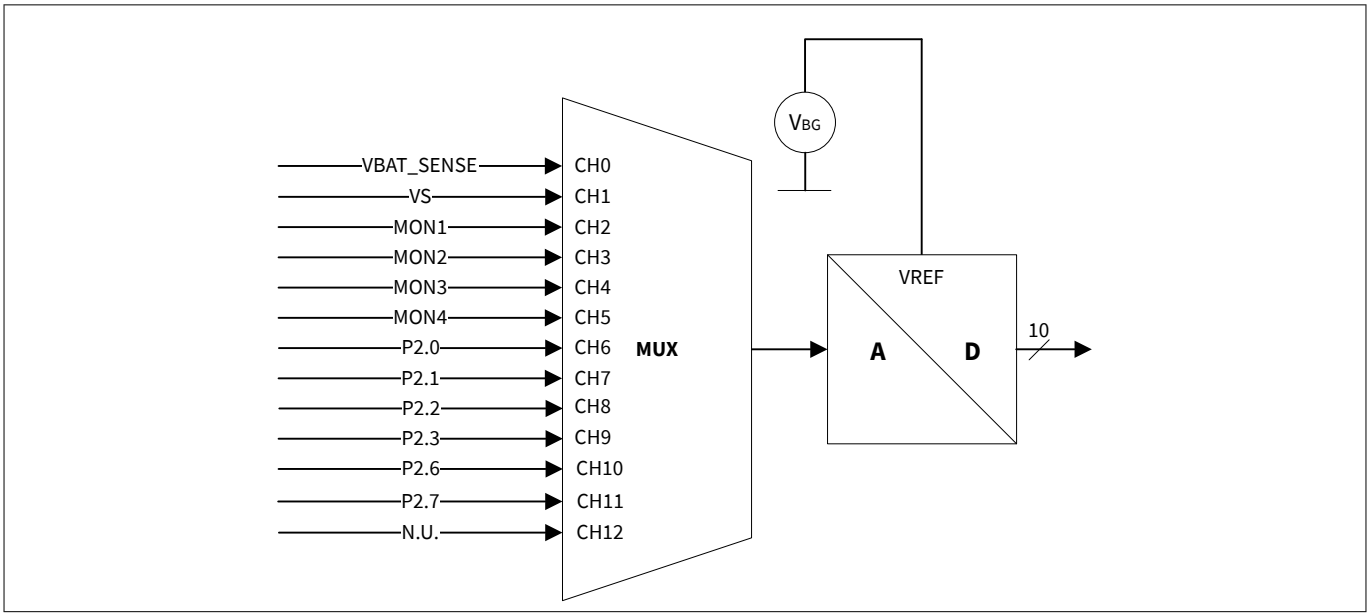


Figure 191 10-bit ADC (ADC1) channel allocation (4MON)

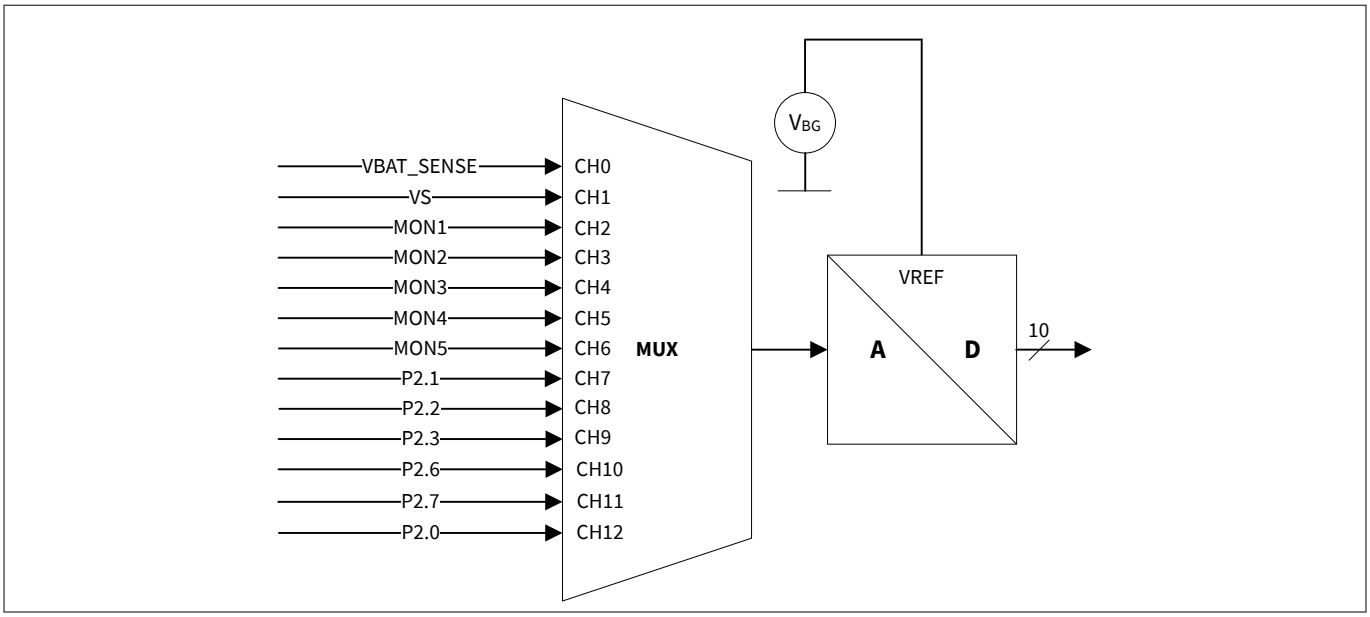


Figure 192 10-bit ADC (ADC1) channel allocation (5MON)

- *VS* pin voltage measurement
- *VBAT_SENSE* pin voltage measurement
- *MON1-5*[8)] pin voltage measurement
- *P2.X* pin voltage measurement

8 MON5 is device variant specific.

Table 149 ADC1 channel selection and voltage ranges

Channel #	Measurement input pin	Gain of channel	Vin_FS [V] at V_{BG}
0	vsense_vbat_ai	12/256	25.77
1	vsense_vs_ai	12/256	25.77
2	vsense_vmon1_ai	10/256	31.05
3	vsense_vmon2_ai	10/256	31.05
4	vsense_vmon3_ai	10/256	31.05
5	vsense_vmon4_ai	10/256	31.05
6	vsense_vmon5_ai	10/256	31.05
7	vsense_p21_ai	56/256	5.53
8	vsense_p22_ai	56/256	5.53
9	vsense_p23_ai	56/256	5.53
10	vsense_p26_ai	56/256	5.53
11	vsense_p27_ai	56/256	5.53
12	vsense_p20_ai	56/256	5.53

21.5 Central and PMU regulator temperature sensor

This module is a quasi combination of a main on-chip temperature sensor and a PMU regulator temperature sensor.

Modules Features

- Two operation modes with:
  Mode 1 – temperature range corresponds to differential output voltage range 0…1.2 V (output voltage shift enabled), resolution approximately 10°C.
  Mode 2 – temperature range corresponds to differential output voltage range 0.6…1.2 V, resolution approx. 15°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

This temperature sensor, including two sensing elements, monitors the chip temperature and PMU regulator temperature. One sensing element is placed in the centre of the device to get the average device temperature status, while the other sensing element is close to the PMU regulator

The voltage calculation of the temperature is done with the following formula:

$$ADC2out = floor\left(\frac{Vtemp}{Vlsb} + 1\right) \tag{18}$$

The LSB voltage is calculated:

$$Vlsb = \frac{Vref}{256} \tag{19}$$

Vtemp is the direct proportional to temperature input voltage and is calculated by:

$$Vtemp\,(T) = a + b \times (T - T_0) \tag{20}$$

where the coefficient a is 628 mV, b is 2,31 mV/K and T_0 is 273 K:

The next chapter lists the available registers to configure both temperature sensors.

21.5.1 Temperature sensor control register

The temperature sensor is fully controllable by the below listed SFR register.

The registers are addressed bytewise.

21.5.1.1 Register overview - Temperature sensor control registers (ascending offset address)

Table 150 Register overview - Temperature sensor control registers (ascending offset address)

Short name	Long name	Offset address	Page number
MF_TEMPSENSE_CTRL	Temperature sensor control register	0010_H	674

21.6 Supplement modules

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%
- Bandgap is monitored by an independent reference voltage
- ADC1 reference with accuracy ± 1%
- ADC1 reference has overload detection

The next chapter lists the configuration possibilities of the on chip references.

21.6.1 Functional safety concept

8-bit ADC module 2

- A known voltage, for example reference voltage of the main supply module, is periodically measured as part of the measurement sequence in normal operation (The local ADC's reference voltage can, of course, not be used for this purpose since a local reference voltage error would not be detectable.)
- The conversion result of the functional safety measurement is evaluated in the post processing unit. If the results is not within the expected range an error is indicated

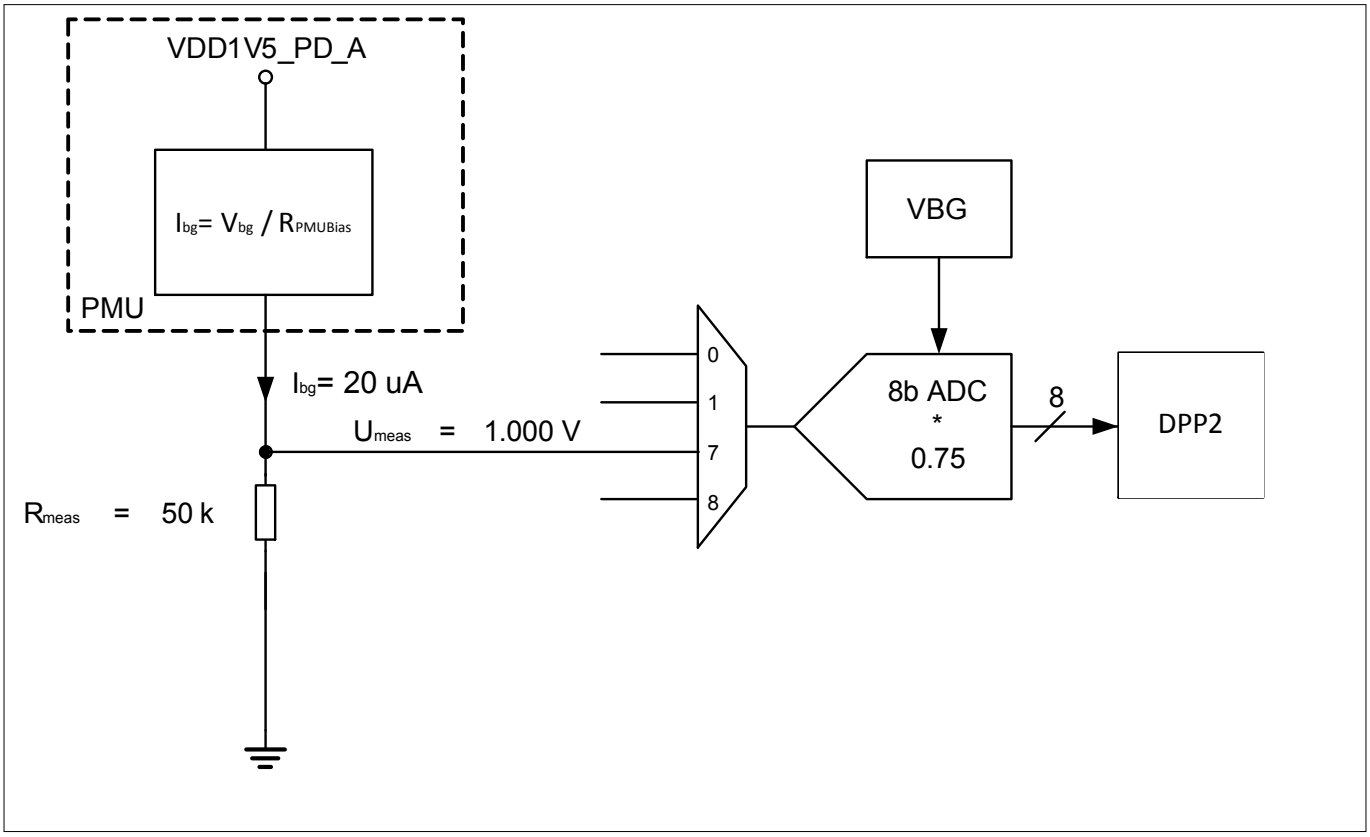


Figure 193 Principle of PMU bandgap measurement

21.6.2 Supplement modules control and status register

The next chapter lists the diagnosis and configuration possibilities of the supplement modules.

The registers are addressed bytewise.

21.6.2.1 Register overview - Supplement modules control and status registers (ascending offset address)

Table 151 Register overview - Supplement modules control and status registers (ascending offset address)

Short name	Long name	Offset address	Page number
MF_REF1_STS	Reference 1 status register	0014_H	675

21.7 Measurement unit (MF) register definition

21.7.1 Register address space - MF

Table 152 Registers address space - MF

Module	Base address	End address	Note
MF	48018000_H	$4801BFFF_H$	Measurement unit registers

21.7.2 Register overview - MF (ascending offset address)

Table 153 Register overview - MF (ascending offset address)

Short name	Long name	Offset address	Page number
MF_TEMPSENSE_CTRL	Temperature sensor control register	0010_H	674
MF_REF1_STS	Reference 1 status register	0014_H	675

21.7.3 Temperature sensor control register

MF_TEMPSENSE_CTRL Offset address: 0010_H

Temperature sensor control register RESET_TYPE_4 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						RES		SYS_OT_STS	SYS_OTWARN_STS	LS_OT_STS	LS_OTWARN_STS	RES			
r						r		r	r	r	r	r			

Field	Bits	Type	Description
RES	3:0, 9:8, 31:10	r	**Reserved** Always read as 0
LS_OTWARN_STS	4	r	**Low-side overtemperature warning (MU) status** 0_B **INACTIVE**: Write clears status 1_B **ACTIVE**: Interrupt status set
LS_OT_STS	5	r	**Low-side overtemperature (MU) status** 0_B **INACTIVE**: Write clears status 1_B **ACTIVE**: Interrupt status set
SYS_OTWARN_STS	6	r	**System overtemperature warning (MU) status** 0_B **INACTIVE**: Write clears status 1_B **ACTIVE**: Interrupt status set
SYS_OT_STS	7	r	**System overtemperature (MU) status** 0_B **INACTIVE**: Write clears status 1_B **ACTIVE**: Interrupt status set

21.7.4 Reference 1 status register

MF_REF1_STS Offset address: 0014_H

Reference 1 status register RESET_TYPE_3 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						RES				REFBG_UPTHWARN_STS	REFBG_LOTHWARN_STS	RES	RES		
r						r				r	r	r	r		

Field	Bits	Type	Description
RES	2:0, 3, 9:6, 31:10	r	**Reserved** Always read as 0
REFBG_LOTHWARN_STS	4	r	**Status for Undervoltage threshold measurement of internal VAREF** 0_B **UPPER_TRIG_RESET**: Write clears status 1_B **UPPER_TRIG_SET**: Trigger status set
REFBG_UPTHWARN_STS	5	r	**Status for overvoltage threshold measurement of internal VAREF** 0_B **UPPER_TRIG_RESET**: Write clears status 1_B **UPPER_TRIG_SET**: Trigger status set

22 Measurement core module (incl. ADC2)

22.1 Features

- 7 individually programmable channels split into two groups of user configurable and non user configurable
- Individually programmable channel prioritization scheme for 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 status for all channel thresholds
- Operation down to reset threshold of entire system

22.2 Introduction

The basic function of this block is the digital post-processing of several analog digitized measurement signals by means of filtering level comparison and interrupt generation. The measurement post-processing block is built of seven identical channel units attached to the outputs of the 7-channel 8-bit ADC (ADC2). It processes seven channels, where the channel sequence and prioritization is programmable within a wide range.

22.2.1 Block diagram

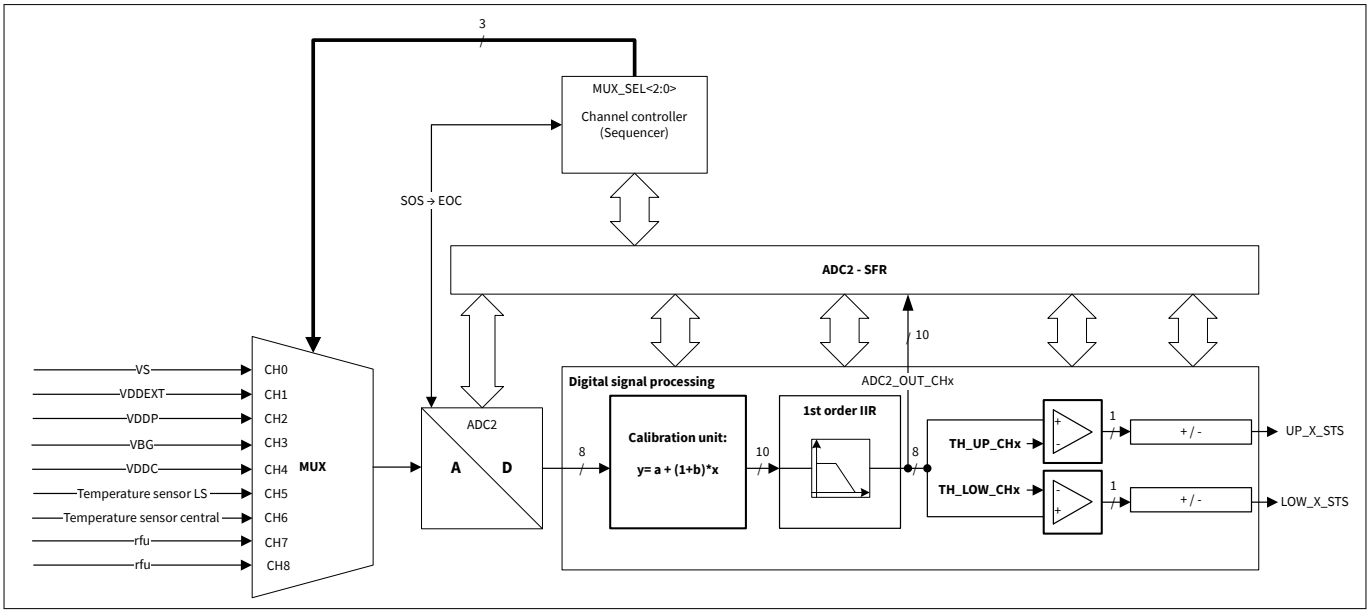


Figure 194 Module block diagram

22.2.2 Measurement core module modes overview

The basic function of this unit, is the digital signal processing of several analog digitized measurement signals by means of filtering, level comparison and interrupt generation. The measurement core module processes eight channels in a quasi parallel evaluation process.

As shown in the figure above, the ADC2 post processing consists of a channel controller (Sequencer), an 8-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 on the digital domain after the ADC2. As described in the following section, the channel sequence can be controlled in a flexible way, which allows a certain degree of channel prioritization.

This capability can be used for example to set a higher priority to supply voltage channels compared to the other channel measurements. The measurement core module offers additionally two different post-processing measurement modes for over-/undervoltage detection and for two-level threshold detection.

Usually the external register settings should only be changed during the start-up phase (ADC2_CTRL2).

Software mode, sequencer and exceptional interrupt measurement is disabled, each measurement is triggered by software.

The IIR filter can be bypassed through ADC2_FILT_UP_CTRL and ADC2_FILT_LO_CTRL for the data transferred to post processing only. The threshold counter can be bypassed (counting only 1 measurement) through CNT_LO_CHx.

22.3 ADC2 - Core (8-bit ADC)

22.3.1 Functional description

The different sequencer modes are controlled by SFR register

- Normal sequencer mode described in Chapter 22.4
- Exceptional interrupt measurement (EIM), upon hardware event , the channel programmed in ADC2_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
- 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 hardware events are ignored

Software mode

- Software mode can be entered by
  - writing one of the sequence registers SQn (e.g. SQ_{1_4}[7:0] zero) or
  - using debug suspend mode
- In software mode, the channel selection by the sequencer is disabled. After the software mode is entered, the conversions are controlled via ADC2_CTRL_STS
- The software mode is left when
  - the maximum time is reached (maximum time specified in ADC2_MAX_TIME) or
  - when the sequence which started the software mode is reprogrammed with at least one channel set is registers SQn (e.g. to SQ_{1_4}[7:0] zero
  - leaving debug suspend mode

Software mode:

In Software mode measurements are triggered by writing the ADC2_CTRL_STS.SOS bit. This bit is active as long as the conversion is in progress. The user polls the ADC2_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 ADC2_CTRL_STS.SOS.

Debug Suspend Mode:

During Debug Suspend Mode the Sequencer is stopped once the current measurement is finished (after the next EOC event) and Software Mode is entered. 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/Software Mode. The maximum time of software mode is disabled in suspend mode. EIM events are ignored during Debug Suspend Mode.

The ADC2 timing is controlled by SFR register

- Sample time adjustment described in the register ADC2_CTRL2.

22.3.2 ADC2 control registers

The ADC2 is fully controllable by the below listed SFR registers. The control must be enabled by setting all bits sequencer bits to zero. To enable the sequencer again this corresponding bits in the sequencer register must be set to one again.

The registers are addressed wordwise.

22.3.2.1 Register overview - ADC2 control registers (ascending offset address)

Table 154 Register overview - ADC2 control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC2_CTRL_STS	ADC2 control and status register	0000_H	696
ADC2_STATUS	ADC2 HV status register	$00BC_H$	697

22.4 Channel controller

22.4.1 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 155 Measurement channel sequence definition example (used as default sequence)

Measurement channel n	CH6	CH5	CH4	CH3	CH2	CH1	LSB CH0
Registers {SQ_1_4[6:0]}	0	1	1	0	1	1	1
Registers {SQ_1_4[14:8]}	1	0	0	1	0	0	0
Registers {SQ_1_4[22:16]}	0	1	1	0	1	1	0
Registers {SQ_1_4[30:24]}	1	0	0	1	0	0	1
Registers {SQ_5_8[6:0]}	0	1	1	0	1	1	0
Registers {SQ_5_8[14:8]}	1	0	0	1	0	0	0
Registers {SQ_5_8[22:16]}	0	1	1	0	1	1	1

The sequence registers SQ_n and define the time sequence of the measurement channels by the following rules:

- The sequence registers define the measurement sequence and are evaluated from sequence 1 to 7 and for each register from MSB to LSB, which defines a max. overall measurement periodicity of 49 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 not set, this measurement phase is skipped

In the upper example, the resulting channel sequence is defined as:

CH5, CH4, CH2, CH1, CH0, CH6, CH3, CH5, CH4, CH2, CH1,..., CH5, CH4, CH2, CH1, CH0.

In MOTIX™ TLE984xQX channels 0 - 6 can not be programmed by the user. All sequence registers, especially for high priority channels are protected to ensure a fast update of measurement results used for internal system diagnosis. The overall periodicity is mainly determined by this two channels. The channels 0-6 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.

The overall measurement periodicity of all measurements in A/D conversion cycles is defined as:

$$\overline{N_{\text{meas}}} = \sum_{m=1}^{7}\left(\sum_{n=0}^{6} SQ_m[n]\right) \tag{21}$$

The average measurement periodicity of channel n in A/D conversion cycles is defined as

$$\overline{N_{\text{meas},\,n}} = \frac{\left(\sum_{m=1}^{7} SQ_m[n]\right)}{\overline{T_{\text{meas}}}} \tag{22}$$

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 consumes, as

default setting, 12 clock cycles for the sampling of the input voltage. The conversion time for a single channel measurement value is 10 clock cycles.

As already mentioned above, the channel controller has a 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, for example shutdown in case of overtemperature for the low sides and protection overtemperature protection of the system.

The minimum measurement periodicity, which can be achieved, by enabling only channel 1 in the sequence registers, depends on the MI_CLK frequency and is given by:

$$\overline{T_{\text{meas\_CH1\_min}}} = \frac{32}{f_{\text{MI\_CLK}}} \tag{23}$$

This following calculations include already the sampling time of ADC2. If all programmable channels are enabled, the maximum periodicity is calculated:

$$\overline{T_{\text{meas\_CH1\_max}}} = \frac{320}{f_{\text{MI\_CLK}}} \tag{24}$$

For a MI_CLK frequency of 24 MHz, the channel 1 is measured with min. 4 µs. The maximum update time of channel 1 with 24 MHz clock frequency is 10 µs. As mentioned before, this is calculated with the assumption, that all channels are enabled and channel 1 is enabled in every sequence register. As a prerequisite for this calculation we take ADC2_CTRL2 = 4 (sample period = 14 MI_CLK clock cycles).

22.4.2 Channel controller control registers

The channel controller can be configured by the SFR registers listed below.
The registers which cannot be written by the user have the attribute rwpt.
The registers are addressed wordwise.

22.4.2.1 Register overview - Channel controller control registers (ascending offset address)

Table 156 Register overview - Channel controller control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC2_SQ_FB	Sequencer feedback register	0004_H	698
ADC2_CHx_EIM	Channel settings bits for exceptional interrupt measurement register	0008_H	700
ADC2_MAX_TIME	Maximum time for software mode register	0010_H	702
ADC2_CTRL1	Measurement unit control 1 register	0014_H	703
ADC2_CTRL2	Measurement unit control 2 register	0018_H	704
ADC2_CTRL4	Measurement unit control 4 register	$001C_H$	705
ADC2_SQ1_4	Measurement channel enable bits for cycle 1-4 register	0020_H	706
ADC2_SQ5_8	Measurement channel enable bits for cycle 5-8 register	0024_H	708

22.5 Calibration unit

22.5.1 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 chapter describe usage and setup of the calibration unit.

22.5.1.1 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. This 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

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.

22.5.1.2 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 100 TP 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 a 8 bit, 2th complement format.

The function applied to calculate the calibrated ADC2 value is

ADC_cal_CHx = (1 + <ADC2_CALGAIN_CHx>/256) * ADC_uncal_CHx + <ADC2_CALOFFS_CHx>/2

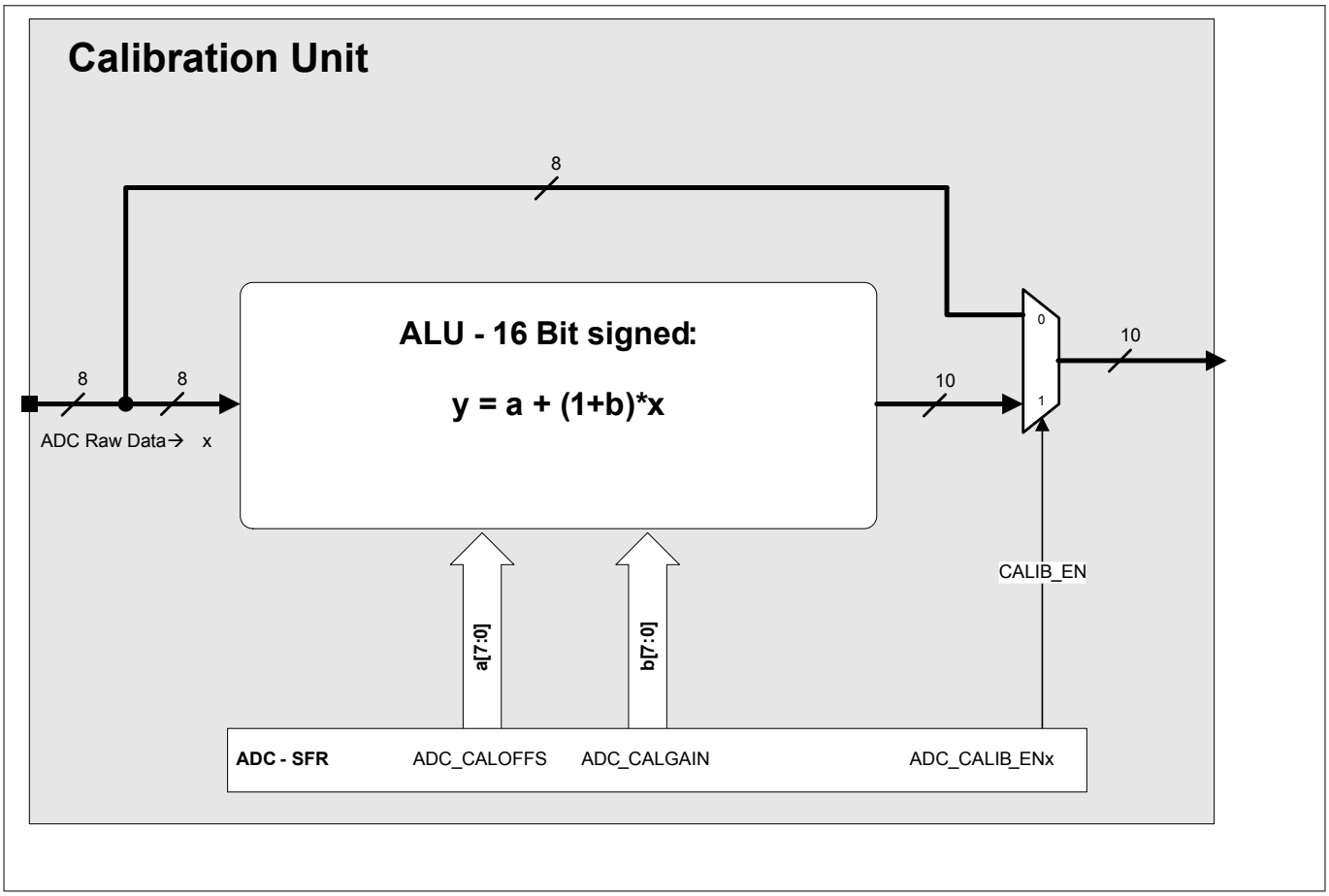


Figure 195 Structure of calibration unit

22.5.2 Calibration unit control registers

The calibration unit can be configured by the SFR registers listed below.

These calibration registers cannot be written by the user.

The registers are addressed wordwise.

22.5.2.1 Register overview - Calibration unit control registers (ascending offset address)

Table 157 Register overview - Calibration unit control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC2_CAL_CH0_1	Calibration for channel 0 and 1 register	0034_H	709
ADC2_CAL_CH2_3	Calibration for channel 2 and 3 register	0038_H	710
ADC2_CAL_CH4_5	Calibration for channel 4 and 5 register	$003C_H$	711
ADC2_CAL_CH6_7	Calibration for channel 6 and 7 register	0040_H	712

22.6 IIR-filter

22.6.1 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.

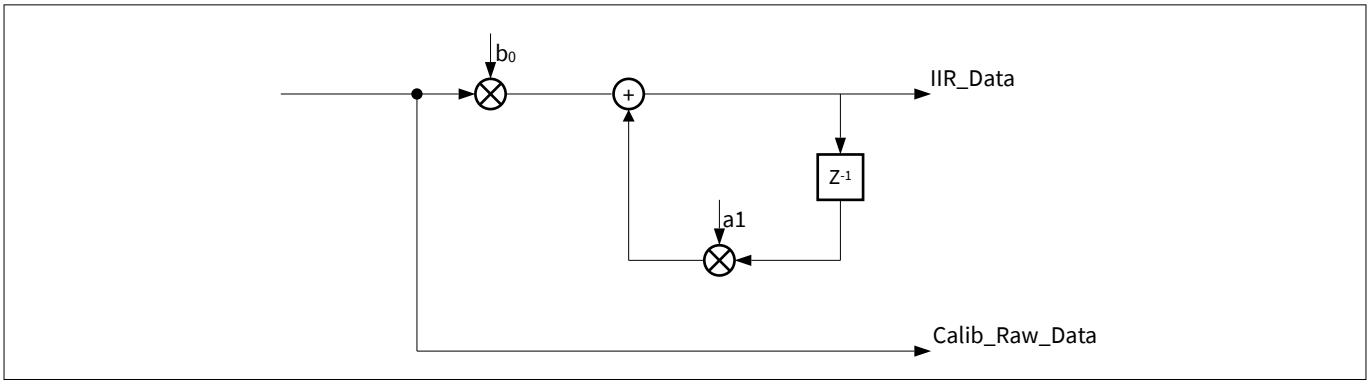


Figure 196 IIR-filter implementation structure

$$H_{\mathrm{IIR}}\left(z\right) = \frac{b_0}{1 - a_1 \times z^{-1}} \tag{25}$$

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:

$$b_0 = 1 - a_1 \tag{26}$$

With the coefficient b implemented in the IIR filter transfer function, it looks like:

$$H_{\mathrm{IIR}}\left(z\right) = \frac{1 - a_1}{\left(1 - a_1 \times z^{-1}\right)} \tag{27}$$

The IIR filter transfer function is shown in the plot below.

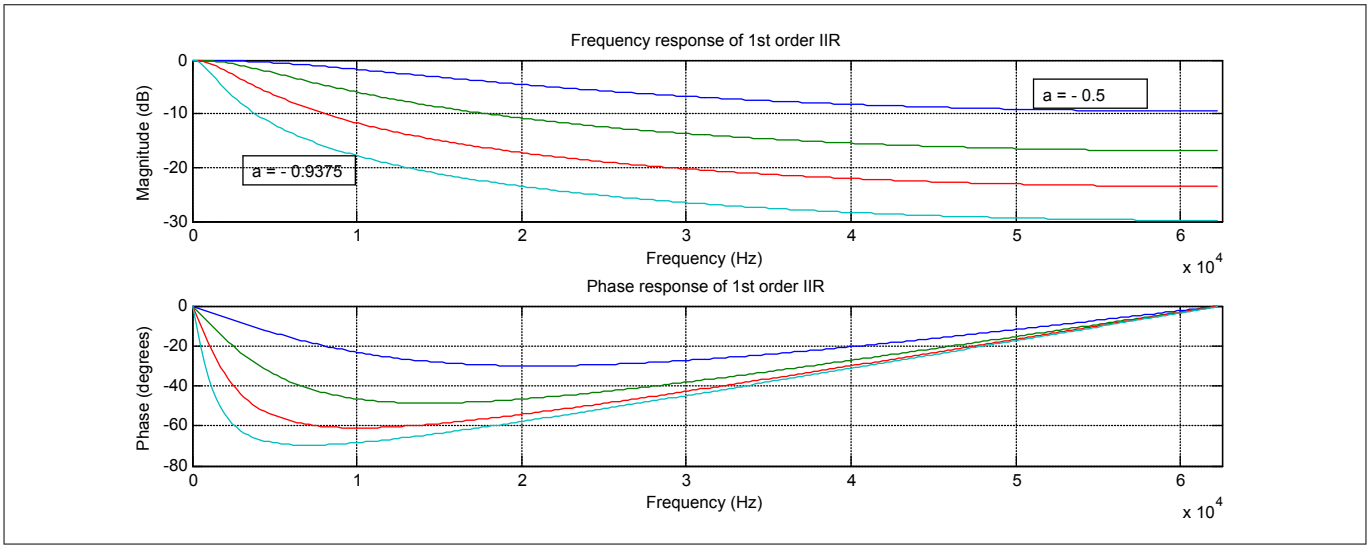


Figure 197 IIR filter transfer function for different filter length fl (1 MHz corresponds to 1/2*channel sampling frequency)

22.6.1.1 Step response

The step responseof the IIR filter time is shown in the following figure:

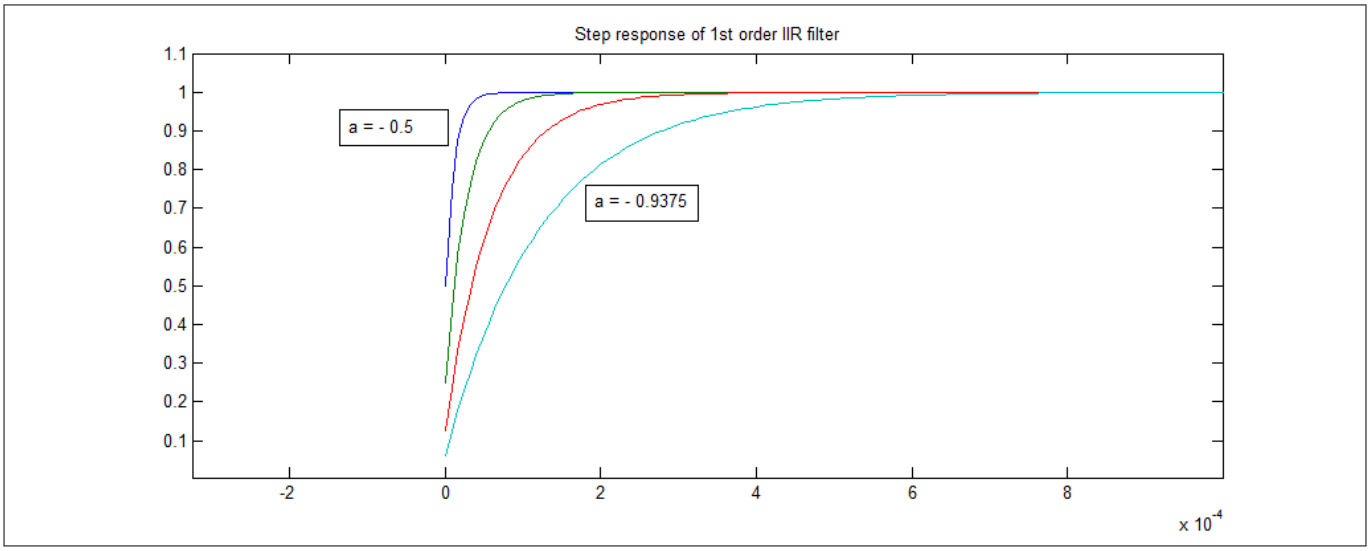


Figure 198 IIR step response Time

The following table summarizes the main filter characteristics.

Table 158 IIR filter characteristics

Filter coefficient	Group delay at = ω0
a	τ[samples]
-2^{-1}	2
-2^{-2}	4
-2^{-3}	8
-2^{-4}	16

22.6.2 IIR filter control registers

The IIR filter can also be configured by the SFR registers listed below.

The registers which cannot be written by the user have the attribute rwp.

The ADC2_FILT_OUT0 to ADC2_FILT_OUT6 registers are 10 bits wide, but the ADC delivers only a resolution of 8 bits. The following table shows how the lower two bits are determined.

Table 159 ADC2_FILT_OUT register setting

ADC2_CTRL1.calib_en	ADC2_CTRL4.filt_out_sel	ADC2_FILT_OUT0.output[1:0]
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 (see Chapter 22.5.1.2), the output is feed into the IIR filter. The internal result of the IIR filter is 12 bits (see Chapter 22.6.1), the output is converted to 10 bit and feed into the post processing. The user can monitor the calculated values in the ADC2_FILT_OUT0 to ADC_FILT_OUT6 registers and gets access to 10 bit wide result information.

The registers are addressed wordwise.

22.6.2.1 Register overview - IIR filter control registers (ascending offset address)

Table 160 Register overview - IIR filter control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC2_FILTCOEFF0_7	Filter coefficients ADC channel 0-7 register	0048_H	713
ADC2_FILT_OUT0	ADC or filter output channel 0 register	0050_H	715
ADC2_FILT_OUT1	ADC or filter output channel 1 register	0054_H	716
ADC2_FILT_OUT2	ADC or filter output channel 2 register	0058_H	717
ADC2_FILT_OUT3	ADC or filter output channel 3 register	$005C_H$	718
ADC2_FILT_OUT4	ADC or filter output channel 4 register	0060_H	719
ADC2_FILT_OUT5	ADC or filter output channel 5 register	0064_H	720
ADC2_FILT_OUT6	ADC or filter output channel 6 register	0068_H	721

22.7 Signal processing

22.7.1 Functional description

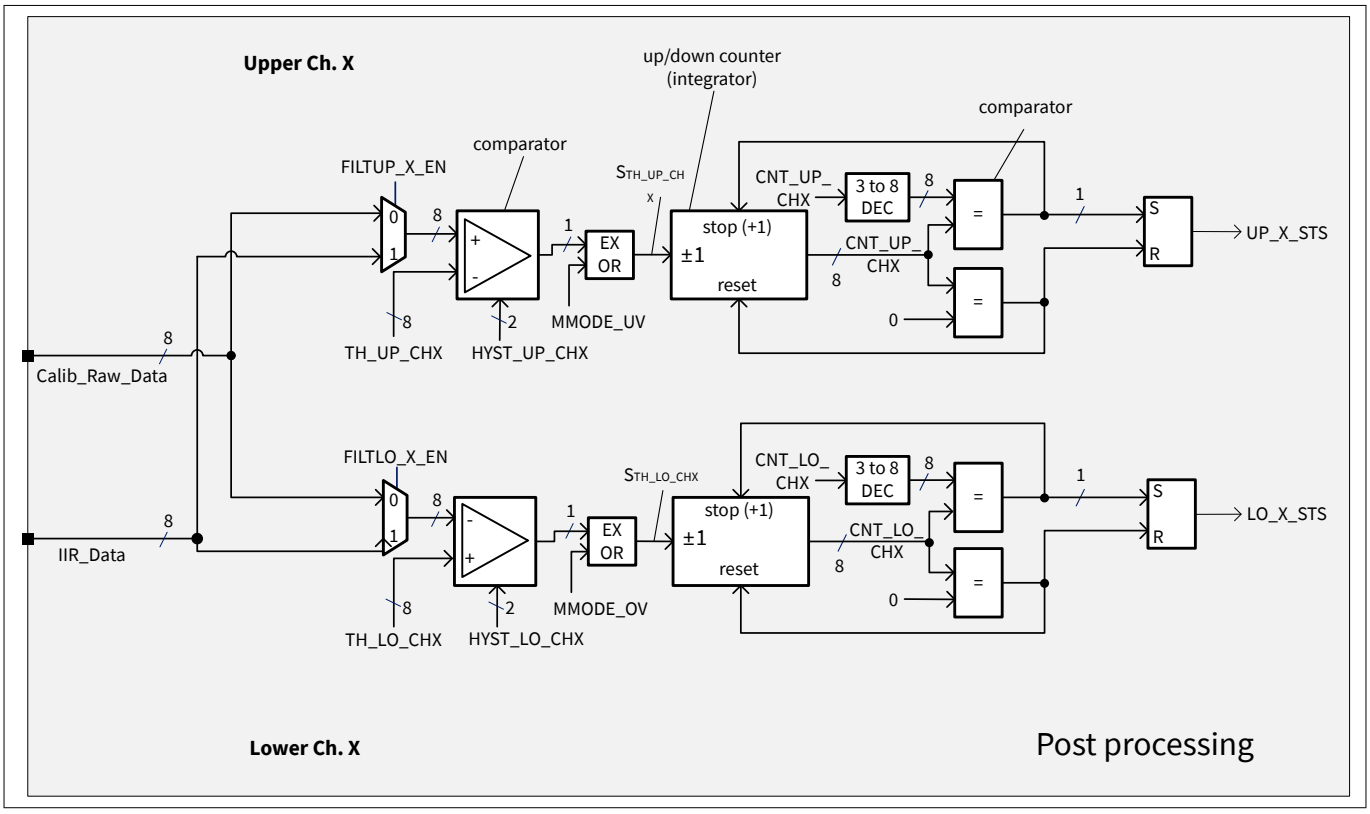


Figure 199 Post processing channel block diagram for voltage and temperature measurements

As shown in the figure below 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 TH_UP_CHX and a lower threshold TH_LO_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 HYST_UP_CHX and 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 TH_LO_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 $2^{CNT\_LO/HI\_CHX}$.

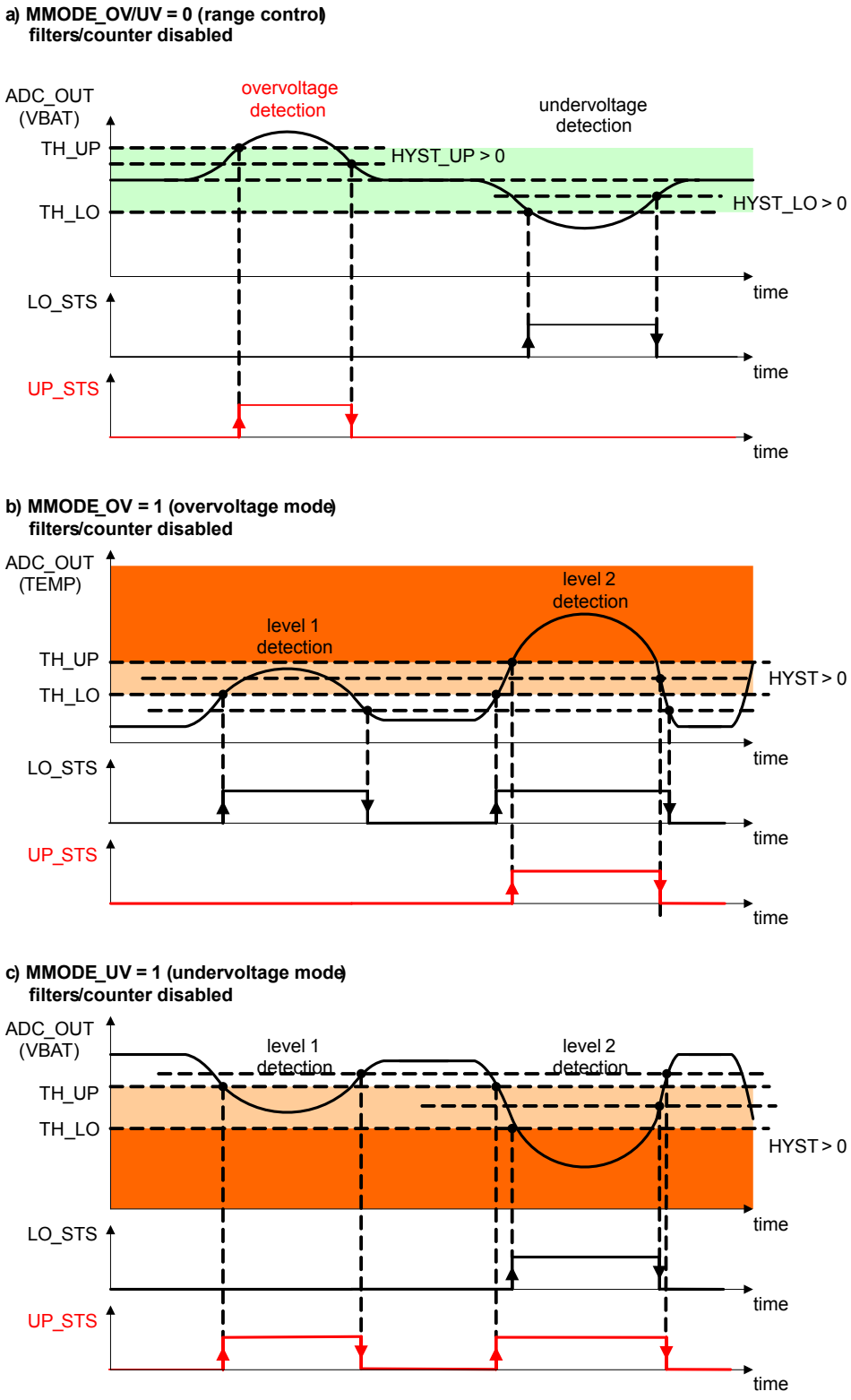


Figure 200 Measurement examples of a measurement Channel with disabled filters

The figure below shows three examples, an over- and undervoltage detection (e.g. VBAT_SENSE monitoring), a 2-step overvoltage and a 2-step undervoltage detection. The modes MMODE_OV/UV = 1 can be used as pre-warning for the application software (e.g. close to overtemperature or supply undervoltage).

22.7.2 Postprocessing control registers

The temperature sensor is fully controllable by the below listed SFR registers.
The registers are addressed wordwise.

22.7.2.1 Register overview - Postprocessing control registers (ascending offset address)

Table 161 Register overview - Postprocessing control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC2_FILT_UP_CTRL	Upper threshold filter enable register	0078_H	722
ADC2_FILT_LO_CTRL	Lower threshold filter enable register	$007C_H$	723
ADC2_TH0_3_LOWER	Lower comparator trigger level channel 0-3 register	0080_H	724
ADC2_TH4_7_LOWER	Lower comparator trigger level channel 4-7 register	0084_H	725
ADC2_TH0_3_UPPER	Upper comparator trigger level channel 0-3 register	$008C_H$	726
ADC2_TH4_7_UPPER	Upper comparator trigger level channel 4-7 register	0090_H	727
ADC2_CNT0_3_LOWER	Lower counter trigger level channel 0-3 register	0098_H	728
ADC2_CNT4_7_LOWER	Lower counter trigger level channel 4-7 register	$009C_H$	730
ADC2_CNT0_3_UPPER	Upper counter trigger level channel 0-3 register	$00A4_H$	732
ADC2_CNT4_7_UPPER	Upper counter trigger level channel 4-7 register	$00A8_H$	734
ADC2_MMODE0_7	Overvoltage measurement mode of channel 0-7 register	$00B0_H$	736

22.8 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 MI_RDY.

Measurement core start-up procedure: the start-up time of the complete signal chain are 2200 EoC cycles. The IIR-filter coefficient is set to C=2^-1 (fastest response time).

During the start-up phase, the DPP will use SQ=11_1111_1111, regardless of the sequence registers configuration.

22.9 Post processing default values

The following table shows the assigned measurements of the particular channels and the reset default values which read from FW 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 162 Channel allocation and post processing default settings (effective after reset)

Channel#	Name	Function	MMODE[1]	FILTCOEFF[2]		Threshold digital[3]	Threshold analog	Hysteresis[4]	Counters[5]
Ch. 0	VS	VS supply voltage	0	1	upper	$E8_H$	28.08 V	1	1
					lower	25_H	4.43 V	1	1
Ch. 1	VDDEXT	5 V supply voltage for external	0	1	upper	$E9_H$	5.42 V	1	1
					lower	BF_H	4.44 V	1	1
Ch. 2	VDDP	5 V port supply voltage	0	1	upper	$E9_H$	5.42 V	1	1
					lower	$6F_H$	2.58 V	1	1
Ch. 3	Vbg	Measures 1 V reference voltage from PMU	0	1	upper	AF_H or $B3_H$	1.10 V or 1.13 V	1	1
					lower	$8F_H$ or $8B_H$	0.90 V or 0.88 V	1	1
Ch. 4	VDDC	1.5 V core supply voltage	0	1	upper	FB_H	1.58 V	1	1
					lower	$D4_H$	1.33 V	1	1
Ch. 5	TEMP_LS	Temperature lowsides	2	1	upper	DD_H	180°C	1	1
					lower	CF_H	151°C	1	1
Ch. 6	TEMP_Central	Temperature central	2	1	upper	DD_H	180°C	1	3
					lower	$C2_H$	124°C	1	3

1) Register MMODE0_7; 0 = range control, 1 = UV, 2 = OV.
2) Register FILTCOEFF0_7; 0 = 1/2, 1 = 1/4, 2 = 1/8, 3 = 1/16.
3) Bitfield THUP_CHn / THLO_CHn.
4) Bitfield HYST_UP_CHn / HYST_LO_CHn; 0 = hyst off, 1 = hyst 4, 2 = hyst 8, 3 = hyst 16.
5) Bitfield CNT_UP_CHn / CNT_LO_CHn; 0 = 1 meas., 1 = 2 meas., 2 = 4 meas., 3 = 8 meas.

22.10 Measurement core module (incl. ADC2) register definition

22.10.1 Register address space - ADC2

Table 163 Registers address space - ADC2

Module	Base address	End address	Note
ADC2	$4801C000_H$	$4801DFFF_H$	Measurement core module (incl. ADC2) registers

22.10.2 Registers overview - ADC2 (sorted by name)

Table 164 Registers overview - ADC2 (sorted by name)

Short name	Long name	Offset address	Page number
ADC2_CAL_CH0_1	Calibration for channel 0 and 1 register	0034_H	709
ADC2_CAL_CH2_3	Calibration for channel 2 and 3 register	0038_H	710
ADC2_CAL_CH4_5	Calibration for channel 4 and 5 register	$003C_H$	711
ADC2_CAL_CH6_7	Calibration for channel 6 and 7 register	0040_H	712
ADC2_CHx_EIM	Channel settings bits for exceptional interrupt measurement register	0008_H	700
ADC2_CNT0_3_LOWER	Lower counter trigger level channel 0-3 register	0098_H	728
ADC2_CNT0_3_UPPER	Upper counter trigger level channel 0-3 register	$00A4_H$	732
ADC2_CNT4_7_LOWER	Lower counter trigger level channel 4-7 register	$009C_H$	730
ADC2_CNT4_7_UPPER	Upper counter trigger level channel 4-7 register	$00A8_H$	734
ADC2_CTRL1	Measurement unit control 1 register	0014_H	703
ADC2_CTRL2	Measurement unit control 2 register	0018_H	704
ADC2_CTRL4	Measurement unit control 4 register	$001C_H$	705
ADC2_CTRL_STS	ADC2 control and status register	0000_H	696
ADC2_FILTCOEFF0_7	Filter coefficients ADC channel 0-7 register	0048_H	713
ADC2_FILT_LO_CTRL	Lower threshold filter enable register	$007C_H$	723
ADC2_FILT_OUT0	ADC or filter output channel 0 register	0050_H	715
ADC2_FILT_OUT1	ADC or filter output channel 1 register	0054_H	716
ADC2_FILT_OUT2	ADC or filter output channel 2 register	0058_H	717
ADC2_FILT_OUT3	ADC or filter output channel 3 register	$005C_H$	718
ADC2_FILT_OUT4	ADC or filter output channel 4 register	0060_H	719
ADC2_FILT_OUT5	ADC or filter output channel 5 register	0064_H	720
ADC2_FILT_OUT6	ADC or filter output channel 6 register	0068_H	721
ADC2_FILT_UP_CTRL	Upper threshold filter enable register	0078_H	722

(table continues...)

Table 164 (continued) Registers overview - ADC2 (sorted by name)

Short name	Long name	Offset address	Page number
ADC2_MAX_TIME	Maximum time for software mode register	0010_H	702
ADC2_MMODE0_7	Overvoltage measurement mode of channel 0-7 register	$00B0_H$	736
ADC2_SQ1_4	Measurement channel enable bits for cycle 1-4 register	0020_H	706
ADC2_SQ5_8	Measurement channel enable bits for cycle 5-8 register	0024_H	708
ADC2_SQ_FB	Sequencer feedback register	0004_H	698
ADC2_STATUS	ADC2 HV status register	$00BC_H$	697
ADC2_TH0_3_LOWER	Lower comparator trigger level channel 0-3 register	0080_H	724
ADC2_TH0_3_UPPER	Upper comparator trigger level channel 0-3 register	$008C_H$	726
ADC2_TH4_7_LOWER	Lower comparator trigger level channel 4-7 register	0084_H	725
ADC2_TH4_7_UPPER	Upper comparator trigger level channel 4-7 register	0090_H	727

22.10.3 ADC2 control and status register

ADC2_CTRL_STS Offset address: 0000_H

ADC2 control and status register RESET_TYPE_3 value: $0000\ 0001_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				IN_MUX_SEL				RES				EOC	SOS	RES	RES1
r				rw				r				rh	rwh1	r	r

Field	Bits	Type	Description
RES1	0	r	**Reserved** Always read as 1.
RES	1, 7:4, 23:12, 24, 31:25	r	**Reserved** Always read as 0.
SOS	2	rwh1	**ADC2 start of sampling/conversion (software mode)** *Note: Bit is set by software to start sampling and 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
EOC	3	rh	**ADC2 end of conversion (software mode)** 0_B **PENDING**: Conversion still running 1_B **FINISHED**: Conversion has finished
IN_MUX_SEL	11:8	rw	**Channel for software mode** Other bit combinations are reserved, do not use. 0_H **CH0_EN**: Channel 0 enable 1_H **CH1_EN**: Channel 1 enable 2_H **CH2_EN**: Channel 2 enable 3_H **CH3_EN**: Channel 3 enable 4_H **CH4_EN**: Channel 4 enable 5_H **CH5_EN**: Channel 5 enable 6_H **CH6_EN**: Channel 6 enable 7_H **RFU**: Reserved for future use … F_H **RFU**: Reserved for future use

22.10.4 ADC2 HV status register

ADC2_STATUS Offset address: $00BC_H$

ADC2 HV status register RESET_TYPE_3 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	0, 31:2	r	**Reserved** Always read as 0.
READY	1	r	**HVADC 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

22.10.5 Sequencer feedback register

ADC2_SQ_FB Offset address: 0004_H

Sequencer feedback register RESET_TYPE_3 value: $00XX\ XX0X_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		SQx_STS			RES	EIM_ACTIVE	SQ_STOP	RES				SQ_FB			
r		r			r	r	r	r				r			

Field	Bits	Type	Description
SQ_FB	3:0	r	**Current sequence that caused software mode** Other bit combinations are n.u., not used. 0_H **SQ1**: Sequence 1 1_H **SQ2**: Sequence 2 2_H **SQ3**: Sequence 3 3_H **SQ4**: Sequence 4 4_H **SQ5**: Sequence 5 5_H **SQ6**: Sequence 6 6_H **SQ7**: Sequence 7 A_H **RES**: Reserved B_H **CH_MASK**: Channel mask = 0 C_H **SUSPEND**: Debug suspend mode
RES	7:4, 10, 15:14, 31:20	r	**Reserved** Always read as 0.
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/software mode entered
EIM_ACTIVE	9	r	**ADC2 EIM active** 0_B **NOT_ACTIVE**: EIM not active 1_B **ACTIVE**: EIM active
SQx_STS	13:11	r	**Current active ADC2 sequence** Other bit combinations are reserved, do not use. 000_B **SQ1**: Sequence 1 enable 001_B **SQ2**: Sequence 2 enable 010_B **SQ3**: Sequence 3 enable 011_B **SQ4**: Sequence 4 enable

(table continues...)

(continued)

Field	Bits	Type	Description
			100_B **SQ5**: Sequence 5 enable 101_B **SQ6**: Sequence 6 enable 110_B **SQ7**: Sequence 7 enable
CHx_STS	19:16	r	**Current ADC2 channel** Other bit combinations are reserved, do not use. 0_H **CH0**: Channel 0 enable 1_H **CH1**: Channel 1 enable 2_H **CH2**: Channel 2 enable 3_H **CH3**: Channel 3 enable 4_H **CH4**: Channel 4 enable 5_H **CH5**: Channel 5 enable 6_H **CH6**: Channel 6 enable

22.10.6 Channel settings bits for exceptional interrupt measurement register

ADC2_CHx_EIM

Channel settings bits for exceptional interrupt measurement register

Offset address: 0008_H

RESET_TYPE_3 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_SEL		
r			rw	rw	rw			r					rw		

Field	Bits	Type	Description
CHx_SEL	2:0	rw	**Channel set for exceptional interrupt measurement (EIM)** Other bit combinations are n.u., not used. 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 **RFU**: Reserved for future use
RES	7:3, 31:13	r	**Reserved** Always read as 0.
REP	10:8	rw	**Repeat count for exceptional interrupt measurement (EIM)** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **64**: 64 measurements 111_B **128**: 128 measurements

(table continues...)

(continued)

Field	Bits	Type	Description
EN	11	rw	**Exceptional interrupt measurement (EIM) trigger event enable** 0_B **DISABLE**: Start of EIM disabled 1_B **ENABLE**: Start of EIM enabled
SEL	12	rw	**Exceptional interrupt measurement (EIM) trigger select** 0_B **TRIGGERS**: GPT12PISEL.T3_GPT12_SEL, GPT12_PISEL triggers EIM 1_B **NOT_SUPPORTED**: Not supported

22.10.7 Maximum time for software mode register

ADC2_MAX_TIME

Offset address: 0010_H

Maximum time for software mode register

RESET_TYPE_3 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								MAX_TIME							
r								rw							

Field	Bits	Type	Description
MAX_TIME	7:0	rw	**Maximum time in software mode** Maximum time in software mode with the unit of 50 ns. Software mode is active for ADC2_MAX_TIME * 50 ns. 00_H **MIN**: Software mode is immediately left FF_H **MAX**: Software mode is active for 12.75 us
RES	31:8	r	**Reserved** Always read as 0.

22.10.8 Measurement unit control 1 register

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.

ADC2_CTRL1 Offset address: 0014_H

Measurement unit control 1 register RESET_TYPE_4 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									**CALIB_EN_6_0**						
r									rwpt						

Field	Bits	Type	Description
CALIB_EN_6_0	6:0	rwpt	**Calibration enable for channels 6 to 0** The following values can be ored: 01_H **CH0_EN**: Channel 0 calibration enable 02_H **CH1_EN**: Channel 1 calibration enable 04_H **CH2_EN**: Channel 2 calibration enable 08_H **CH3_EN**: Channel 3 calibration enable 10_H **CH4_EN**: Channel 4 calibration enable 20_H **CH5_EN**: Channel 5 calibration enable 40_H **CH6_EN**: Channel 6 calibration enable
RES	31:7	r	**Reserved** Always read as 0.

22.10.9 Measurement unit control 2 register

This register is used for controlling the calibration unit of channels 0-7 of the measurement core module. This registers are protected for the purpose mentioned at the beginning of this chapter. Furthermore this register contains the sample time adjustment for ADC2. The default value is 14 clock cycles. Values above 14 clock cycles are not recommended, because they increase the overall response time of the measurement system.

ADC2_CTRL2 Offset address: 0018_H

Measurement unit control 2 register RESET_TYPE_4 value: $0000\ 0401_H$

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				SAMPLE_TIME_int				MCM_RDY	RES		RES				MCM_PD_N
r				rw				r	r		r				rwpt

Field	Bits	Type	Description
MCM_PD_N	0	rwpt	**Power down signal for MCM** 0_B **DISABLED**: Measurement core module disabled 1_B **ENABLED**: Measurement core module enabled
RES	4:1, 6:5, 15:12, 31:16	r	**Reserved** Always read as 0.
MCM_RDY	7	r	**Ready signal for MCM** Measurement core module after power-on or reset. 0_B **NOT_READY**: Measurement core module in start-up phase 1_B **READY**: Measurement core module start-up phase finished
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 (default) 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 **NU**: Not used … F_H **NU**: Not used

22.10.10 Measurement unit control 4 register

ADC2_CTRL4 Offset address: $001C_H$

Measurement unit control 4 register RESET_TYPE_4 value: $0000\ 007F_H$

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	FILT_OUT_SEL_6_0						
r						r		r	rw						

Field	Bits	Type	Description
FILT_OUT_SEL_6_0	6:0	rw	**Output filter selection for channels 0 to 6** 00_H **UNFILTERED**: ADC2 unfiltered data can be monitored in the corresponding ADC2_FILT_OUTx registers 01_H **CH_0**: Channel 0 IIR data enabled for ADC2_FILT_OUT0 register 02_H **CH_1**: Channel 1 IIR data enabled for ADC2_FILT_OUT1 register 04_H **CH_2**: Channel 2 IIR data enabled for ADC2_FILT_OUT2 register 08_H **CH_3**: Channel 3 IIR data enabled for ADC2_FILT_OUT3 register 10_H **CH_4**: Channel 4 IIR data enabled for ADC2_FILT_OUT4 register 20_H **CH_5**: Channel 5 IIR data enabled for ADC2_FILT_OUT5 register 40_H **CH_6**: Channel 6 IIR data enabled for ADC2_FILT_OUT6 register $7F_H$ **ALL**: For channels 6-0 IIR data is enabled for ADC2_FILT_OUTx registers
RES	7, 9:8, 31:10	r	**Reserved** Always read as 0.

22.10.11 Measurement channel enable bits for cycle 1-4 register

ADC2_SQ1_4 Offset address: 0020_H

Measurement channel enable bits for cycle 1-4 register RESET_TYPE_4 value: $4936\ 4837_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**SQ4**							**RES**	**SQ3**						
r	rwpt							r	rwpt						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**SQ2**							**RES**	**SQ1**						
r	rwpt							r	rwpt						

Field	Bits	Type	Description
SQ1	6:0	rwpt	**Sequence 1 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable
RES	7, 15, 23, 31	r	**Reserved** Always read as 0.
SQ2	14:8	rwpt	**Sequence 2 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable
SQ3	22:16	rwpt	**Sequence 3 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable

(table continues...)

(continued)

Field	Bits	Type	Description
SQ4	30:24	rwpt	**Sequence 4 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable

22.10.12 Measurement channel enable bits for cycle 5-8 register

ADC2_SQ5_8 Offset address: 0024_H

Measurement channel enable bits for cycle 5-8 register RESET_TYPE_4 value: $0037\ 4836_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES									SQ7						
r									rwpt						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	SQ6							RES	SQ5						
r	rwpt							r	rwpt						

Field	Bits	Type	Description
SQ5	6:0	rwpt	**Sequence 5 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable
RES	7, 15, 31:23	r	**Reserved** Always read as 0.
SQ6	14:8	rwpt	**Sequence 6 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable
SQ7	22:16	rwpt	**Sequence 7 channel enable** The following values can be ored: 01_H **CH0_EN**: Channel 0 enable 02_H **CH1_EN**: Channel 1 enable 04_H **CH2_EN**: Channel 2 enable 08_H **CH3_EN**: Channel 3 enable 10_H **CH4_EN**: Channel 4 enable 20_H **CH5_EN**: Channel 5 enable 40_H **CH6_EN**: Channel 6 enable

22.10.13 Calibration for channel 0 and 1 register

ADC2_CAL_CH0_1 Offset address: 0034_H

Calibration for channel 0 and 1 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH1								**RES**			**OFFS_CH1**				
rwpt								r			rwpt				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH0								**RES**			**OFFS_CH0**				
rwpt								r			rwpt				

Field	Bits	Type	Description
OFFS_CH0	4:0	rwpt	**Offset calibration for channel 0** For ADC output, set CALIB_EN_0 = 0.
RES	7:5, 23:21	r	**Reserved** Always read as 0.
GAIN_CH0	15:8	rwpt	**Gain calibration for channel 0** For ADC output, set CALIB_EN_0 = 0.
OFFS_CH1	20:16	rwpt	**Offset calibration for channel 1** For ADC output, set CALIB_EN_1 = 0.
GAIN_CH1	31:24	rwpt	**Gain calibration for channel 1** For ADC output, set CALIB_EN_1 = 0.

22.10.14 Calibration for channel 2 and 3 register

ADC2_CAL_CH2_3 Offset address: 0038_H

Calibration for channel 2 and 3 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH3								**RES**			**OFFS_CH3**				
rwpt								r			rwpt				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH2								**RES**			**OFFS_CH2**				
rwpt								r			rwpt				

Field	Bits	Type	Description
OFFS_CH2	4:0	rwpt	**Offset calibration for channel 2** For ADC output, set CALIB_EN_2 = 0.
RES	7:5, 23:21	r	**Reserved** Always read as 0.
GAIN_CH2	15:8	rwpt	**Gain calibration for channel 2** For ADC output, set CALIB_EN_2 = 0.
OFFS_CH3	20:16	rwpt	**Offset calibration for channel 3** For ADC output, set CALIB_EN_3 = 0.
GAIN_CH3	31:24	rwpt	**Gain calibration for channel 3** For ADC output, set CALIB_EN_3 = 0.

22.10.15 Calibration for channel 4 and 5 register

ADC2_CAL_CH4_5 Offset address: $003C_H$

Calibration for channel 4 and 5 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GAIN_CH5								**RES**			**OFFS_CH5**				
rwpt								r			rwpt				

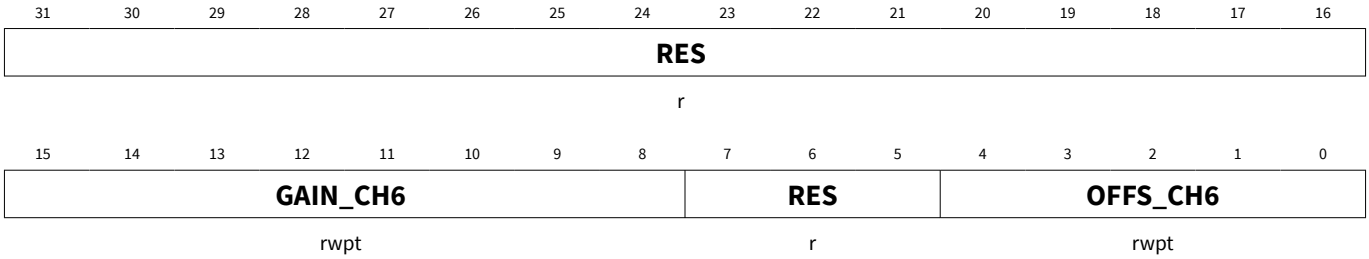
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GAIN_CH4								**RES**			**OFFS_CH4**				
rwpt								r			rwpt				

Field	Bits	Type	Description
OFFS_CH4	4:0	rwpt	**Offset calibration for channel 4** For ADC output, set CALIB_EN_4 = 0.
RES	7:5, 23:21	r	**Reserved** Always read as 0.
GAIN_CH4	15:8	rwpt	**Gain calibration for channel 4** For ADC output, set CALIB_EN_4 = 0.
OFFS_CH5	20:16	rwpt	**Offset calibration for channel 5** For ADC output, set CALIB_EN_5 = 0.
GAIN_CH5	31:24	rwpt	**Gain calibration for channel 5** For ADC output, set CALIB_EN_5 = 0.

22.10.16 Calibration for channel 6 and 7 register

ADC2_CAL_CH6_7 Offset address: 0040_H

Calibration for channel 6 and 7 register RESET_TYPE_4 value: $0000\ 0000_H$



Field	Bits	Type	Description
OFFS_CH6	4:0	rwpt	**Offset calibration for channel 6** For ADC output, set CALIB_EN_6 = 0.
RES	7:5, 31:16	r	**Reserved** Always read as 0.
GAIN_CH6	15:8	rwpt	**Gain calibration for channel 6** For ADC output, set CALIB_EN_6 = 0.

22.10.17 Filter coefficients ADC channel 0-7 register

ADC2_FILTCOEFF0_7 Offset address: 0048_H

Filter coefficients ADC channel 0-7 register RESET_TYPE_4 value: $0000\ 1555_H$

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
A_CH7		A_CH6		A_CH5		A_CH4		A_CH3		A_CH2		A_CH1		A_CH0	
rwpt		rwpt		rwpt		rwpt		rwpt		rwpt		rwpt		rwpt	

Field	Bits	Type	Description
A_CH0	1:0	rwpt	**Filter coefficient A for ADC channel 0** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
A_CH1	3:2	rwpt	**Filter coefficient A for ADC channel 1** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
A_CH2	5:4	rwpt	**Filter coefficient A for ADC channel 2** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
A_CH3	7:6	rwpt	**Filter coefficient A for ADC channel 3** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
A_CH4	9:8	rwpt	**Filter coefficient A for ADC channel 4** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
A_CH5	11:10	rwpt	**Filter coefficient A for ADC channel 5** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample

(table continues...)

(continued)

Field	Bits	Type	Description
A_CH6	13:12	rwpt	**Filter coefficient A for ADC channel 6** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
A_CH7	15:14	rwpt	**Filter coefficient A for ADC channel 7** *Note: These bits are dedicated for future use. They are always read as 0.* 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
RES	31:16	r	**Reserved** Always read as 0.

22.10.18 ADC or filter output channel 0 register

This registers reflects the current value of channel 0 of the measurement chain, which is assigned to VBAT_SENSE measurement.

ADC2_FILT_OUT0 Offset address: 0050_H

ADC or filter output channel 0 register RESET_TYPE_3 value: $0000\ 0XXX_H$

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

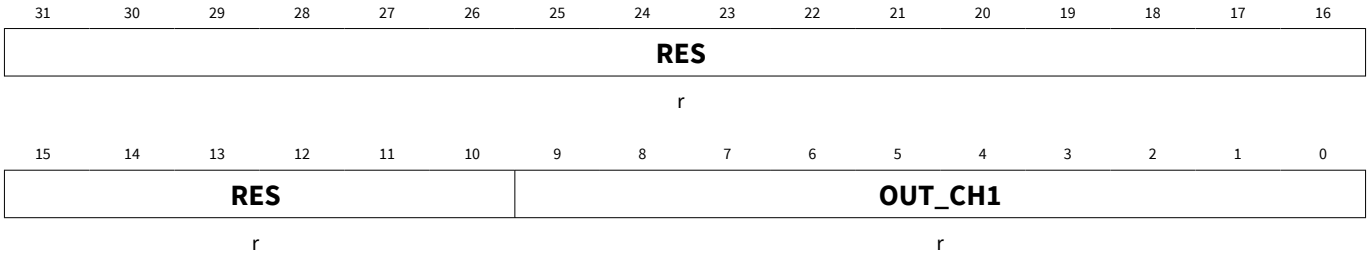
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						OUT_CH0									
r						r									

Field	Bits	Type	Description
OUT_CH0	9:0	r	**ADC or filter output value channel 0** For ADC output, set ADC2_FILTUP_0_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

22.10.19 ADC or filter output channel 1 register

ADC2_FILT_OUT1 Offset address: 0054_H

ADC or filter output channel 1 register RESET_TYPE_3 value: $0000\ 0XXX_H$



Field	Bits	Type	Description
OUT_CH1	9:0	r	**ADC or filter output value channel 1** For ADC output, set ADC2_FILTUP_1_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

22.10.20 ADC or filter output channel 2 register

ADC2_FILT_OUT2 Offset address: 0058_H

ADC or filter output channel 2 register RESET_TYPE_3 value: $0000\ 0XXX_H$

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

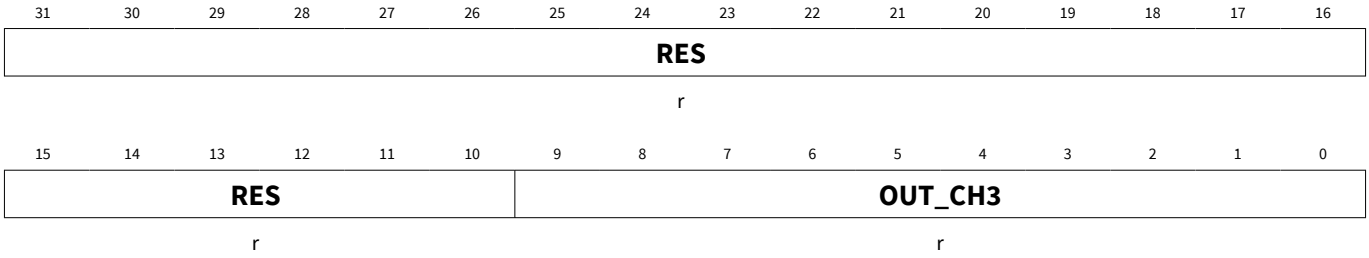
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES						**OUT_CH2**									
r						r									

Field	Bits	Type	Description
OUT_CH2	9:0	r	**ADC or filter output value channel 2** For ADC output, set ADC2_FILTUP_2_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

22.10.21 ADC or filter output channel 3 register

ADC2_FILT_OUT3 Offset address: $005C_H$

ADC or filter output channel 3 register RESET_TYPE_3 value: $0000\ 0XXX_H$

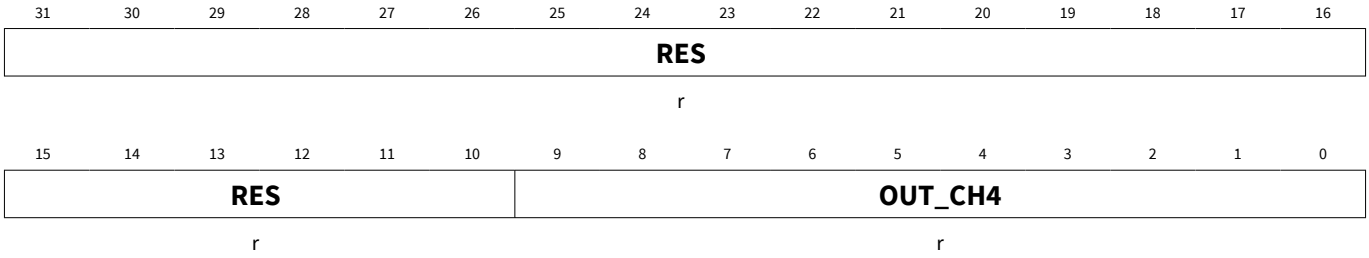


Field	Bits	Type	Description
OUT_CH3	9:0	r	**ADC or filter output value channel 3** For ADC output, set ADC2_FILTUP_3_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

22.10.22 ADC or filter output channel 4 register

ADC2_FILT_OUT4 Offset address: 0060_H

ADC or filter output channel 4 register RESET_TYPE_3 value: $0000\ 0XXX_H$



Field	Bits	Type	Description
OUT_CH4	9:0	r	**ADC or filter output value channel 4** For ADC output, set ADC2_FILTUP_4_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

22.10.23 ADC or filter output channel 5 register

ADC2_FILT_OUT5 Offset address: 0064_H

ADC or filter output channel 5 register RESET_TYPE_3 value: $0000\ 0XXX_H$

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						**OUT_CH5**									
r						r									

Field	Bits	Type	Description
OUT_CH5	9:0	r	**ADC or filter output value channel 5** For ADC output, set ADC2_FILTUP_5_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

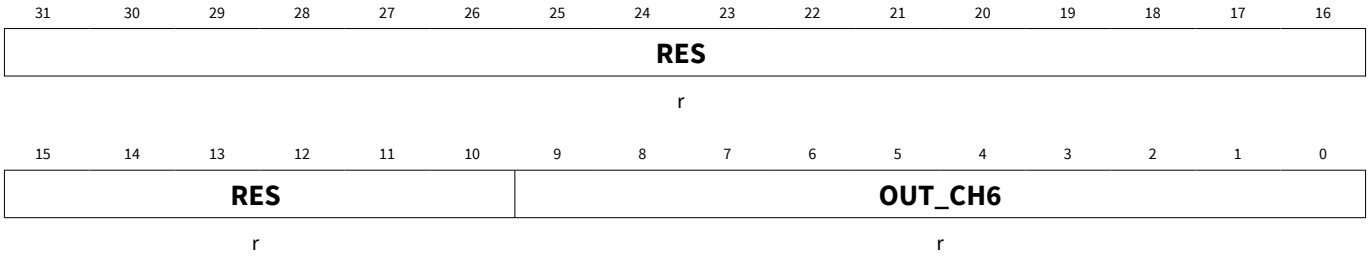
22.10.24 ADC or filter output channel 6 register

ADC2_FILT_OUT6

Offset address: 0068_H

ADC or filter output channel 6 register

RESET_TYPE_3 value: $0000\ 0XXX_H$



Field	Bits	Type	Description
OUT_CH6	9:0	r	**ADC or filter output value channel 6** For ADC output, set ADC2_FILTUP_6_EN = 0.
RES	31:10	r	**Reserved** Always read as 0.

22.10.25 Upper threshold filter enable register

ADC2_FILT_UP_CTRL Offset address: 0078_H

Upper threshold filter enable register RESET_TYPE_4 value: $0000\ 007F_H$

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									UPEN_Ch6	UPEN_Ch5	UPEN_Ch4	UPEN_Ch3	UPEN_Ch2	UPEN_Ch1	UPEN_Ch0
r									rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
UPEN_Ch0	0	rw	**Upper threshold IIR filter enable ch 0** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
UPEN_Ch1	1	rw	**Upper threshold IIR filter enable ch 1** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
UPEN_Ch2	2	rw	**Upper threshold IIR filter enable ch 2** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
UPEN_Ch3	3	rw	**Upper threshold IIR filter enable ch 3** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
UPEN_Ch4	4	rw	**Upper threshold IIR filter enable ch 4** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
UPEN_Ch5	5	rw	**Upper threshold IIR filter enable ch 5** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
UPEN_Ch6	6	rw	**Upper threshold IIR filter enable ch 6** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
RES	31:7	r	**Reserved** Always read as 0.

22.10.26 Lower threshold filter enable register

Setting this register enables the IIR filter structure for the post processing of the lower threshold. This can be used for example as shutdown signal for the system, in case of supply loss.

ADC2_FILT_LO_CTRL Offset address: $007C_H$

Lower threshold filter enable register RESET_TYPE_4 value: $0000\ 007F_H$

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									LOEN_Ch6	LOEN_Ch5	LOEN_Ch4	LOEN_Ch3	LOEN_Ch2	LOEN_Ch1	LOEN_Ch0
r									rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
LOEN_Ch0	0	rw	**Lower threshold IIR filter enable ch 0** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
LOEN_Ch1	1	rw	**Lower threshold IIR filter enable ch 1** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
LOEN_Ch2	2	rw	**Lower threshold IIR filter enable ch 2** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
LOEN_Ch3	3	rw	**Lower threshold IIR filter enable ch 3** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
LOEN_Ch4	4	rw	**Lower threshold IIR filter enable ch 4** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
LOEN_Ch5	5	rw	**Lower threshold IIR filter enable ch 5** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
LOEN_Ch6	6	rw	**Lower threshold IIR filter enable ch 6** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
RES	31:7	r	**Reserved** Always read as 0.

22.10.27 Lower comparator trigger level channel 0-3 register

ADC2_TH0_3_LOWER Offset address: 0080_H

Lower comparator trigger level channel 0-3 register RESET_TYPE_4 value: $9D6F\ BF25_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THLO_CH3								**THLO_CH2**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THLO_CH1								**THLO_CH0**							
rw								rw							

Field	Bits	Type	Description
THLO_CH0	7:0	rw	**Channel 0 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
THLO_CH1	15:8	rw	**Channel 1 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
THLO_CH2	23:16	rw	**Channel 2 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
THLO_CH3	31:24	rw	**Channel 3 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value

22.10.28 Lower comparator trigger level channel 4-7 register

ADC2_TH4_7_LOWER Offset address: 0084_H

Lower comparator trigger level channel 4-7 register RESET_TYPE_4 value: $00C8\ D4D4_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THLO_CH7								**THLO_CH6**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THLO_CH5								**THLO_CH4**							
rw								rw							

Field	Bits	Type	Description
THLO_CH4	7:0	rw	**Channel 4 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
THLO_CH5	15:8	rw	**Channel 5 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
THLO_CH6	23:16	rw	**Channel 6 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
THLO_CH7	31:24	rw	**Channel 7 lower trigger level** *Note: These bits are dedicated for future use. They are always read as 0.* 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value

22.10.29 Upper comparator trigger level channel 0-3 register

ADC2_TH0_3_UPPER Offset address: $008C_H$

Upper comparator trigger level channel 0-3 register RESET_TYPE_4 value: $EBE9\ E9E4_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THUP_CH3								**THUP_CH2**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THUP_CH1								**THUP_CH0**							
rw								rw							

Field	Bits	Type	Description
THUP_CH0	7:0	rw	**Channel 0 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
THUP_CH1	15:8	rw	**Channel 1 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
THUP_CH2	23:16	rw	**Channel 2 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
THUP_CH3	31:24	rw	**Channel 3 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255

22.10.30 Upper comparator trigger level channel 4-7 register

ADC2_TH4_7_UPPER Offset address: 0090_H

Upper comparator trigger level channel 4-7 register RESET_TYPE_4 value: $00E2\ E2FB_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
THUP_CH7								**THUP_CH6**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
THUP_CH5								**THUP_CH4**							
rw								rw							

Field	Bits	Type	Description
THUP_CH4	7:0	rw	**Channel 4 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
THUP_CH5	15:8	rw	**Channel 5 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
THUP_CH6	23:16	rw	**Channel 6 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
THUP_CH7	31:24	rw	**Channel 7 upper trigger level** *Note: These bits are dedicated for future use. They are always read as 0.* 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255

22.10.31 Lower counter trigger level channel 0-3 register

ADC2_CNT0_3_LOWER Offset address: 0098_H

Lower counter trigger level channel 0-3 register RESET_TYPE_4 value: $0909\ 0909_H$

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
CNT_LO_CH0	2:0	rw	**Lower timer trigger threshold channel 0** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH0	4:3	rw	**Channel 0 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0.
CNT_LO_CH1	10:8	rw	**Lower timer trigger threshold channel 1** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH1	12:11	rw	**Channel 1 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH2	18:16	rw	**Lower timer trigger threshold channel 2** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH2	20:19	rw	**Channel 2 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH3	26:24	rw	**Lower timer trigger threshold channel 3** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH3	28:27	rw	**Channel 3 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

22.10.32 Lower counter trigger level channel 4-7 register

ADC2_CNT4_7_LOWER Offset address: $009C_H$

Lower counter trigger level channel 4-7 register RESET_TYPE_4 value: $000B\ 0909_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_LO_CH7		CNT_LO_CH7			RES			HYST_LO_CH6		CNT_LO_CH6		
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_CH5		CNT_LO_CH5			RES			HYST_LO_CH4		CNT_LO_CH4		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_LO_CH4	2:0	rw	**Lower timer trigger threshold channel 4** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH4	4:3	rw	**Channel 4 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0.
CNT_LO_CH5	10:8	rw	**Lower timer trigger threshold channel 5** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH5	12:11	rw	**Channel 5 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH6	18:16	rw	**Lower timer trigger threshold channel 6** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH6	20:19	rw	**Channel 6 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH7	26:24	rw	**Lower timer trigger threshold channel 6** *Note: These bits are dedicated for future use. They are always read as 0.* 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH7	28:27	rw	**Channel 6 lower hysteresis** *Note: These bits are dedicated for future use. They are always read as 0.* 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

22.10.33 Upper counter trigger level channel 0-3 register

ADC2_CNT0_3_UPPER Offset address: $00A4_H$

Upper counter trigger level channel 0-3 register RESET_TYPE_4 value: $0909\ 0909_H$

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
CNT_UP_CH0	2:0	rw	**Upper timer trigger threshold channel 0** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH0	4:3	rw	**Channel 0 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0.
CNT_UP_CH1	10:8	rw	**Upper timer trigger threshold channel 1** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH1	12:11	rw	**Channel 1 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH2	18:16	rw	**Upper timer trigger threshold channel 2** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH2	20:19	rw	**Channel 2 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH3	26:24	rw	**Upper timer trigger threshold channel 3** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH3	28:27	rw	**Channel 3 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

22.10.34 Upper counter trigger level channel 4-7 register

ADC2_CNT4_7_UPPER Offset address: $00A8_H$

Upper counter trigger level channel 4-7 register RESET_TYPE_4 value: $000B\ 0909_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_UP_CH7		CNT_UP_CH7			RES			HYST_UP_CH6		CNT_UP_CH6		
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_CH5		CNT_UP_CH5			RES			HYST_UP_CH4		CNT_UP_CH4		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_UP_CH4	2:0	rw	**Upper timer trigger threshold channel 4** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH4	4:3	rw	**Channel 4 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0.
CNT_UP_CH5	10:8	rw	**Upper timer trigger threshold channel 5** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH5	12:11	rw	**Channel 5 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH6	18:16	rw	**Upper timer trigger threshold channel 6** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH6	20:19	rw	**Channel 6 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH7	26:24	rw	**Upper timer trigger threshold channel 7** *Note: These bits are dedicated for future use. They are always read as 0.* 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH7	28:27	rw	**Channel 7 upper hysteresis** *Note: These bits are dedicated for future use. They are always read as 0.* 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 Hysteresis = 8 11_B **HYST16**: Hysteresis = 16 Hysteresis = 16

22.10.35 Overvoltage measurement mode of channel 0-7 register

ADC2_MMODE0_7

Overvoltage measurement mode of channel 0-7 register

Offset address: $00B0_H$

RESET_TYPE_4 value: $0000\ 2800_H$

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
MSEL_Ch7		MSEL_Ch6		MSEL_Ch5		MSEL_Ch4		MSEL_Ch3		MSEL_Ch2		MSEL_Ch1		MSEL_Ch0	
rwpt		rwpt		rwpt		rwpt		rwpt		rwpt		rwpt		rwpt	

Field	Bits	Type	Description
MSEL_Ch0	1:0	rwpt	**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
MSEL_Ch1	3:2	rwpt	**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
MSEL_Ch2	5:4	rwpt	**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
MSEL_Ch3	7:6	rwpt	**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
MSEL_Ch4	9:8	rwpt	**Measurement mode ch 4** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 10_B **RESERVED**: Reserved
MSEL_Ch5	11:10	rwpt	**Measurement mode ch 5** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement

(table continues...)

(continued)

Field	Bits	Type	Description
			11_B **RESERVED**: Reserved Reserved
MSEL_Ch6	13:12	rwpt	**Measurement mode ch 6** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved Reserved
MSEL_Ch7	15:14	rwpt	**Measurement mode ch 7** *Note: These bits are dedicated for future use. They are always read as 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 Reserved
RES	31:16	r	**Reserved** Always read as 0.

23 10-bit analog digital converter (ADC1)

23.1 Features

The basic function of this block is the digital post-processing of several analog digitized measurement signals by means of filtering, level comparison and interrupt generation. The measurement post-processing block is built of twelve identical channel units attached to the outputs of the 13-channel 10-bit ADC. It processes twelve channels, where the channel sequence and prioritization is programmable within a wide range.

Functional features

- 10-bit SAR ADC with conversion time of 17 clock cycles
- Programmable clock divider for sequencer and ADC
- 12 individually programmable channels (ch0…ch11):
  - 6 or 7 (product variant dependent) HV channels: VS, VBAT_SENSE, MON1…MON4 or MON5 (product variant dependent)
  - 5 or 6 (product variant dependent) LV channels: P2.1, P2.2, P2.3, P2.6, P2.7, P2.0 (product variant dependent)
- One additional channel, ch12, connected to P2.0 (product variant dependent). This channel is only programmable in software mode, no calibration and no digital post-processing are available in this case
- All channels are fully calibrated and user configurable
- Individually programmable channel prioritization scheme for digital post-processing (dpp)
- 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 upper threshold and lower threshold interrupts and status for all channel thresholds
- Four additional differential channels (build with MON1 to MON4) with post-processing and interrupt generation (product variant dependent, only TLE9845QX)
- ADC reference completely integrated

Note: In case the MONx should be evaluated by the ADC1, it is recommended to add 6.8 nF capacitors close to the MONx pin of the device, in order to build an external RC filter to limit the bandwidth of the input signal.

23.2 Introduction

The basic function of this unit, is the digital signal processing of several analog digitized measurement signals by means of filtering, level comparison and interrupt generation. The measurement core module processes twelve channels in a quasi parallel process.

23.2.1 Block diagram

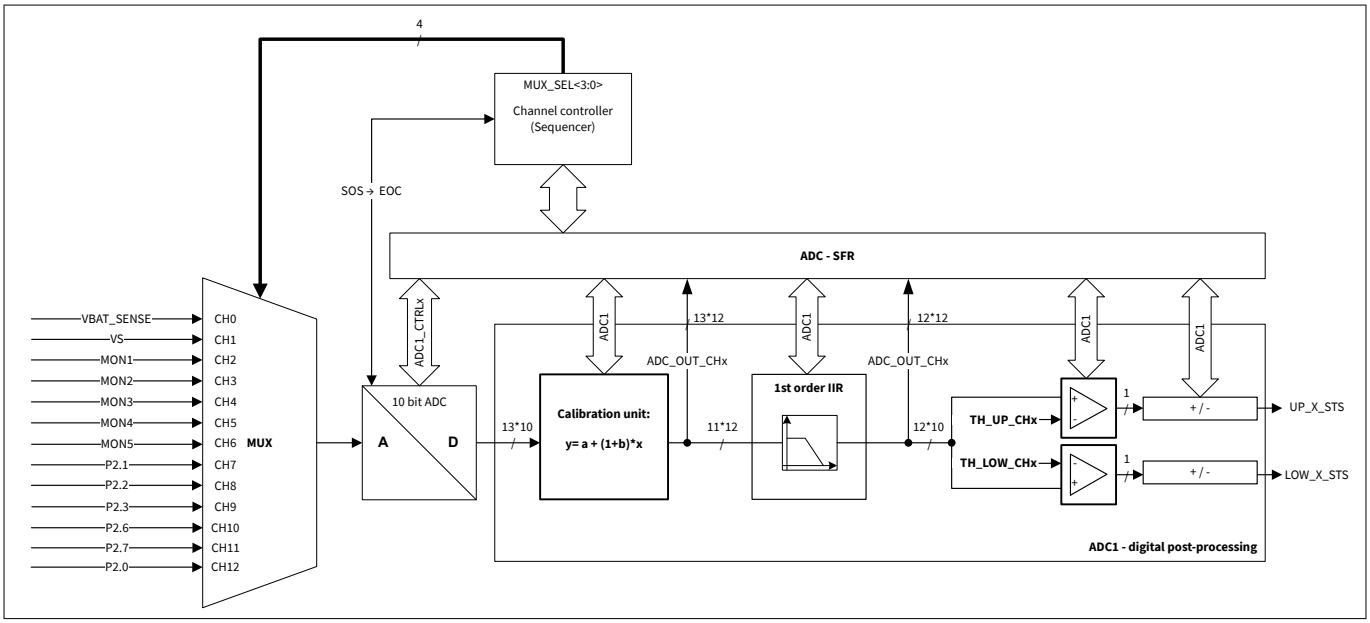


Figure 201 Module block diagram

As shown in the figure above, the ADC post-processing consists of a channel controller (sequencer), a 12 -channel demultiplexer and the signal processing block, which filters and compares the sampled ADC values for each channel individually. The channel control block controls the multiplexer sequencing on the analog side before the ADC and on the digital domain after the ADC. As described in the following section, the channel sequence can be controlled in a flexible way, which allows a certain degree of channel prioritization.

This capability can be used for example to set a higher priority to supply voltage channels compared to the other channel measurements. The measurement core module offers additionally two different post-processing measurement modes for over/undervoltage detection and for two-level threshold detection.

23.2.2 ADC1 modes overview

Usually the external register settings should only be changed during the start-up phase

“Exceptional Interrupt Measurement”, a high priority channel is inserted into the current sequence. The current actual measurement is not destroyed.

“Exceptional Sequence Measurement”, upon a hardware event, a complete sequence is inserted after the current measurement is finished. The current sequence is interrupted by the exception sequence.

“Software mode”, sequencer and exceptional interrupt and sequence measurement is disabled, each measurement is triggered by software.

The threshold counter can be bypassed ADC1_FILT_UP_CTRL and ADC1_FILT_LO_CTRL.

23.3 ADC1 - Core (10-bit ADC)

23.3.1 Functional description

The different sequencer modes are controlled by SFR register

- “Normal sequencer mode” is described in Chapter 23.5
- “Exceptional interrupt measurement” (EIM), upon a hardware event, the channel programmed in ADC1_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. Up to max. 63 consecutive measurements are possible
- “Exceptional sequence measurement“ (ESM), upon a hardware event, the sequence programmed in ADC1_CHx_ESM is inserted after the current measurement is finished. After the sequence (up to 12 measurements) exception is finished the next measurement from the interrupted sequence is selected. After the exceptional sequence measurement is finished an interrupt is issued
- “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

Software mode

- Software mode can be entered in different ways:
  - By writing one of the sequence registers SQn (for example to SQ_1[11:0]) to zero or setting the register ADC1_CTRL3.SW_MODE
  - By writing the exceptional sequence measurement (ADC1_CHx_ESM) to zero and enable the exceptional sequence measurement
  - Using debug suspend mode
- In software mode, the channel selection by the Sequencer is disabled. The entry of software mode is acknowledged in the ADC1_SQ_FB

  After the software mode is entered, the conversion are controlled through ADC1_CTRL_STS
- The software mode is left
  - When the maximum time is reached (maximum time specified in ADC1_MAX_TIME)
  - When the sequence which started the software mode is reprogrammed with at least one channel set in its registers SQn (for example to SQ_1) not equal to zero
  - When the exceptional sequence measurement (ADC1_CHx_ESM) is reprogrammed with at least one channel set
  - Leaving debug suspend mode

Important note

The ADC1 may give wrong results on channel 10 (P2.6) or channel 11 (P2.7), in case register SQ0_1.SQ0 == 0x000 (that is software mode is executed automatically)

Workaround option 1: Do not set register SQ0_1.SQ0 = 0x000. Reason: If at least one bit is set, the software mode is not started.

Workaround option 2: If SQ0_1.SQ0 = 0x000 needs to be used, make sure that MAX_TIME.MAX_TIME is >= 0x05.

Software mode

The default mode of the DPP1 is the sequencer mode. To change from this default mode to Software mode the corresponding flag ADC1.CTRL3.SW_MODE has to be set. In software mode, measurements are triggered by writing the ADC1_CTRL_STS.SOS bit. This bit is active as long as the conversion is in progress. The user polls the ADC1_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 ADC1_CTRL_STS.SOS.

Debug suspend mode

During debug suspend mode the sequencer is stopped once the current measurement is finished (after the next EOC event) and software mode is entered. 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/software mode. The maximum time of software mode is disabled in suspend mode. EIM and ESM events are ignored during debug suspend mode.

The ADC timing is controlled by SFR register

Sample time adjustment described in the register ADC1_CTRL3.

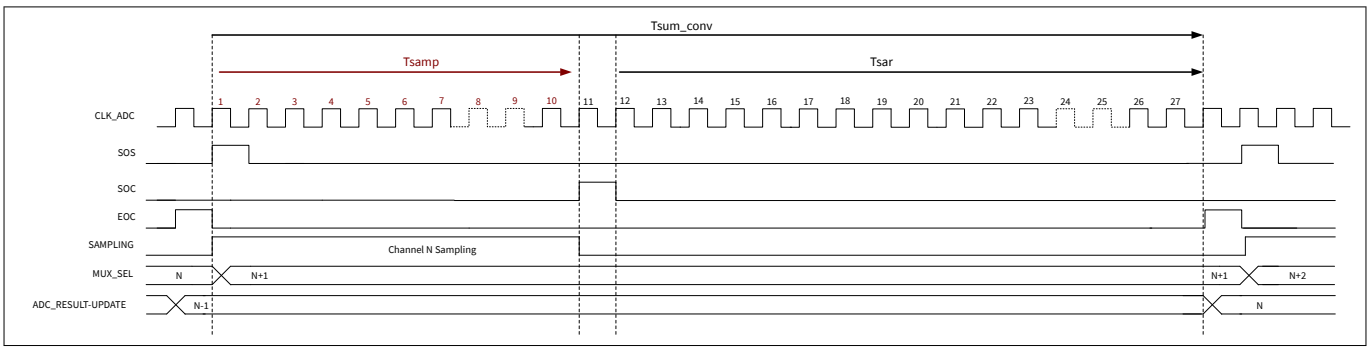


Figure 202 10-bit ADC timing - Single conversion

23.3.2 ADC1 control and status registers

The ADC1 is fully controllable by the below listed special function registers in software mode.
The registers are addressed wordwise.

23.3.2.1 Register overview - Control and status registers (ascending offset address)

Table 165 Register overview - Control and status registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_CTRL_STS	ADC1 control and status register	0000_H	777

23.4 ADC - Trigger unit

The DPP unit provides also a trigger block. This trigger block provides the following functionality:

- “Exceptional interrupt measurement” (EIM), upon hardware event, the channel programmed in ADC1_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 ADC1_CHx_ESM is inserted after the current measurement is finished. After the sequence (up to 12 measurements) exception is finished the next measurement from the interrupted sequence is selected. After the exceptional sequence measurement is finished an interrupt is issued
- “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.

23.5 Channel controller

23.5.1 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 166 Measurement channel sequence definition example (used as default sequence)

Measurement channel n	MSB CH11	CH10	CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	LSB CH0
Registers SQ_0_1[11:0]	1	1	1	1	1	1	1	1	1	1	1	1
Registers SQ_0_1[27:16]	0	0	0	0	0	0	1	1	1	1	0	0
Registers SQ_2_3[11:0]	1	0	0	0	0	0	0	0	0	0	0	0
Registers SQ_2_3[27:16]	1	1	1	1	1	1	1	1	1	1	1	1
Registers SQ_4_5[11:0]	0	0	0	0	0	0	0	0	0	0	0	0
Registers SQ_4_5[27:16]	0	0	0	0	0	0	0	0	0	0	0	0
Registers SQ_6_7[11:0]	1	1	1	1	1	1	1	1	1	1	1	1
Registers SQ_6_7[27:16]	0	0	0	0	0	0	0	0	0	0	0	0
Registers SQ_8_9[11:0]	0	0	0	0	0	0	0	0	0	0	0	0
Registers SQ_8_9[27:16]	1	1	1	1	1	1	1	1	1	1	1	1
Registers SQ_10_11[11:0]	0	0	0	0	0	0	0	0	0	0	0	0
Registers SQ_10_11[27:16]	0	0	0	0	0	0	0	0	0	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 12 and for each register from MSB to LSB, which defines a max. overall measurement periodicity of 144 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 not set, this measurement phase is skipped

In the upper example, the resulting channel sequence is defined as:

CH11, CH10, CH9, CH8, CH7, CH6, CH5, CH4, CH3,....., CH5, CH4, CH3, CH2, CH11, CH11

In MOTIX™ TLE984xQX Channels 0 - 11 can be fully programmed. The channels 0-11 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.

The overall measurement periodicity of all measurements in A/D conversion cycles is defined as:

$$\overline{N_{\text{meas}}} = \sum_{m=1}^{12} \left(\sum_{n=0}^{11} \text{SQ}_m[n] \right) \tag{28}$$

which results in 144 A/D conversion cycles. The average measurement periodicity of channel n in A/D conversion cycles is defined as

$$\overline{N_{\text{meas},\,n}} = \frac{\left(\sum_{m=1}^{12} \text{SQ}_m[n]\right)}{\overline{T_{\text{meas}}}} \tag{29}$$

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 consumes, as default setting, 12 clock cycles for the sampling of the input voltage. The conversion time for a single channel measurement value is 17 clock cycles.

The minimum measurement periodicity, which can be achieved, by enabling only channel 1 in the sequence registers, depends on the ADC1_CLK frequency and is given by:

$$\overline{T_{\text{meas\_CHI\_min}}} = \frac{26}{f_{\text{adc1\_clk}}} \tag{30}$$

This following calculations include already the sampling time of ADC. If all programmable channels are enabled, the maximum periodicity is calculated:

$$\overline{T_{\text{meas\_CHI\_min}}} = \frac{312}{f_{\text{adc1\_clk}}} \tag{31}$$

For a ADC1_CLK frequency of 24 MHz, the channel 1 is measured with min. 1.1 µs. The maximum update time of channel 1 with 24 MHz clock frequency is 10 µs. As mentioned before, this is calculated with the assumption, that all channels are enabled and channel1 is enabled in every sequence register. As a prerequisite for this calculation we take ADC1_CTRL3 = 4 (sample period = 12 ADC1_CLK clock cycles).

23.5.2 Channel controller control registers

The channel controller can fully be configured by the SFR registers listed below.
The registers are addressed wordwise.

23.5.2.1 Register overview - Channel controller control registers (ascending offset address)

Table 167 Register overview - Channel controller control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_SQ_FB	Sequencer feedback register	0004_H	779
ADC1_CHx_EIM	Channel setting bits for exceptional interrupt measurement register	0008_H	781
ADC1_CHx_ESM	Channel setting bits for exceptional sequence measurement register	$000C_H$	783
ADC1_MAX_TIME	Maximum time for software mode register	0010_H	785
ADC1_CTRL2	Measurement unit 1 control 2 register	0014_H	786
ADC1_CTRL3	Measurement unit 1 control 3 register	0018_H	787
ADC1_CTRL5	Measurement unit 1 control 5 register	$001C_H$	789
ADC1_SQ0_1	Measurement unit 1 channel enable bits for cycle 0-1 register	0020_H	790
ADC1_SQ2_3	Measurement unit 1 channel enable bits for cycle 2-3 register	0024_H	791
ADC1_SQ4_5	Measurement unit 1 channel enable bits for cycle 4-5 register	0028_H	792
ADC1_SQ6_7	Measurement unit 1 channel enable bits for cycle 6-7 register	$002C_H$	793
ADC1_SQ8_9	Measurement unit 1 channel enable bits for cycle 8-9 register	0030_H	794
ADC1_SQ10_11	Measurement unit 1 channel enable bits for cycle 10-11 register	0034_H	795
ADC1_CTRL4	Measurement unit 1 control 4 register	0038_H	796

23.6 Calibration unit

23.6.1 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 chapter describe usage and setup of the calibration unit.

23.6.1.1 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. This non-idealities are caused by the corresponding measurement chain modules.

Those first order non-idealities are:

- Offset and gain error of ADC1
- Offset and gain error of the attenuator (especially voltage measurement)
- Offset and gain error 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 VBAT_SENSE-Pin and the HV-monitoring-pins was done without external resistor.

23.6.1.2 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 100 TP 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.

The parameters ADC1_CALOFFS_CHx and ADC1_CALGAIN_CHx are stored in a 8-bit, 2th complement format. The function applied to calculate the calibrated ADC value is

ADC_cal_CHx = (1 + <ADC_CALGAIN_CHx>/1024) × ADC_uncal_CHx + <ADC_CALOFFS_CHx>/2

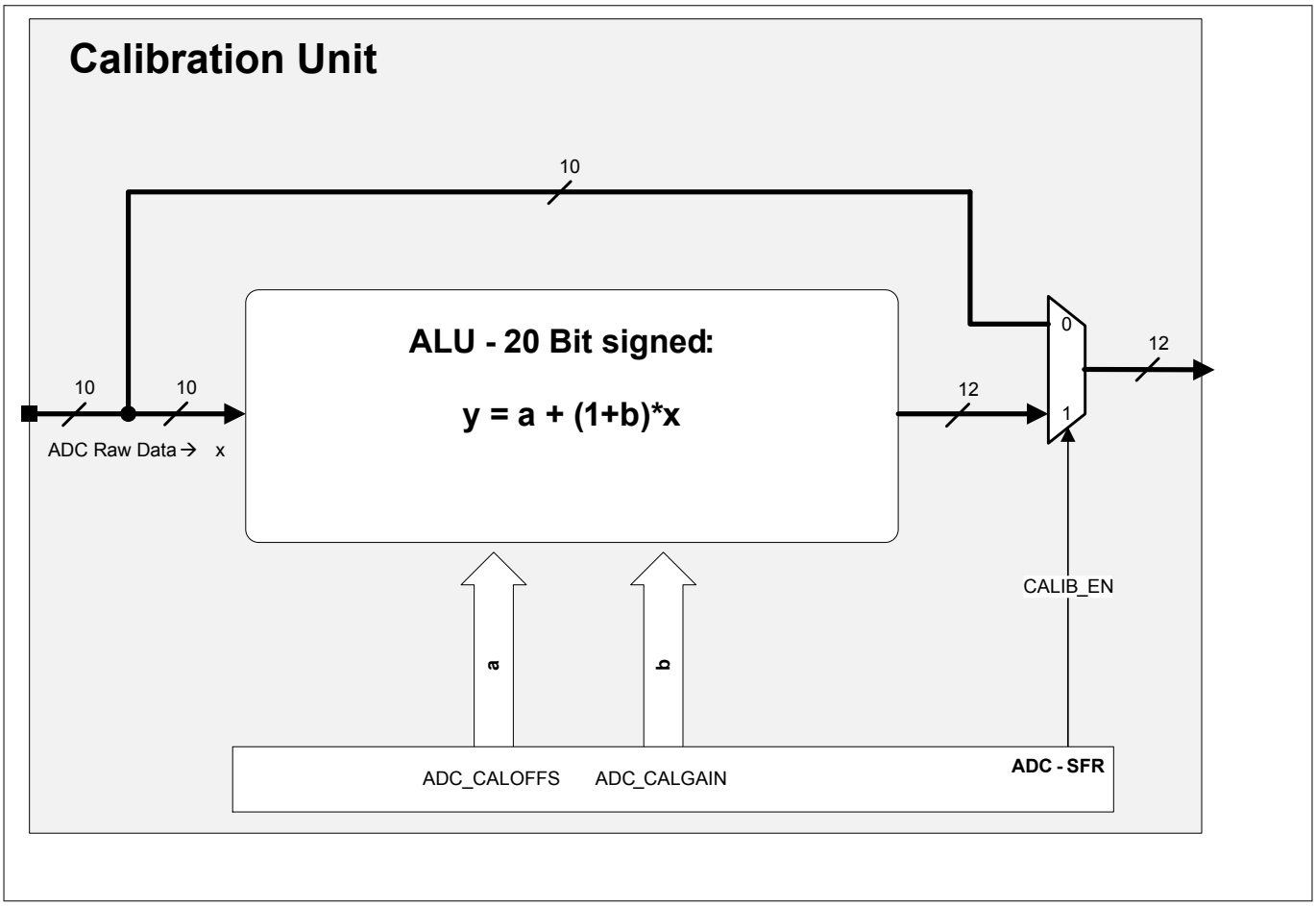


Figure 203 Structure of calibration unit

23.6.2 Calibration unit control registers

The calibration unit can be configured by the SFR registers listed below.

All calibration registers can be written by the user. This allows an in-system recalibration of a dedicated measurement.

The registers are addressed wordwise.

23.6.2.1 Register overview - Calibration unit control registers (ascending offset address)

Table 168 Register overview - Calibration unit control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_CAL_CH0_1	Calibration for channel 0 and 1 register	0048_H	797
ADC1_CAL_CH2_3	Calibration for channel 2 and 3 register	$004C_H$	798
ADC1_CAL_CH4_5	Calibration for channel 4 and 5 register	0050_H	799
ADC1_CAL_CH6_7	Calibration for channel 6 and 7 register	0054_H	800
ADC1_CAL_CH8_9	Calibration for channel 8 and 9 register	0058_H	801
ADC1_CAL_CH10_11	Calibration for channel 10 and 11 register	$005C_H$	802

23.7 IIR-filter

23.7.1 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.

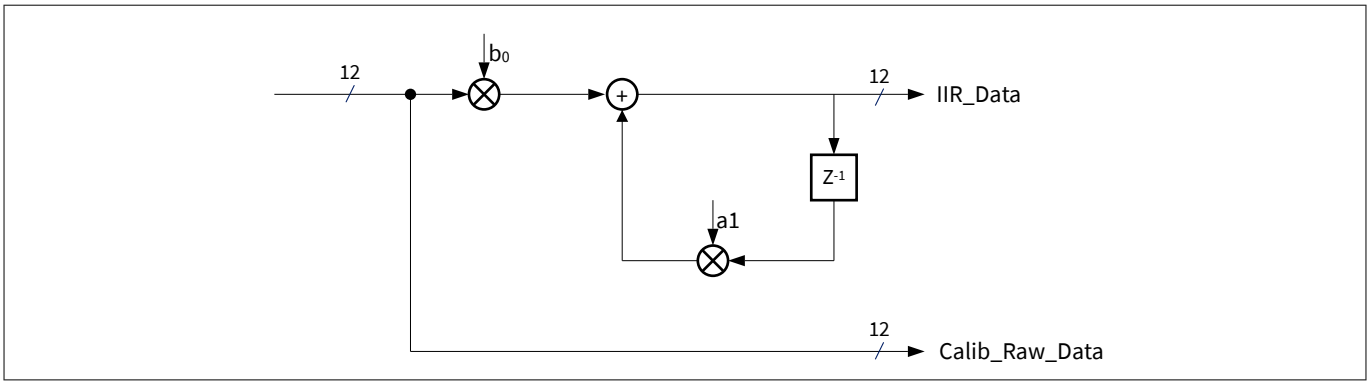


Figure 204 IIR-filter implementation structure

$$H_{IIR}\left(z\right) = \frac{b}{1 - a \times z^{-1}} \tag{32}$$

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:

$$b = 1 - a \tag{33}$$

The IIR Filter transfer function is shown in the plot below.

$$H_{IIR}\left(z\right) = \frac{1 - a}{\left(1 - a \times z^{-1}\right)} \tag{34}$$

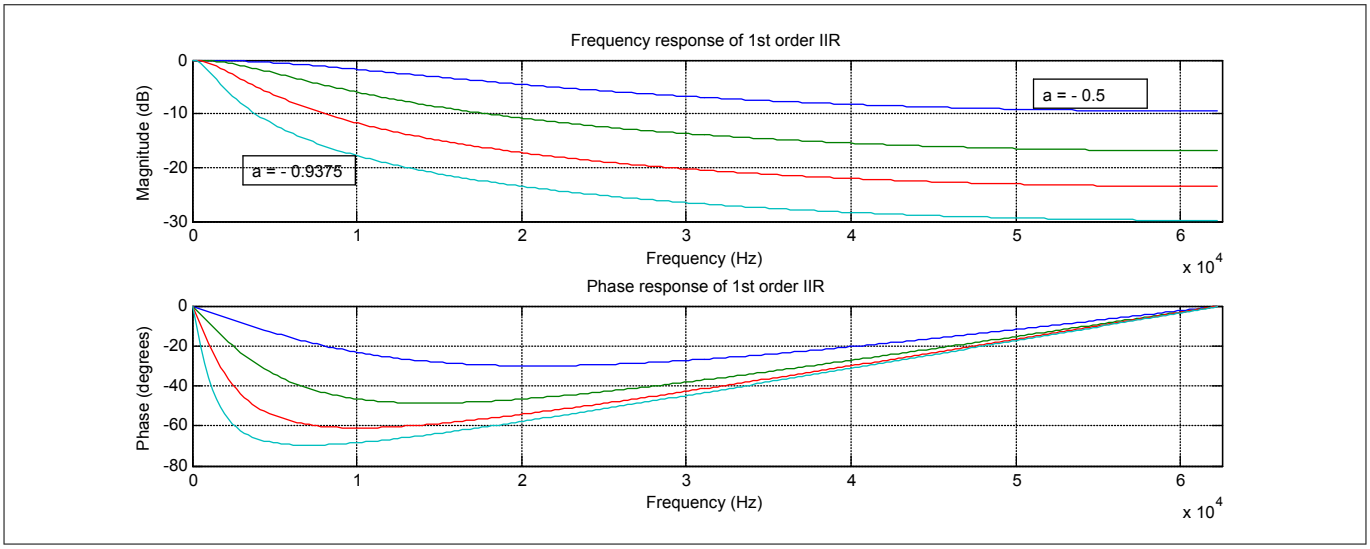


Figure 205 IIR filter transfer function for different filter length fl (1 MHz corresponds to 1/2*channel sampling frequency)

23.7.1.1 Step response

The step response of the IIR filter time is shown in the following figure:

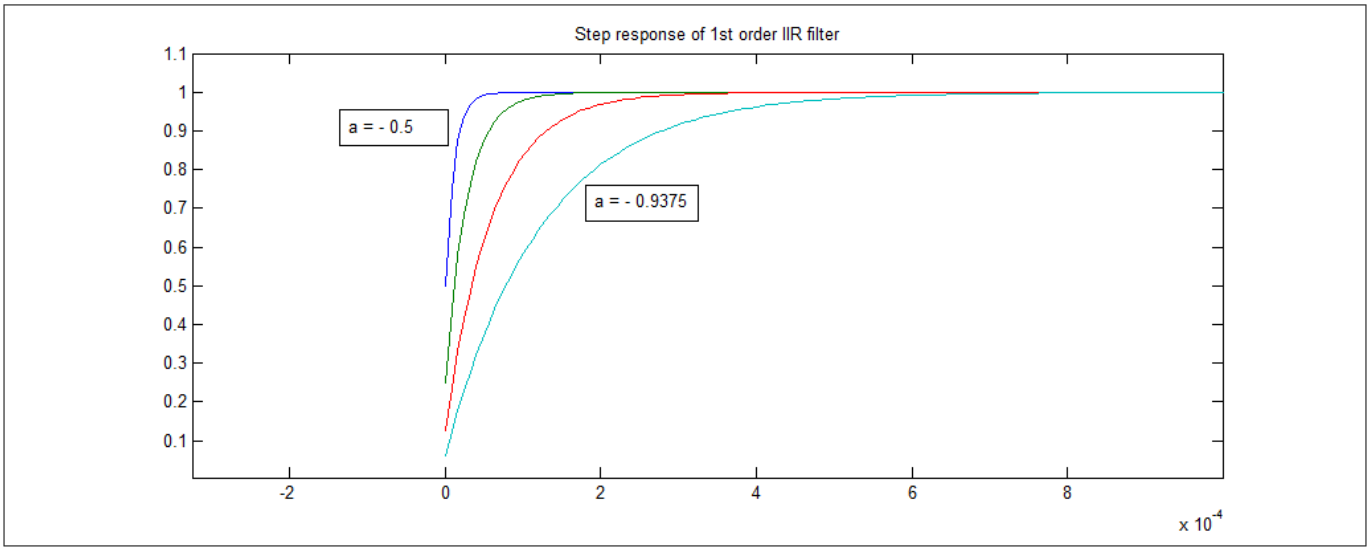


Figure 206 IIR step response time

The following table summarizes the main filter characteristics.

Table 169 IIR filter characteristics

Filter coefficient	Group delay at = ω0	Normalized -3 dB frequency [1)]	-3 dB frequency @ $f_{s\_ch}/2$ = 250 KHz
a	τ[samples]	$f_{-3dB}/(f_{s\_ch}/2)$	f_{-3dB} [Hz]
2^{-1}	2		
2^{-2}	4		
2^{-3}	8		

(table continues...)

Table 169 (continued) IIR filter characteristics

Filter coefficient	Group delay at = ω0	Normalized -3 dB frequency [1)]	-3 dB frequency @ $f_{s\_ch}/2$ = 250 KHz
a	τ[samples]	$f_{-3dB}/(f_{s\_ch}/2)$	f_{-3dB} [Hz]
2^{-4}	16		

1) The - 3dB frequency of the filter is normalized to half the channel sampling frequency (Nyquist frequency).

23.7.2 IIR filter control registers

The IIR filter can be configured by the SFR registers listed below.

The ADC1_FILT_OUT0 to ADC1_FILT_OUT11 registers are 12 bits wide, but the ADC delivers only a resolution of 10 bits. Table 170 shows how the lower two bits are determined.

Table 170 ADC1_FILT_OUT register setting

ADC1_CTRL2.calib_en	ADC1_CTRL5.filt_out_sel	ADC1_FILT_OUT0.output[1:0]
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 12 bits (see Chapter 23.6.1.2), the output is feed into the IIR filter. The internal result of the IIR filter is 12 bits (see Chapter 23.7.1), the output is converted to 10 bit and feed into the post processing. The user can monitor the calculated values in the ADC1_FILT_OUT0 to ADC1_FILT_OUT11 registers and gets access to 10 bit wide result information.

The registers are addressed wordwise.

23.7.2.1 Register overview - IIR filter control registers (ascending offset address)

Table 171 Register overview - IIR filter control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_FILTCOEFF0_11	Filter coefficients measurement unit channel 0-11 register	0060_H	803
ADC1_FILT_OUT0	ADC1 or filter output channel 0 register	0070_H	805
ADC1_FILT_OUT1	ADC1 or filter output channel 1 register	0074_H	806
ADC1_FILT_OUT2	ADC1 or filter output channel 2 register	0078_H	807
ADC1_FILT_OUT3	ADC1 or filter output channel 3 register	$007C_H$	808
ADC1_FILT_OUT4	ADC1 or filter output channel 4 register	0080_H	809
ADC1_FILT_OUT5	ADC1 or filter output channel 5 register	0084_H	810
ADC1_FILT_OUT6	ADC1 or filter output channel 6 register	0088_H	811
ADC1_FILT_OUT7	ADC1 or filter output channel 7 register	$008C_H$	812
ADC1_FILT_OUT8	ADC1 or filter output channel 8 register	0090_H	813
ADC1_FILT_OUT9	ADC1 or filter output channel 9 register	0094_H	814
ADC1_FILT_OUT10	ADC1 or filter output channel 10 register	0098_H	815
ADC1_FILT_OUT11	ADC1 or filter output channel 11 register	$009C_H$	816
ADC1_DIFFCH_OUT1	ADC1 differential channel output 1 register	$00A0_H$	817
ADC1_DIFFCH_OUT2	ADC1 differential channel output 2 register	$00A4_H$	818
ADC1_DIFFCH_OUT3	ADC1 differential channel output 3 register	$00A8_H$	819
ADC1_DIFFCH_OUT4	ADC1 differential channel output 4 register	$00AC_H$	820

(table continues...)

Table 171 (continued) Register overview - IIR filter control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_FILT_OUT12	ADC1 or filter output channel 12 register	0110_H	821
ADC1_FILT_OUTEIM	ADC1 or filter output of EIM register	0120_H	822

23.8 Signal processing

23.8.1 Functional description

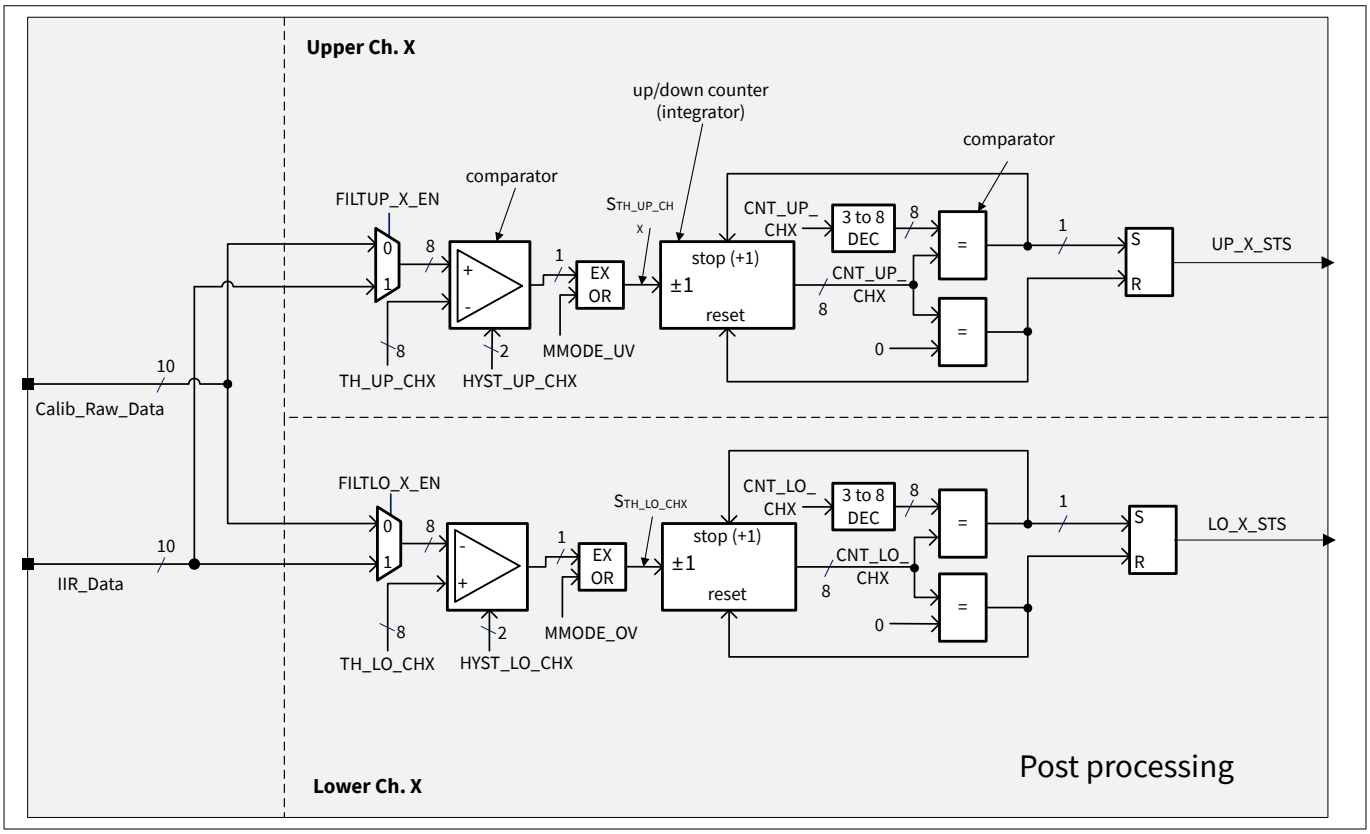


Figure 207 Post processing channel block diagram for voltage measurements

As shown in the previous figure 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 de-multiplexed ADC output signal ADC_OUTX is compared with an upper threshold TH_UP_CHX and a lower threshold TH_LO_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 HYST_UP_CHX and 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 TH_LO_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 $2^{CNT\_LO/HI\_CHX}$.

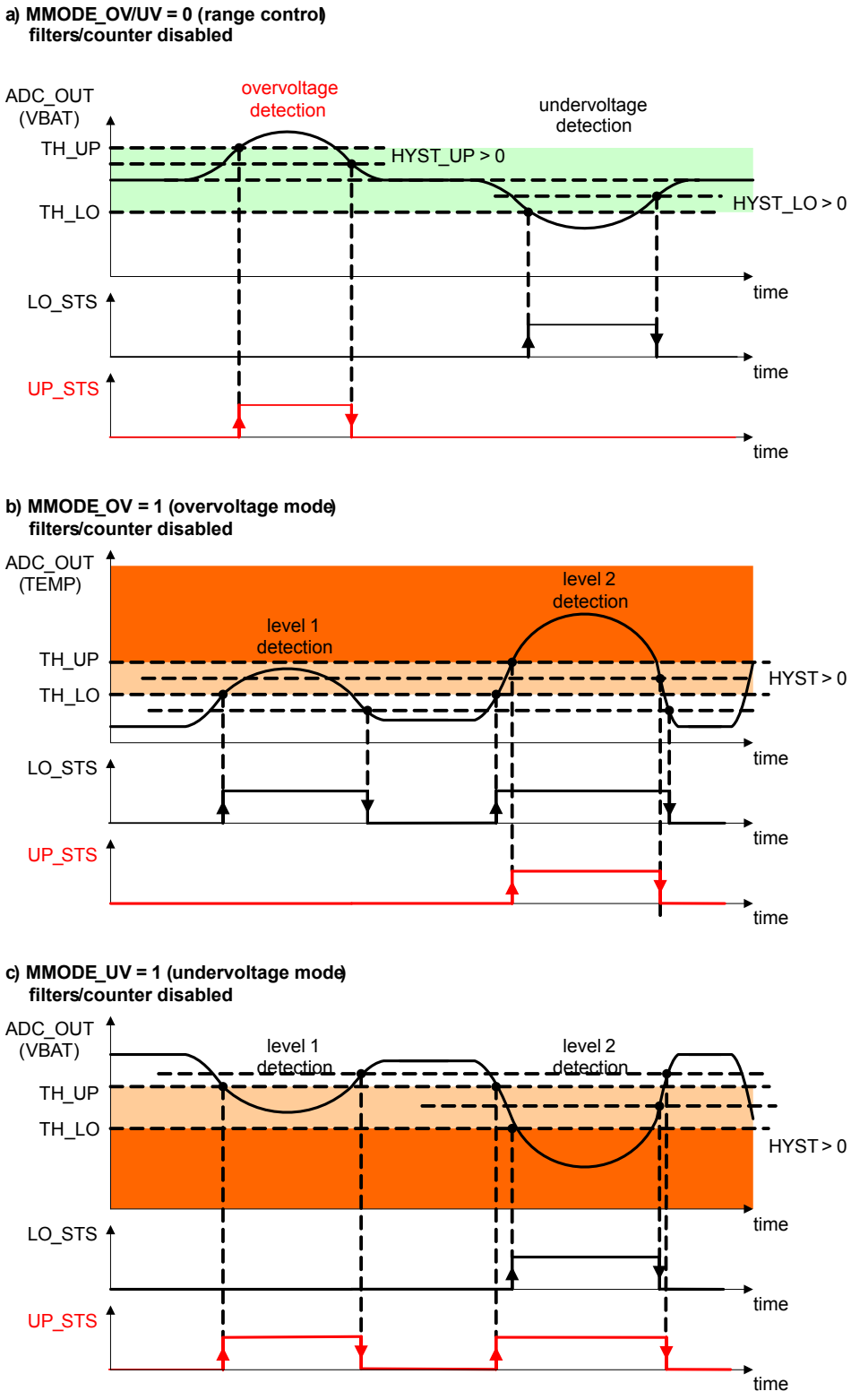


Figure 208 Measurement examples of a measurement channel with disabled filters

The previous figure shows three examples, an over- and undervoltage detection (for example VBAT_SENSE monitoring), a 2-step overvoltage and a 2-step undervoltage detection. The modes MMODE_OV/UV = 1 can be used as pre-warning for the application software (for example close to supply undervoltage).

23.8.2 Postprocessing control registers

The postprocessing block is fully controllable by the below listed SFR registers.

The registers are addressed wordwise.

23.8.2.1 Register overview - Postprocessing control registers (ascending offset address)

Table 172 Register overview - Postprocessing control registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_TH0_3_LOWER	Lower comparator trigger level channel 0-3	0040_H	823
ADC1_TH4_7_LOWER	Lower comparator trigger level channel 4-7	0044_H	824
ADC1_FILT_UP_CTRL	Upper threshold filter enable	$00B0_H$	825
ADC1_FILT_LO_CTRL	Lower Threshold filter enable	$00B4_H$	827
ADC1_TH8_11_LOWER	Lower comparator trigger level channel 8-11	$00C0_H$	829
ADC1_DCHTH1_4_LOWER	Lower comparator trigger level differential channel 1-4 register	$00C4_H$	830
ADC1_TH0_3_UPPER	Upper comparator trigger level channel 0-3 register	$00C8_H$	831
ADC1_TH4_7_UPPER	Upper comparator trigger level channel 4-7 register	$00CC_H$	832
ADC1_TH8_11_UPPER	Upper comparator trigger level channel 8-11 register	$00D0_H$	833
ADC1_DCHTH1_4_UPPER	Upper comparator trigger level differential channel 1-4 register	$00D4_H$	834
ADC1_CNT0_3_LOWER	Lower counter trigger level channel 0-3 register	$00D8_H$	835
ADC1_CNT4_7_LOWER	Lower counter trigger level channel 4-7 register	$00DC_H$	837
ADC1_CNT8_11_LOWER	Lower counter trigger level channel 8-11 register	$00E0_H$	839
ADC1_DCHCNT1_4_LOWER	Lower counter trigger level differential channel 1-4 register	$00E4_H$	841
ADC1_CNT0_3_UPPER	Upper counter trigger level channel 0-3 register	$00E8_H$	843
ADC1_CNT4_7_UPPER	Upper counter trigger level channel 4-7 register	$00EC_H$	845
ADC1_CNT8_11_UPPER	Upper counter trigger level channel 8-11 register	$00F0_H$	847
ADC1_DCHCNT1_4_UPPER	Upper counter trigger level differential channel 1-4 register	$00F4_H$	849
ADC1_MMODE0_11	Overvoltage measurement mode of channel 0-11 register	$00F8_H$	851

23.9 Interrupt handling

23.9.1 Functional description

The following figure shows the interrupt generation of ADC1. The generated interrupts are assigned to several nodes. The exact mapping can be red in the corresponding interrupt chapter of this device.

Note: All status flags and interrupt status flags are blanked within the start-up procedure of the sequencer. The purpose of this is to avoid wrong setting of those flags due to settling behavior of the integrated filter structures.

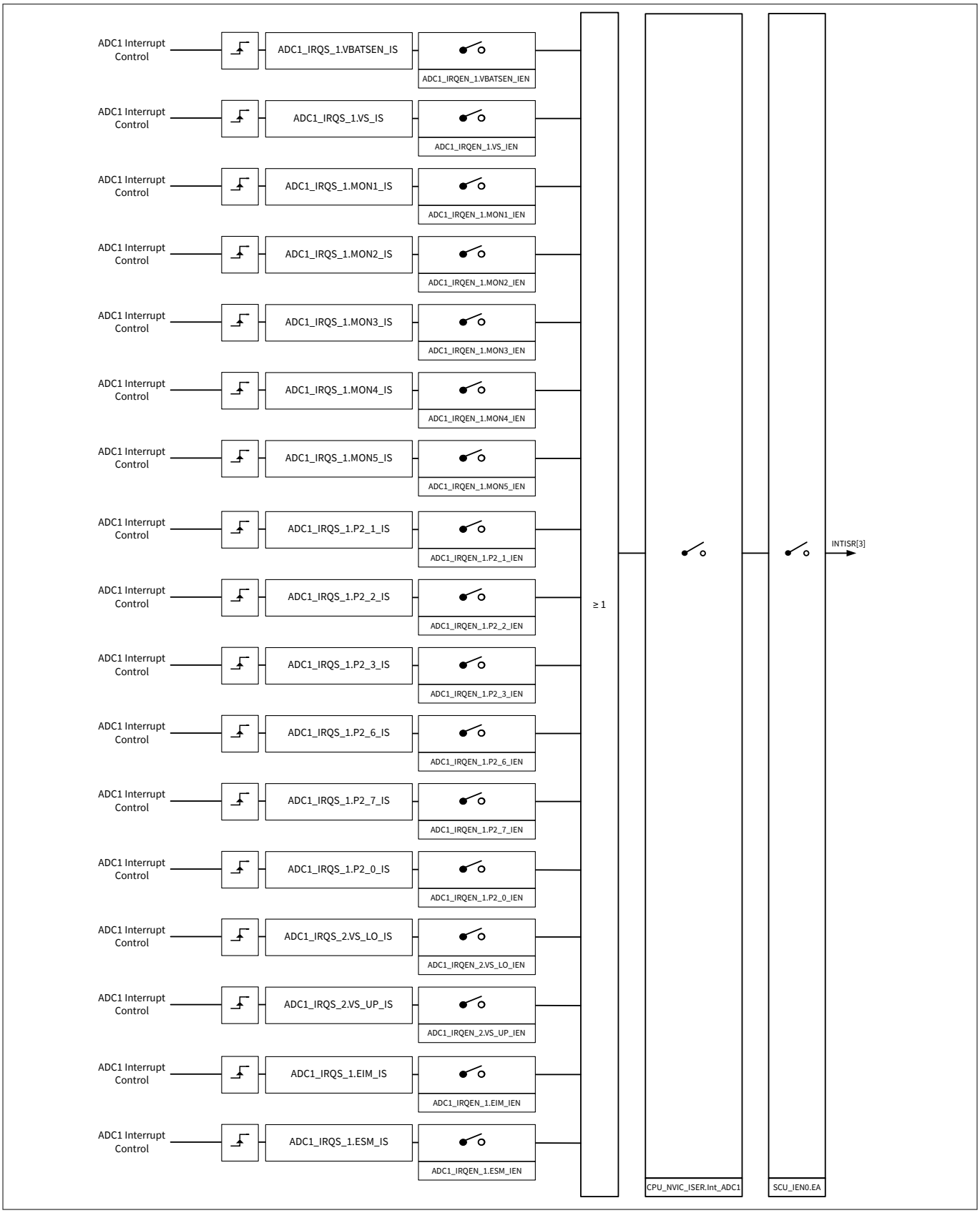


Figure 209 ADC1 interrupt generation of all existing channels

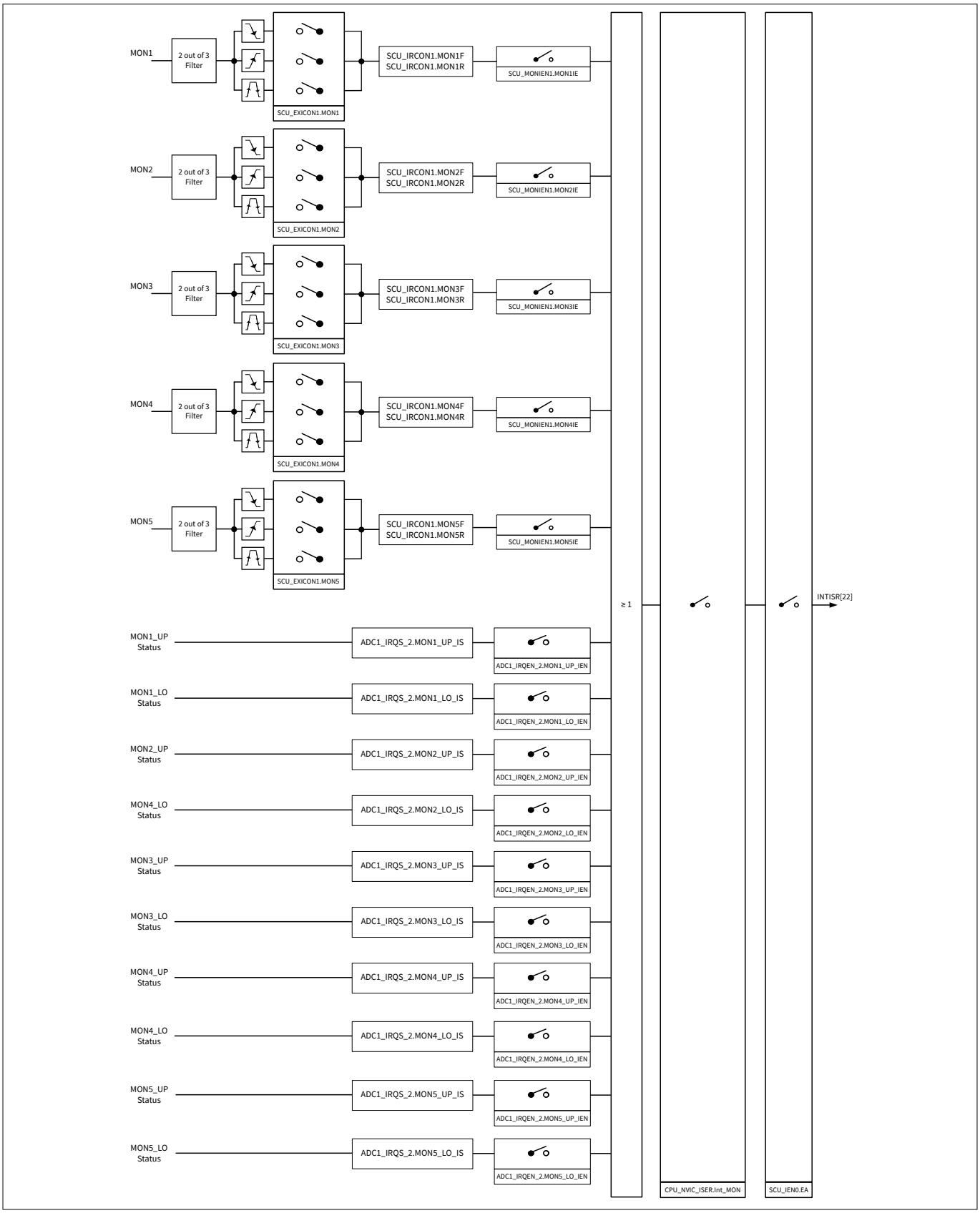


Figure 210 ADC1 interrupt generation for monitoring input

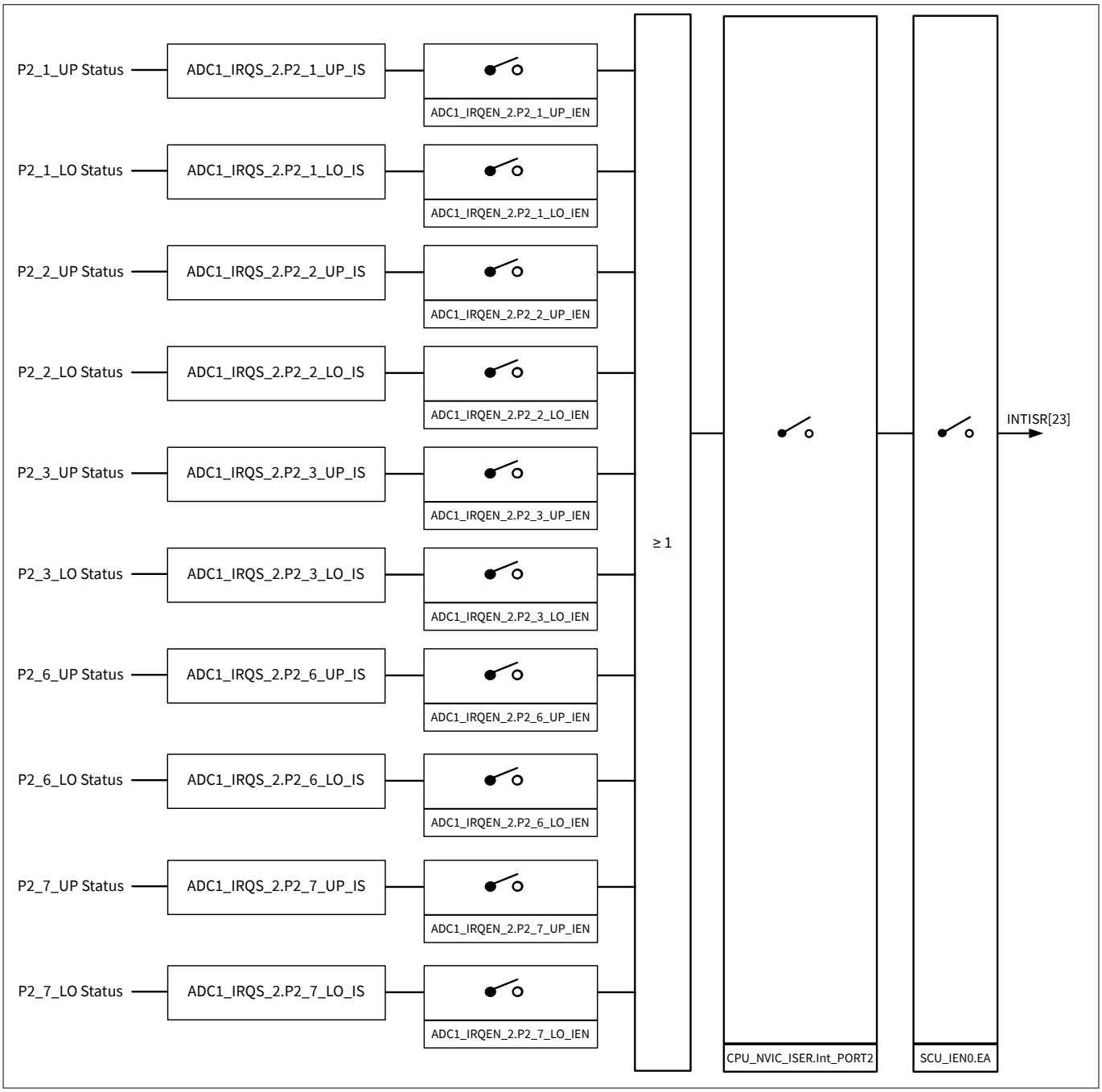


Figure 211 ADC1 interrupt generation for port 2 input

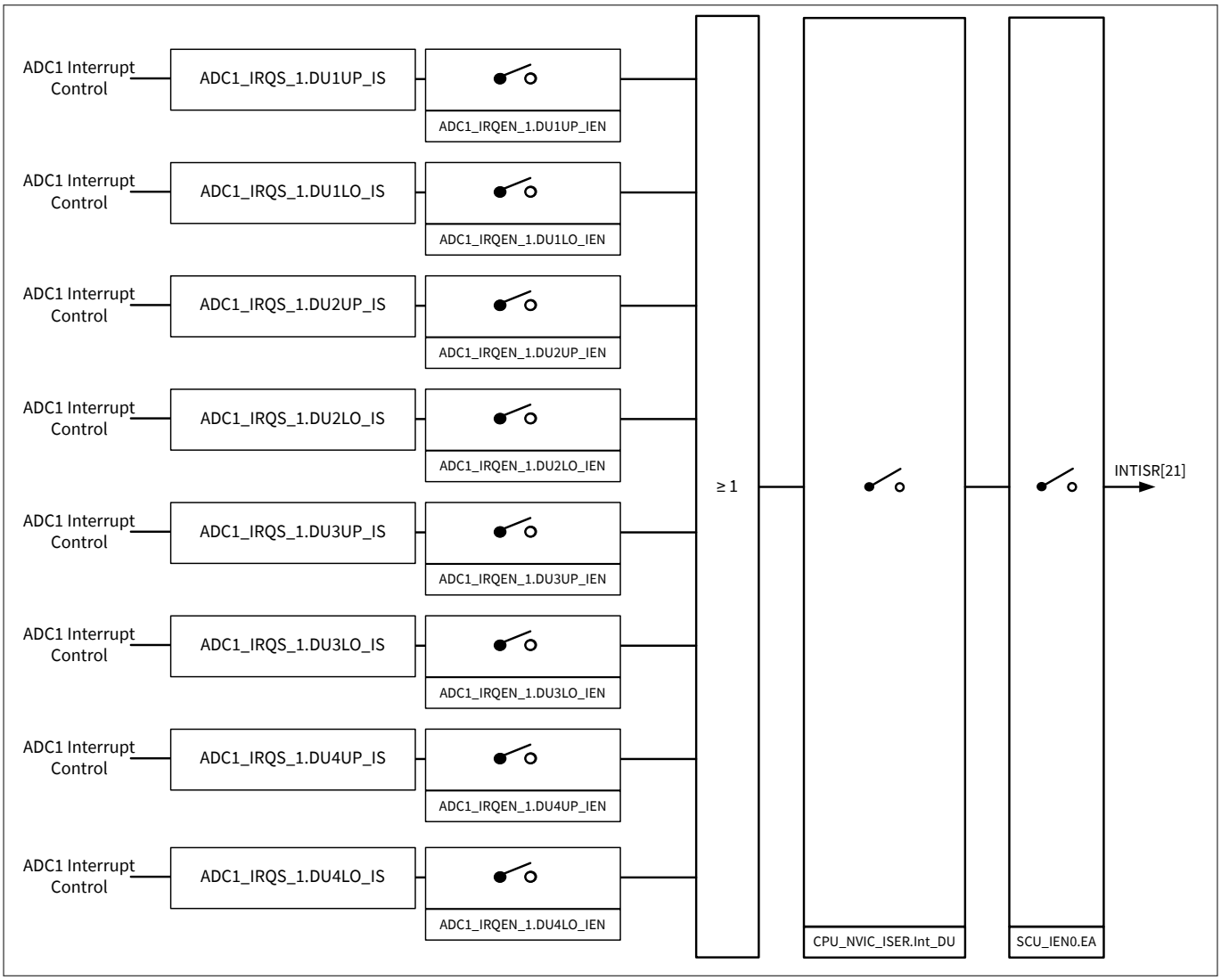


Figure 212 ADC1 interrupt generation for differential unit

23.9.2 Interrupt registers

The registers are addressed wordwise.

23.9.2.1 Register overview - Interrupt registers (ascending offset address)

Table 173 Register overview - Interrupt registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_IRQS_1	ADC1 interrupt status 1 register	0064_H	854
ADC1_IRQEN_1	ADC1 interrupt enable 1 register	0068_H	857
ADC1_IRQCLR_1	ADC1 interrupt status clear 1 register	$006C_H$	860
ADC1_IRQS_2	ADC1 interrupt status 2 register	0100_H	863
ADC1_STS_2	ADC1 status 2 register	0104_H	866
ADC1_IRQCLR_2	ADC1 interrupt status clear 2 register	0108_H	869
ADC1_IRQEN_2	ADC1 interrupt enable 2 register	$010C_H$	872
ADC1_STS_1	ADC1 status 1 register	0124_H	875
ADC1_STSCLR_1	ADC1 status clear 1 register	0128_H	877

23.10 Module interfaces

The following figure shows the ADC1 trigger mechanism:

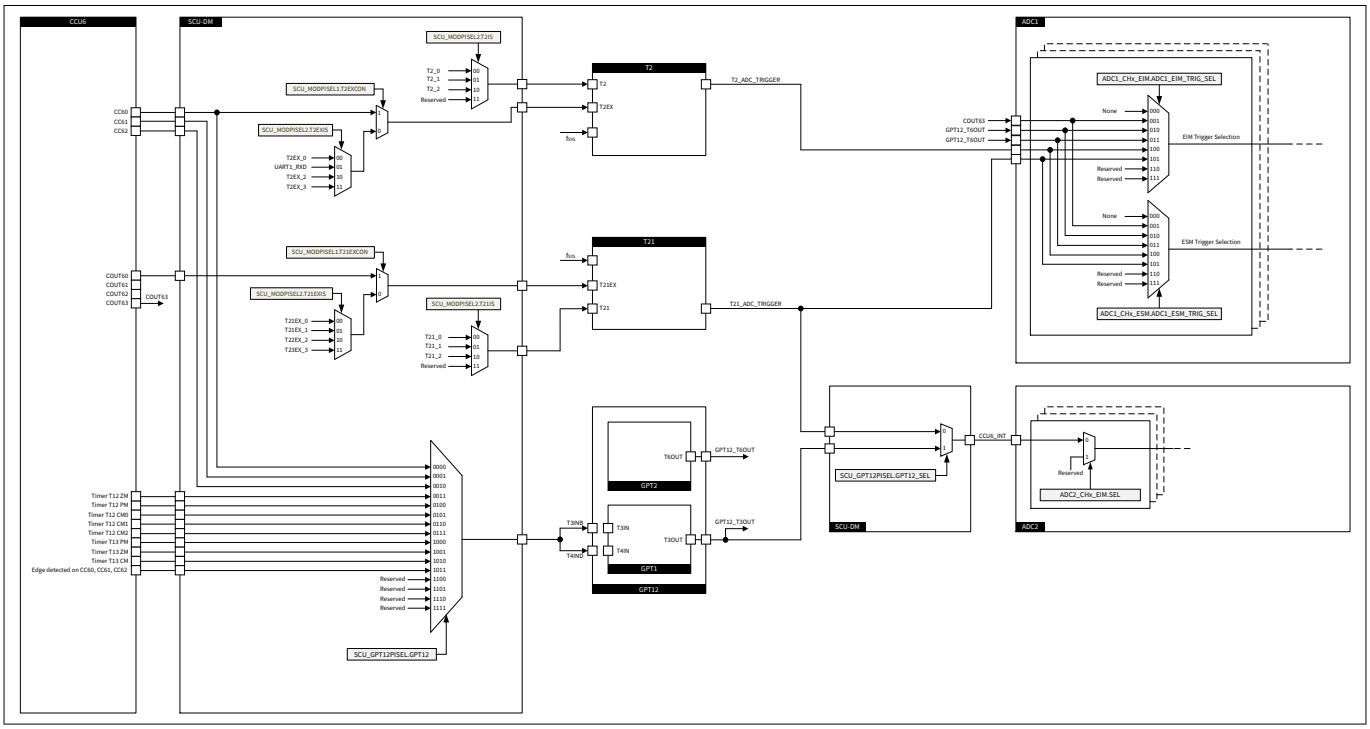


Figure 213 ADC1 trigger mechanism

23.11 Differential measurement unit (only TLE9845QX)

23.11.1 Motivation for differential measurement unit

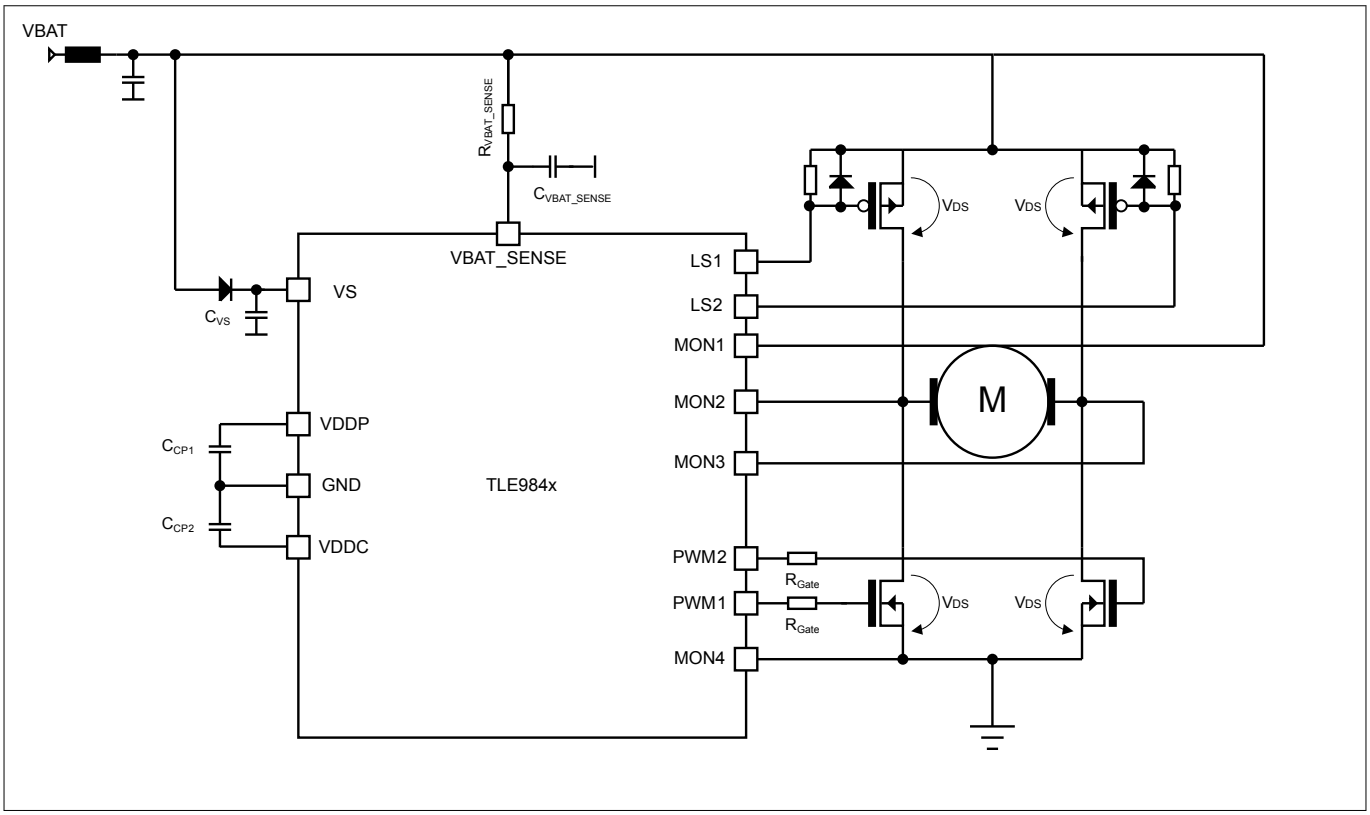


Figure 214 Motivation for differential measurement unit

23.11.2 Implementation of differential measurement unit

The differential measurement unit is a sub-unit of the digital post processing. It calculates the difference between selectable monitoring channels. The structure is shown in the following figure.

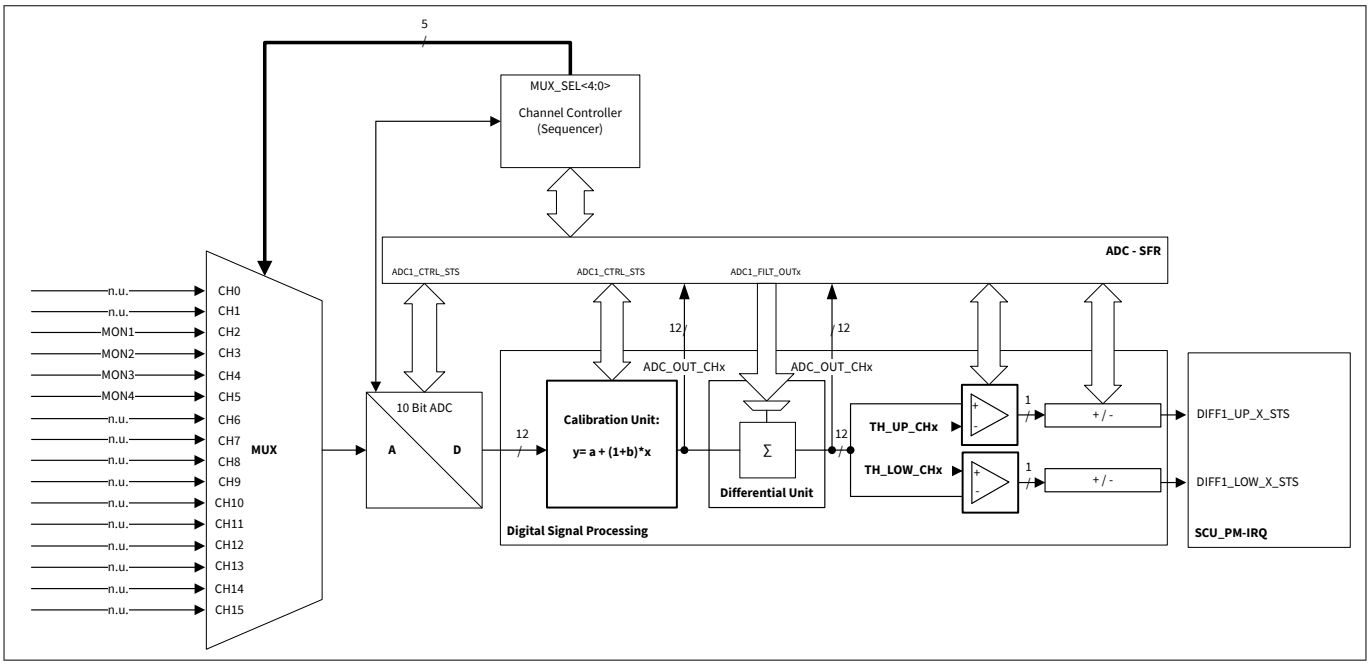


Figure 215 Structure of differential measurement unit

The data processing unit also offers a differential evaluation of the monitoring channels. This offers the possibility to build up a V_{DS} monitoring for H-bridge control. The V_{DS} monitoring is realized by the sequencer. The user enables the 4 required MONs in the sequencer and the sequencer triggers the ADC to perform the measurements. In a failure case, CTRAP_3 is configurable to switch CCU6-channels to passive state without CPU load or interrupt handling of differential unit (TLE9845QX only).

Due to the fact that this measurements need to be aligned to a certain PWM control scenario there is necessary to blank the measurements which are falling in the switching phase of a PWM channel. For this purpose there is a special enable procedure for the DU unit whose timing is sketched in the following figure.

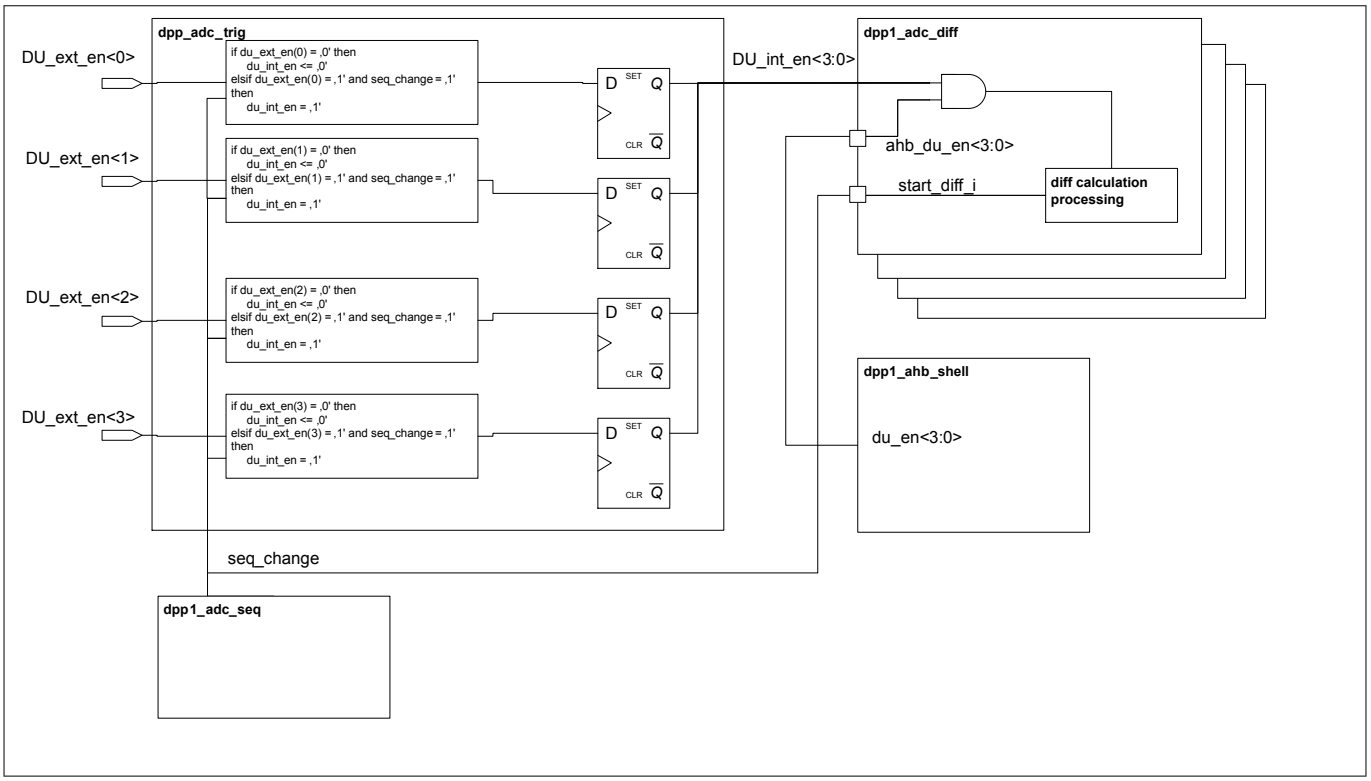


Figure 216 Generation of enable signals for the DU unit

The enable inputs are generated in the SCU_DM module. They logic sketched in the previous figure is blanking the DU unit inputs from incoming ADC results of used MON channels for the emulated V_{DS} Monitoring. The timing of a typical ccu6 patttern can be seen in the following figure. The green ‘pulses’ show the valid sequences which are calculated by the DU unit.

The DU_int_en<x> signals are configured in register SCU_MODPISEL4. All DU_int_en<x> signals are AND-gated with COUT63.

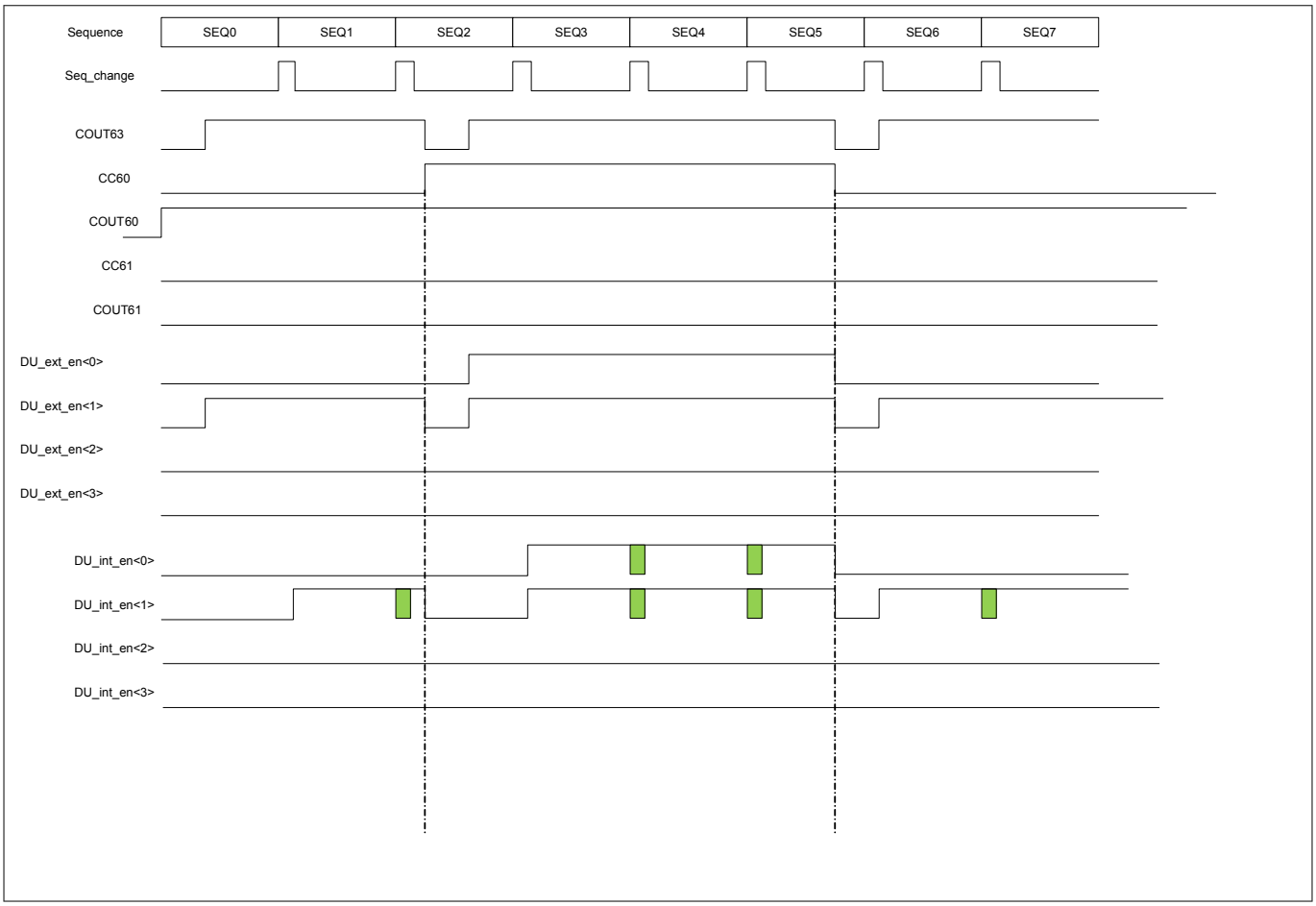


Figure 217 Timing of enable signals for the DU unit

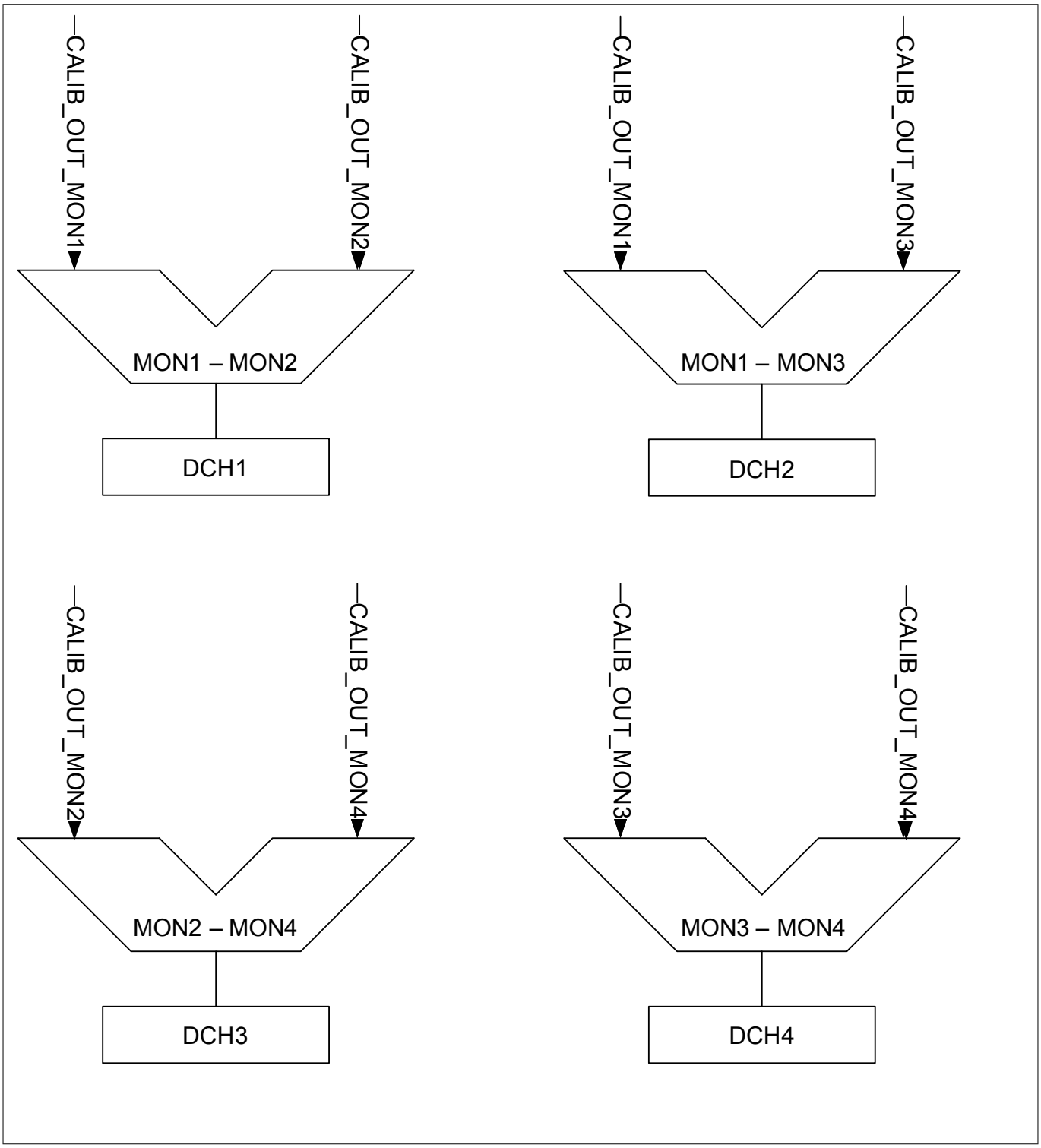


Figure 218 Structure of differential measurement unit

23.11.3 ADC1 differential unit input selection register

The registers are addressed wordwise.

23.11.3.1 Register overview - Differential unit input selection registers (ascending offset address)

Table 174 Register overview - Differential unit input selection registers (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_DUIN_SEL	Measurement unit 1 - Differential unit input selection register	$00FC_H$	879

23.12 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 MI_RDY, in bit ADC1_CTRL3.MCM_RDY.

Measurement core start-up procedure: the start-up time of the complete signal chain are 2200 EoC cycles. The IIR-filter coefficient is set to C = 2^- 1 (fastest response time).

During the start-up phase, the DPP will use SQ = 1111_1111_1111, regardless of the sequence registers configuration.

During the start-up phase, the output registers ADC1_FILT_OUTx are normally updated with the converted values. It is recommended to clear all registers before the will be used for application purposes.

23.13 Post processing default values

The following table shows the assigned measurements of the particular channels and the reset default values which are configured by FW during power-up. Since all channels are configurable by the user, the reset values can be reconfigured by writing the corresponding registers.

Table 175 Channel allocation and post processing default settings (effective after reset)

Channel#	Name	Function	MMODE[1].	FILTCOEFF[2].		Threshold digital[3]	Threshold analog	Hysteresis[4]	Counters[5]
Ch. 0	VBAT_SENSE	VBAT_SENSE	0	3	Upper	$C0_H$	19.27 V	3	2
					Lower	$3A_H$	5.79 V	2	2
Ch. 1	VS	VS supply voltage	0	3	Upper	$C5_H$	19.78 V	3	3
					Lower	42_H	6.59 V	2	3
Ch. 2	MON1	MON1	0	3	Upper	FF_H	30.87 V	0	3
					Lower	00_H	0 V	0	3
Ch. 3	MON2	MON2	0	3	Upper	FF_H	30.87 V	0	2
					Lower	00_H	0 V	0	2
Ch. 4	MON3	MON3	0	3	Upper	FF_H	30.87 V	0	0
					Lower	00_H	0 V	0	0
Ch. 5	MON4	MON4	0	3	Upper	FF_H	30.87 V	0	0
					Lower	00_H	0 V	0	0
Ch. 6	MON5/ P2.0	MON5 or P2.0 (device variant dependent)	0	3	Upper	FF_H	30.87 V/ 5.50 V	0	0
					Lower	00_H	0 V	0	0
Ch. 7	P2.1	P2.1	0	3	Upper	FF_H	5.50 V	0	0
					Lower	00_H	0 V	0	0
Ch. 8	P2.2	P2.2	0	3	Upper	FF_H	5.50 V	0	0
					Lower	00_H	0 V	0	0
Ch. 9	P2.3	P2.3	0	3	Upper	FF_H	5.50 V	0	0
					Lower	00_H	0 V	0	0
Ch10	P2.6	P2.6	0	3	Upper	FF_H	5.50 V	0	0
					Lower	00_H	0 V	0	0
Ch11	P2.7	P2.7	0	3	Upper	FF_H	5.50 V	0	0
					Lower	00_H	0 V	0	0

1) Register MMODE0_11; 0 = range control, 1 = UV, 2 = OV
2) Register FILTCOEFF0_11; 0 = 1/2, 1 = 1/4, 2 = 1/8, 3 = 1/16
3) Bit-field CHn_UP/CHn_LOW.
4) Bit-field HYST_UP_CHn/HYST_LO_CHn; 0 = hyst off, 1 = hyst 4, 2 = hyst 8, 3 = hyst 16.
5) Bit-field CNT_UP_CHn/CNT_LO_CHn; 0 = 1 meas., 1 = 2 meas., 2 = 4 meas., 3 = 8 meas.

23.14 10-bit analog digital converter (ADC1) register definition

Note: HS2 and MON5 are device variant specific. In devices featuring only HS1 the HS2_XXX bitfields can be ignored. In devices featuring only MON1-4 the HS MON5_XXX bitfields can be ignored. Writing to these bitfields has no effect.

23.14.1 Register address space - ADC1

Table 176 Registers address space - ADC1

Module	Base address	End address	Note
ADC1	40004000_H	$40007FFF_H$	10-bit Analog Digital Converter (ADC1) registers

23.14.2 Register overview - ADC1 (ascending offset address)

Table 177 Register overview - ADC1 (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_CTRL_STS	ADC1 control and status register	0000_H	777
ADC1_SQ_FB	Sequencer feedback register	0004_H	779
ADC1_CHx_EIM	Channel setting bits for exceptional interrupt measurement register	0008_H	781
ADC1_CHx_ESM	Channel setting bits for exceptional sequence measurement register	$000C_H$	783
ADC1_MAX_TIME	Maximum time for software mode register	0010_H	785
ADC1_CTRL2	Measurement unit 1 control 2 register	0014_H	786
ADC1_CTRL3	Measurement unit 1 control 3 register	0018_H	787
ADC1_CTRL5	Measurement unit 1 control 5 register	$001C_H$	789
ADC1_SQ0_1	Measurement unit 1 channel enable bits for cycle 0-1 register	0020_H	790
ADC1_SQ2_3	Measurement unit 1 channel enable bits for cycle 2-3 register	0024_H	791
ADC1_SQ4_5	Measurement unit 1 channel enable bits for cycle 4-5 register	0028_H	792
ADC1_SQ6_7	Measurement unit 1 channel enable bits for cycle 6-7 register	$002C_H$	793
ADC1_SQ8_9	Measurement unit 1 channel enable bits for cycle 8-9 register	0030_H	794
ADC1_SQ10_11	Measurement unit 1 channel enable bits for cycle 10-11 register	0034_H	795
ADC1_CTRL4	Measurement unit 1 control 4 register	0038_H	796
ADC1_TH0_3_LOWER	Lower comparator trigger level channel 0-3	0040_H	823
ADC1_TH4_7_LOWER	Lower comparator trigger level channel 4-7	0044_H	824
ADC1_CAL_CH0_1	Calibration for channel 0 and 1 register	0048_H	797

(table continues...)

Table 177 (continued) Register overview - ADC1 (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_CAL_CH2_3	Calibration for channel 2 and 3 register	$004C_H$	798
ADC1_CAL_CH4_5	Calibration for channel 4 and 5 register	0050_H	799
ADC1_CAL_CH6_7	Calibration for channel 6 and 7 register	0054_H	800
ADC1_CAL_CH8_9	Calibration for channel 8 and 9 register	0058_H	801
ADC1_CAL_CH10_11	Calibration for channel 10 and 11 register	$005C_H$	802
ADC1_FILTCOEFF0_11	Filter coefficients measurement unit channel 0-11 register	0060_H	803
ADC1_IRQS_1	ADC1 interrupt status 1 register	0064_H	854
ADC1_IRQEN_1	ADC1 interrupt enable 1 register	0068_H	857
ADC1_IRQCLR_1	ADC1 interrupt status clear 1 register	$006C_H$	860
ADC1_FILT_OUT0	ADC1 or filter output channel 0 register	0070_H	805
ADC1_FILT_OUT1	ADC1 or filter output channel 1 register	0074_H	806
ADC1_FILT_OUT2	ADC1 or filter output channel 2 register	0078_H	807
ADC1_FILT_OUT3	ADC1 or filter output channel 3 register	$007C_H$	808
ADC1_FILT_OUT4	ADC1 or filter output channel 4 register	0080_H	809
ADC1_FILT_OUT5	ADC1 or filter output channel 5 register	0084_H	810
ADC1_FILT_OUT6	ADC1 or filter output channel 6 register	0088_H	811
ADC1_FILT_OUT7	ADC1 or filter output channel 7 register	$008C_H$	812
ADC1_FILT_OUT8	ADC1 or filter output channel 8 register	0090_H	813
ADC1_FILT_OUT9	ADC1 or filter output channel 9 register	0094_H	814
ADC1_FILT_OUT10	ADC1 or filter output channel 10 register	0098_H	815
ADC1_FILT_OUT11	ADC1 or filter output channel 11 register	$009C_H$	816
ADC1_DIFFCH_OUT1	ADC1 differential channel output 1 register	$00A0_H$	817
ADC1_DIFFCH_OUT2	ADC1 differential channel output 2 register	$00A4_H$	818
ADC1_DIFFCH_OUT3	ADC1 differential channel output 3 register	$00A8_H$	819
ADC1_DIFFCH_OUT4	ADC1 differential channel output 4 register	$00AC_H$	820
ADC1_FILT_UP_CTRL	Upper threshold filter enable	$00B0_H$	825
ADC1_FILT_LO_CTRL	Lower Threshold filter enable	$00B4_H$	827
ADC1_TH8_11_LOWER	Lower comparator trigger level channel 8-11	$00C0_H$	829
ADC1_DCHTH1_4_LOWER	Lower comparator trigger level differential channel 1-4 register	$00C4_H$	830
ADC1_TH0_3_UPPER	Upper comparator trigger level channel 0-3 register	$00C8_H$	831
ADC1_TH4_7_UPPER	Upper comparator trigger level channel 4-7 register	$00CC_H$	832
ADC1_TH8_11_UPPER	Upper comparator trigger level channel 8-11 register	$00D0_H$	833
		$00D4_H$	834

(table continues...)

Table 177 (continued) Register overview - ADC1 (ascending offset address)

Short name	Long name	Offset address	Page number
ADC1_DCHTH1_4_UPPER	Upper comparator trigger level differential channel 1-4 register		
ADC1_CNT0_3_LOWER	Lower counter trigger level channel 0-3 register	$00D8_H$	835
ADC1_CNT4_7_LOWER	Lower counter trigger level channel 4-7 register	$00DC_H$	837
ADC1_CNT8_11_LOWER	Lower counter trigger level channel 8-11 register	$00E0_H$	839
ADC1_DCHCNT1_4_LOWER	Lower counter trigger level differential channel 1-4 register	$00E4_H$	841
ADC1_CNT0_3_UPPER	Upper counter trigger level channel 0-3 register	$00E8_H$	843
ADC1_CNT4_7_UPPER	Upper counter trigger level channel 4-7 register	$00EC_H$	845
ADC1_CNT8_11_UPPER	Upper counter trigger level channel 8-11 register	$00F0_H$	847
ADC1_DCHCNT1_4_UPPER	Upper counter trigger level differential channel 1-4 register	$00F4_H$	849
ADC1_MMODE0_11	Overvoltage measurement mode of channel 0-11 register	$00F8_H$	851
ADC1_DUIN_SEL	Measurement unit 1 - Differential unit input selection register	$00FC_H$	879
ADC1_IRQS_2	ADC1 interrupt status 2 register	0100_H	863
ADC1_STS_2	ADC1 status 2 register	0104_H	866
ADC1_IRQCLR_2	ADC1 interrupt status clear 2 register	0108_H	869
ADC1_IRQEN_2	ADC1 interrupt enable 2 register	$010C_H$	872
ADC1_FILT_OUT12	ADC1 or filter output channel 12 register	0110_H	821
ADC1_FILT_OUTEIM	ADC1 or filter output of EIM register	0120_H	822
ADC1_STS_1	ADC1 status 1 register	0124_H	875
ADC1_STSCLR_1	ADC1 status clear 1 register	0128_H	877

23.14.3 ADC1 control and status register

ADC1_CTRL_STS Offset address: 0000_H

ADC1 control and status register RESET_TYPE_3 value: $0000\ 0000_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SW_CH_SEL				EOC	RES	CAL_ SIGN	REA DY	RES	SOS	RES	PD_ N
r				rw				rh	r	rh	r	r	rwh1	r	rw

Field	Bits	Type	Description
PD_N	0	rw	**ADC1 Power-down signal** 0_B **POWER_DOWN**: ADC1 is powered down 1_B **ACTIVE**: ADC1 is switched on
RES	1, 3, 6, 17:12, 31:19	r	**Reserved** Always write as 0
SOS	2	rwh1	**ADC1 Start of sampling/conversion (software mode)** *Note: Bit is set by software to start sampling and conversion and it is cleared by hardware once the conversion is finished. ADC1_SOS can be only written if the DPP is in software mode.* 0_B **DISABLE**: No conversion is started 1_B **ENABLE**: Conversion is started
READY	4	r	**HVADC 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
CAL_SIGN	5	rh	**Output of comparator to steer gain/offset calibration**
EOC	7	rh	**ADC1 End of Conversion (software mode)** *Note: This flag is not only cleared by a read operation but also automatically be setting ADC1_SOS.* 0_B **PENDING**: Conversion still running 1_B **FINISHED**: Conversion has finished
SW_CH_SEL	11:8	rw	**Channel for software mode** Other bit combinations are reserved, do not use.

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: An rfu combination is automatically mapped to channel 12; If channel number and SOS are written within one register write cycle the channel number is not immediately effective for the triggered conversion by SOS. 0_H **CH0_EN**: Channel 0 enable 1_H **CH1_EN**: Channel 1 enable 2_H **CH2_EN**: Channel 2 enable 3_H **CH3_EN**: Channel 3 enable 4_H **CH4_EN**: Channel 4 enable 5_H **CH5_EN**: Channel 5 enable 6_H **CH6_EN**: Channel 6 enable 7_H **CH7_EN**: Channel 7 enable 8_H **CH8_EN**: Channel 8 enable 9_H **CH9_EN**: Channel 9 enable A_H **CH10_EN**: Channel 10 enable B_H **CH11_EN**: Channel 11 enable C_H **CH12_EN**: Channel 12 enable D_H **RFU**: Reserved for future use … F_H **RFU**: Reserved for future use
STRTUP_DIS	18	rw	**DPP1 startup disable** 0_B **START_UP_ENABLE**: DPP1 start-up enabled 1_B **START_UP_DISABLE**: DPP1 start-up disable

23.14.4 Sequencer feedback register

ADC1_SQ_FB Offset address: 0004_H

Sequencer feedback register RESET_TYPE_4 value: $00XX\ XX0X_H$

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_STOP	RES				SQ_FB			
r	r				r	r	r	r				r			

Field	Bits	Type	Description
SQ_FB	4:0	r	**Current sequence that caused software mode** Other bit combinations are n.u., not used. 00_H **SQ0**: Sequence 0 enable 01_H **SQ1**: Sequence 1 enable 02_H **SQ2**: Sequence 2 enable 03_H **SQ3**: Sequence 3 enable 04_H **SQ4**: Sequence 4 enable 05_H **SQ5**: Sequence 5 enable 06_H **SQ6**: Sequence 6 enable 07_H **SQ7**: Sequence 7 enable 08_H **SQ8**: Sequence 8 enable 09_H **SQ9**: Sequence 9 enable $0A_H$ **SQ10**: Sequence 10 enable $0B_H$ **SQ11**: Sequence 11 enable $0C_H$ **ESM**: ESM $0D_H$ **RFU**: Reserved for future use $0E_H$ **SUSPEND_SW**: Software mode per flag $0F_H$ **SUSPEND_DSG**: Debug suspend mode
RES	7:5, 15, 31:20	r	**Reserved** Always read as 0
SQ_STOP	8	r	**ADC1 sequencer stop signal for DPP** 0_B **DPP_RUNNING**: Postprocessing sequencer in running mode 1_B **DPP_STOPPED**: Postprocessing sequencer stopped/software mode entered
EIM_ACTIVE	9	r	**ADC1 EIM active** *Note: This bit indicates an active or a pending exception measurement; a pending measurement is signalled when software mode is selected (mode with higher priority).*

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **NOT_ACTIVE**: EIM not active 1_B **ACTIVE**: EIM active
ESM_ACTIVE	10	r	**ADC1 ESM active** *Note: This bit indicates an active or a pending sequence measurement; a pending measurement is signalled when EIM or software mode is selected (modes with higher priority).* 0_B **NOT_ACTIVE**: ESM not active 1_B **ACTIVE**: ESM active
SQx	14:11	r	**Current active ADC1 sequence** Other bit combinations are reserved, do not use. 0_H **SQ0**: Sequence 0 enable 1_H **SQ1**: Sequence 1 enable 2_H **SQ2**: Sequence 2 enable 3_H **SQ3**: Sequence 3 enable 4_H **SQ4**: Sequence 4 enable 5_H **SQ5**: Sequence 5 enable 6_H **SQ6**: Sequence 6 enable 7_H **SQ7**: Sequence 7 enable 8_H **SQ8**: Sequence 8 enable 9_H **SQ9**: Sequence 9 enable A_H **SQ10**: Sequence 10 enable B_H **SQ11**: Sequence 11 enable
CHx	19:16	r	**Current ADC1 channel** Other bit combinations are reserved, do not use. 0_H **CH0**: Channel 0 enable 1_H **CH1**: Channel 1 enable 2_H **CH2**: Channel 2 enable 3_H **CH3**: Channel 3 enable 4_H **CH4**: Channel 4 enable 5_H **CH5**: Channel 5 enable 6_H **CH6**: Channel 6 enable 7_H **CH7**: Channel 7 enable 8_H **CH8**: Channel 8 enable 9_H **CH9**: Channel 9 enable A_H **CH10**: Channel 10 enable B_H **CH11**: Channel 11 enable

23.14.5 Channel setting bits for exceptional interrupt measurement register

ADC1_CHx_EIM

Channel setting bits for exceptional interrupt measurement register

Offset address: 0008_H

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													ADC1_EIM_TRIG_SEL		
r													rw		

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

Field	Bits	Type	Description
EIM_CHx	3:0	rw	**Channel set for exceptional interrupt measurement (EIM)** Other bit combinations are **n.u.**, not used. *Note: The selection of an rfu combination will be automatically mapped to CH11.* 0_H **CH0**: Channel 0 enable 1_H **CH1**: Channel 1 enable 2_H **CH2**: Channel 2 enable 3_H **CH3**: Channel 3 enable 4_H **CH4**: Channel 4 enable 5_H **CH5**: Channel 5 enable 6_H **CH6**: Channel 6 enable 7_H **CH7**: Channel 7 enable 8_H **CH8**: Channel 8 enable 9_H **CH9**: Channel 9 enable A_H **CH10**: Channel 10 enable B_H **CH11**: Channel 11 enable C_H **RFU**: Reserved for future use
RES	7:4, 15:12, 31:19	r	**Reserved** Always read as 0
EIM_REP	10:8	rw	**Repeat count for exceptional interrupt measurement (EIM)** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements

(table continues...)

(continued)

Field	Bits	Type	Description
			100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **64**: 64 measurements 111_B **128**: 128 measurements
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
ADC1_EIM_TRIG_SEL	18:16	rw	**Trigger selection for exceptional interrupt measurement (EIM)** 000_B **NONE**: None 001_B **COUT63**: COUT63 010_B **GPT12_T6OUT**: GPT12_T6OUT 011_B **GPT12_T3OUT**: GPT12_T3OUT 100_B **T2**: t2_adc_trigger 101_B **T21**: t21_adc_trigger 110_B **RES**: Reserved 111_B **RES**: Reserved

23.14.6 Channel setting bits for exceptional sequence measurement register

ADC1_CHx_ESM

Channel setting bits for exceptional sequence measurement register

Offset address: $000C_H$

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ESM_STS	**ESM_EN**	**RES**											**ADC1_ESM_TRIG_SEL**		
rwh	rw	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
ESM_0	11:0	rw	**Channel sequence for exceptional sequence measurement (ESM)** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 29:19	r	**Reserved** Always read as 0
ADC1_ESM_TRIG_SEL	18:16	rw	**Trigger selection for exceptional interrupt measurement (ESM)** 000_B **NONE**: None 001_B **COUT63**: COUT63 010_B **GPT12_T6OUT**: GPT12_T6OUT 011_B **GPT12_T3OUT**: GPT12_T3OUT 100_B **T2**: t2_adc_trigger 101_B **T21**: t21_adc_trigger 110_B **RES**: Reserved 111_B **RES**: Reserved
ESM_EN	30	rw	**Enable for Exceptional Sequence Measurement Trigger Event** 0_B **DISABLE**: Start of ESM disabled 1_B **ENABLE**: Start of ESM enabled
ESM_STS	31	rwh	**Exceptional sequence measurement is finished**

(table continues...)

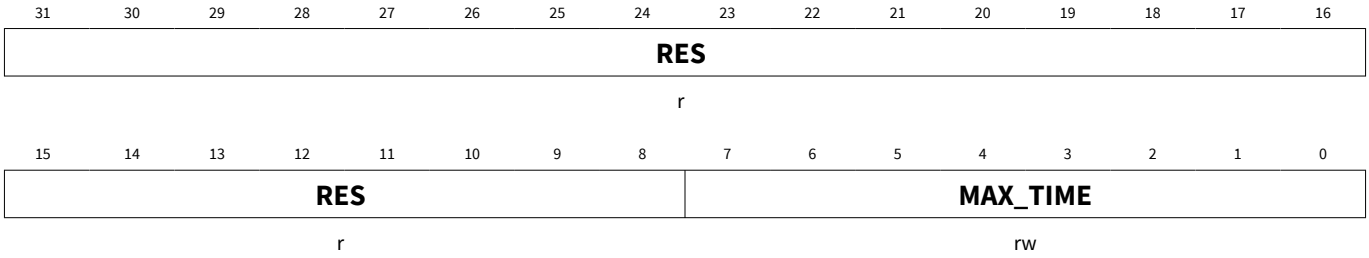
(continued)

Field	Bits	Type	Description
			Note: This bit has to be cleared, additionally to ESM_IS, before further ESM-interrupts can be triggered. 0_B **NOT_ACTIVE**: Exceptional Sequence Measurement not done 1_B **DONE**: Exceptional Sequence Measurement done

23.14.7 Maximum time for software mode register

ADC1_MAX_TIME Offset address: 0010_H

Maximum time for software mode register RESET_TYPE_4 value: $0000\ 0000_H$



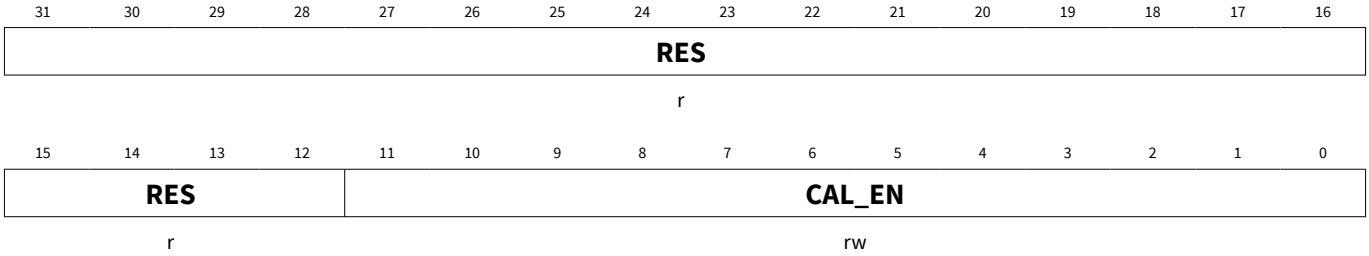
Field	Bits	Type	Description
MAX_TIME	7:0	rw	**Maximum Time in software mode** Maximum time in software mode with the unit of 50 ns Software mode is active for ADC1_MAX_TIME * 50 ns 00_H **MIN**: Software mode is immediately left FF_H **MAX**: Software mode is active for 12.75 us
RES	31:8	r	**Reserved** Always read as 0

23.14.8 Measurement unit 1 control 2 register

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.

ADC1_CTRL2 Offset address: 0014_H

Measurement unit 1 control 2 register RESET_TYPE_4 value: $0000\ 0000_H$



Field	Bits	Type	Description
CAL_EN	11:0	rw	**Calibration enable for channels 0 to 11** The following values can be ored: 001_H **CH0_EN**: Channel 0 calibration enable 002_H **CH1_EN**: Channel 1 calibration enable 004_H **CH2_EN**: Channel 2 calibration enable 008_H **CH3_EN**: Channel 3 calibration enable 010_H **CH4_EN**: Channel 4 calibration enable 020_H **CH5_EN**: Channel 5 calibration enable 040_H **CH6_EN**: Channel 6 calibration enable 080_H **CH7_EN**: Channel 7 calibration enable 100_H **CH8_EN**: Channel 8 calibration enable 200_H **CH9_EN**: Channel 9 calibration enable 400_H **CH10_EN**: Channel 10 calibration enable 800_H **CH11_EN**: Channel 11 calibration enable
RES	31:12	r	**Reserved** Always read as 0

23.14.9 Measurement unit 1 control 3 register

ADC1_CTRL3 Offset address: 0018_H

Measurement unit 1 control 3 register RESET_TYPE_4 value: $0000\ 0401_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SAMPLE_TIME_HVCH				MCM_RDY	EoC_FAIL	RES	EoC_FAIL_CLR	RES		SW_MODE	MCM_PD_N
r				rw				r	r	r	w	r		rw	rw

Field	Bits	Type	Description
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
SW_MODE	1	rw	**Software mode enable** 0_B **SOFTWARE_MODE_DISABLE**: Sequencer running 1_B **SOFTWARE_MODE_ENABLED**: Sequencer stopped
RES	3:2, 5, 15:12, 31:20	r	**Reserved** Always read as 0
EoC_FAIL_CLR	4	w	**Fail of ADC end of conversion signal clear** 0_B **ADC_EoC_FAIL_NOT_CLEAR**: No clear of EoC_FAIL flag 1_B **ADC_EoC_FAIL_CLEAR**: Clear of EoC_FAIL flag
EoC_FAIL	6	r	**Fail of ADC end of conversion signal** 0_B **ADC_EoC_AVAILABLE**: End of conversion signal was sent properly by ADC 1_B **ADC_EoC_NOT_AVAILABLE**: End of conversion signal was not sent properly by ADC
MCM_RDY	7	r	**Ready signal for MCM (Measurement core module) after power on or reset** 0_B **MCM_NOT_READY**: Measurement core module in start-up phase 1_B **MCM_READY**: Measurement core module start-up phase finished
SAMPLE_TIME_HVCH	11:8	rw	**Sample time of ADC1** *Note: The absolute sampling time of a high voltage channel should not be chosen lower than 2 us. Otherwise it is not ensured that the settling time of the input signal is long enough.* 0_H **MICLK4**: 4 ADC1_CLK clock periods

(table continues...)

(continued)

Field	Bits	Type	Description
			1_H **MICLK6**: 6 ADC1_CLK clock periods 2_H **MICLK8**: 8 ADC1_CLK clock periods 3_H **MICLK10**: 10 ADC1_CLK clock periods 4_H **MICLK12**: 12 ADC1_CLK clock periods (default) 5_H **MICLK14**: 14 ADC1_CLK clock periods 6_H **MICLK16**: 16 ADC1_CLK clock periods 7_H **MICLK18**: 18 ADC1_CLK clock periods 8_H **MICLK20**: 20 ADC1_CLK clock periods 9_H **MICLK22**: 22 ADC1_CLK clock periods A_H **MICLK4_1**: 4 ADC1_CLK clock periods … F_H **MICLK4_6**: 4 ADC1_CLK clock periods
SAMPLE_TIME_LVCH	19:16	rw	**Sample time of ADC1** *Note: The absolute sampling time of a low voltage channel should not be chosen lower than 80 ns. Otherwise it is not ensured that the settling time of the input signal is long enough.* 0_H **MICLK4**: 4 ADC1_CLK clock periods (default) 1_H **MICLK6**: 6 ADC1_CLK clock periods 2_H **MICLK8**: 8 ADC1_CLK clock periods 3_H **MICLK10**: 10 ADC1_CLK clock periods 4_H **MICLK12**: 12 ADC1_CLK clock periods 5_H **MICLK14**: 14 ADC1_CLK clock periods 6_H **MICLK16**: 16 ADC1_CLK clock periods 7_H **MICLK18**: 18 ADC1_CLK clock periods 8_H **MICLK20**: 20 ADC1_CLK clock periods 9_H **MICLK22**: 22 ADC1_CLK clock periods A_H **MICLK12_1**: 12 ADC1_CLK clock periods … F_H **MICLK12_6**: 12 ADC1_CLK clock periods

23.14.10 Measurement unit 1 control 5 register

ADC1_CTRL5 Offset address: $001C_H$

Measurement unit 1 control 5 register RESET_TYPE_4 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				FILT_OUT_SEL_11_0											
r				rw											

Field	Bits	Type	Description
FILT_OUT_SEL_11_0	11:0	rw	**Output filter selection for channels 0 to 11** 000_H **UNF**: ADC1 unfiltered data can be monitored in the corresponding FILT_OUTx registers 001_H **CH0**: Channel 0 IIR data enabled for FILT_OUT0 register 002_H **CH1**: Channel 1 IIR data enabled for FILT_OUT1 register 004_H **CH2**: Channel 2 IIR data enabled for FILT_OUT2 register 008_H **CH3**: Channel 3 IIR data enabled for FILT_OUT3 register 010_H **CH4**: Channel 4 IIR data enabled for FILT_OUT4 register 020_H **CH5**: Channel 5 IIR data enabled for FILT_OUT5 register 040_H **CH6**: Channel 6 IIR data enabled for FILT_OUT6 register 080_H **CH7**: Channel 7 IIR data enabled for FILT_OUT7 register 100_H **CH8**: Channel 8 IIR data enabled for FILT_OUT8 register 200_H **CH9**: Channel 9 IIR data enabled for FILT_OUT9 register 400_H **CH10**: Channel 10 IIR data enabled for FILT_OUT10 register 800_H **CH11**: Channel 11 IIR data enabled for FILT_OUT11 register FFF_H **CH11_0**: For channels 11-0 IIR data is enabled for FILT_OUTx registers
RES	31:12	r	**Reserved** Always read as 0

23.14.11 Measurement unit 1 channel enable bits for cycle 0-1 register

ADC1_SQ0_1

Measurement unit 1 channel enable bits for cycle 0-1 register

Offset address: 0020_H

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				SQ1											
r				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SQ0											
r				rw											

Field	Bits	Type	Description
SQ0	11:0	rw	**Sequence 0 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 31:28	r	**Reserved** Always read as 0
SQ1	27:16	rw	**Sequence 1 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable

23.14.12 Measurement unit 1 channel enable bits for cycle 2-3 register

ADC1_SQ2_3

Measurement unit 1 channel enable bits for cycle 2-3 register

Offset address: 0024_H

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				SQ3											
r				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SQ2											
r				rw											

Field	Bits	Type	Description
SQ2	11:0	rw	**Sequence 2 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 31:28	r	**Reserved** Always read as 0
SQ3	27:16	rw	**Sequence 3 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable

23.14.13 Measurement unit 1 channel enable bits for cycle 4-5 register

ADC1_SQ4_5

Offset address: 0028_H

Measurement unit 1 channel enable bits for cycle 4-5 register

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				SQ5											
r				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SQ4											
r				rw											

Field	Bits	Type	Description
SQ4	11:0	rw	**Sequence 4 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 31:28	r	**Reserved** Always read as 0
SQ5	27:16	rw	**Sequence 5 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable

23.14.14 Measurement unit 1 channel enable bits for cycle 6-7 register

ADC1_SQ6_7

Measurement unit 1 channel enable bits for cycle 6-7 register

Offset address: $002C_H$

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				SQ7											
r				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				SQ6											
r				rw											

Field	Bits	Type	Description
SQ6	11:0	rw	**Sequence 6 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 31:28	r	**Reserved** Always read as 0
SQ7	27:16	rw	**Sequence 7 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable

23.14.15 Measurement unit 1 channel enable bits for cycle 8-9 register

ADC1_SQ8_9

Measurement unit 1 channel enable bits for cycle 8-9 register

Offset address: 0030_H

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				**SQ9**											
r				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**SQ8**											
r				rw											

Field	Bits	Type	Description
SQ8	11:0	rw	**Sequence 8 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 31:28	r	**Reserved** Always read as 0
SQ9	27:16	rw	**Sequence 9 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable $7E8_H$ **CH11_EN**: Channel 11 enable

23.14.16 Measurement unit 1 channel enable bits for cycle 10-11 register

ADC1_SQ10_11

Measurement unit 1 channel enable bits for cycle 10-11 register

Offset address: 0034_H

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				**SQ11**											
r				rw											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**SQ10**											
r				rw											

Field	Bits	Type	Description
SQ10	11:0	rw	**Sequence 10 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable
RES	15:12, 31:28	r	**Reserved** Always read as 0
SQ11	27:16	rw	**Sequence 11 channel enable** The following values can be ored: 001_H **CH0_EN**: Channel 0 enable 002_H **CH1_EN**: Channel 1 enable 004_H **CH2_EN**: Channel 2 enable 008_H **CH3_EN**: Channel 3 enable 010_H **CH4_EN**: Channel 4 enable 020_H **CH5_EN**: Channel 5 enable 040_H **CH6_EN**: Channel 6 enable 080_H **CH7_EN**: Channel 7 enable 100_H **CH8_EN**: Channel 8 enable 200_H **CH9_EN**: Channel 9 enable 400_H **CH10_EN**: Channel 10 enable 800_H **CH11_EN**: Channel 11 enable

23.14.17 Measurement unit 1 control 4 register

ADC1_CTRL4 Offset address: 0038_H

Measurement unit 1 control 4 register RESET_TYPE_4 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												**MAX_CALTIME**			
r												rw			

Field	Bits	Type	Description
MAX_CALTIME	3:0	rw	**Maximum ADC calibration time** Defines how often the ADC calibration is done within the sequencer cycle. 0_H **1**: Sequence 1_H **2**: Sequences … F_H **16**: Sequences
RES	31:4	r	**Reserved** Always read as 0

23.14.18 Calibration for channel 0 and 1 register

ADC1_CAL_CH0_1 Offset address: 0048_H

Calibration for channel 0 and 1 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALGAIN_CH1								**RES**			**CALOFFS_CH1**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALGAIN_CH0								**RES**			**CALOFFS_CH0**				
rw								r			rw				

Field	Bits	Type	Description
CALOFFS_CH0	4:0	rw	**Offset calibration for channel 0** For ADC output set CALIB_EN_0 = 0
RES	7:5, 23:21	r	**Reserved** Always read as 0
CALGAIN_CH0	15:8	rw	**Gain calibration for channel 0** For ADC output set CALIB_EN_0 = 0
CALOFFS_CH1	20:16	rw	**Offset calibration for channel 1** For ADC output set CALIB_EN_1 = 0
CALGAIN_CH1	31:24	rw	**Gain calibration for channel 1** For ADC output set CALIB_EN_1 = 0

23.14.19 Calibration for channel 2 and 3 register

ADC1_CAL_CH2_3 Offset address: $004C_H$

Calibration for channel 2 and 3 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALGAIN_CH3								**RES**			**CALOFFS_CH3**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALGAIN_CH2								**RES**			**CALOFFS_CH2**				
rw								r			rw				

Field	Bits	Type	Description
CALOFFS_CH2	4:0	rw	**Offset calibration for channel 2** For ADC output set CALIB_EN_2 = 0
RES	7:5, 23:21	r	**Reserved** Always read as 0
CALGAIN_CH2	15:8	rw	**Gain calibration for channel 2** For ADC output set CALIB_EN_2 = 0
CALOFFS_CH3	20:16	rw	**Offset calibration for channel 3** For ADC output set CALIB_EN_3 = 0
CALGAIN_CH3	31:24	rw	**Gain calibration for channel 3** For ADC output set CALIB_EN_3 = 0

23.14.20 Calibration for channel 4 and 5 register

ADC1_CAL_CH4_5 Offset address: 0050_H

Calibration for channel 4 and 5 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALGAIN_CH5								**RES**			**CALOFFS_CH5**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALGAIN_CH4								**RES**			**CALOFFS_CH4**				
rw								r			rw				

Field	Bits	Type	Description
CALOFFS_CH4	4:0	rw	**Offset calibration for channel 4** For ADC output set CALIB_EN_4 = 0
RES	7:5, 23:21	r	**Reserved** Always read as 0
CALGAIN_CH4	15:8	rw	**Gain calibration for channel 4** For ADC output set CALIB_EN_4 = 0
CALOFFS_CH5	20:16	rw	**Offset calibration for channel 5** For ADC output set CALIB_EN_5 = 0
CALGAIN_CH5	31:24	rw	**Gain calibration for channel 5** For ADC output set CALIB_EN_5 = 0

23.14.21 Calibration for channel 6 and 7 register

ADC1_CAL_CH6_7 Offset address: 0054_H

Calibration for channel 6 and 7 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALGAIN_CH7								**RES**			**CALOFFS_CH7**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALGAIN_CH6								**RES**			**CALOFFS_CH6**				
rw								r			rw				

Field	Bits	Type	Description
CALOFFS_CH6	4:0	rw	**Offset calibration for channel 6** For ADC output set CALIB_EN_6 = 0
RES	7:5, 23:21	r	**Reserved** Always read as 0
CALGAIN_CH6	15:8	rw	**Gain calibration for channel 6** For ADC output set CALIB_EN_6 = 0
CALOFFS_CH7	20:16	rw	**Offset calibration for channel 7** For ADC output set CALIB_EN_7 = 0
CALGAIN_CH7	31:24	rw	**Gain calibration for channel 7** For ADC output set CALIB_EN_7 = 0

23.14.22 Calibration for channel 8 and 9 register

ADC1_CAL_CH8_9 Offset address: 0058_H

Calibration for channel 8 and 9 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALGAIN_CH9								**RES**			**CALOFFS_CH9**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALGAIN_CH8								**RES**			**CALOFFS_CH8**				
rw								r			rw				

Field	Bits	Type	Description
CALOFFS_CH8	4:0	rw	**Offset calibration for channel 8** For ADC output set CALIB_EN_8 = 0
RES	7:5, 23:21	r	**Reserved** Always read as 0
CALGAIN_CH8	15:8	rw	**Gain calibration for channel 8** For ADC output set CALIB_EN_8 = 0
CALOFFS_CH9	20:16	rw	**Offset calibration for channel 9** For ADC output set CALIB_EN_9 = 0
CALGAIN_CH9	31:24	rw	**Gain calibration for channel 9** For ADC output set CALIB_EN_9 = 0

23.14.23 Calibration for channel 10 and 11 register

ADC1_CAL_CH10_11 Offset address: $005C_H$

Calibration for channel 10 and 11 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALGAIN_CH11								**RES**			**CALOFFS_CH11**				
rw								r			rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALGAIN_CH10								**RES**			**CALOFFS_CH10**				
rw								r			rw				

Field	Bits	Type	Description
CALOFFS_CH10	4:0	rw	**Offset calibration for channel 10** For ADC output set CALIB_EN_10 = 0
RES	7:5, 23:21	r	**Reserved** Always read as 0
CALGAIN_CH10	15:8	rw	**Gain calibration for channel 10** For ADC output set CALIB_EN_10 = 0
CALOFFS_CH11	20:16	rw	**Offset calibration for channel 11** For ADC output set CALIB_EN_11 = 0
CALGAIN_CH11	31:24	rw	**Gain calibration for channel 11** For ADC output set CALIB_EN_11 = 0

23.14.24 Filter coefficients measurement unit channel 0-11 register

ADC1_FILTCOEFF0_11

Filter coefficients measurement unit channel 0-11 register

Offset address: 0060_H

RESET_TYPE_4 value: $00AA\ AAAA_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								CH11		CH10		CH9		CH8	
r								rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
rw		rw		rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
CH0	1:0	rw	**Filter coefficients ADC channel 0** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH1	3:2	rw	**Filter coefficients ADC channel 1** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH2	5:4	rw	**Filter coefficients ADC channel 2** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH3	7:6	rw	**Filter coefficients ADC channel 3** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH4	9:8	rw	**Filter coefficients ADC channel 4** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH5	11:10	rw	**Filter coefficients ADC channel 5** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample

(table continues...)

(continued)

Field	Bits	Type	Description
			11_B **1_16**: 1/16 weight of current sample
CH6	13:12	rw	**Filter Coefficients ADC channel 6** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH7	15:14	rw	**Filter coefficients ADC channel 7** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH8	17:16	rw	**Filter coefficients ADC channel 8** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH9	19:18	rw	**Filter coefficients ADC channel 9** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH10	21:20	rw	**Filter coefficients ADC channel 10** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
CH11	23:22	rw	**Filter coefficients ADC channel 11** 00_B **1_2**: 1/2 weight of current sample 01_B **1_4**: 1/4 weight of current sample 10_B **1_8**: 1/8 weight of current sample 11_B **1_16**: 1/16 weight of current sample
RES	31:24	r	**Reserved** Always read as 0

23.14.25 ADC1 or filter output channel 0 register

This registers reflects the current value of channel 0 of the measurement chain, which is assigned to VBAT_SENSE.

ADC1_FILT_OUT0 Offset address: 0070_H

ADC1 or filter output channel 0 register RESET_TYPE_3 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				FILT_OUT_CH0											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH0	11:0	r	**ADC or filter output value channel 0** For ADC output set ADC1_FILTUP_0_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF0	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH0. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH0 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH0 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.26 ADC1 or filter output channel 1 register

ADC1_FILT_OUT1 Offset address: 0074_H

ADC1 or filter output channel 1 register RESET_TYPE_3 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				FILT_OUT_CH1											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH1	11:0	r	**ADC or filter output value channel 1** For ADC output set ADC1_FILTUP_1_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF1	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH1. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH1 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH1 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.27 ADC1 or filter output channel 2 register

ADC1_FILT_OUT2 Offset address: 0078_H

ADC1 or filter output channel 2 register RESET_TYPE_3 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				FILT_OUT_CH2											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH2	11:0	r	**ADC or filter output value channel 2** For ADC output set ADC1_FILTUP_2_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF2	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH2. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH2 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH2 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.28 ADC1 or filter output channel 3 register

ADC1_FILT_OUT3 Offset address: $007C_H$

ADC1 or filter output channel 3 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF3	VF3	WFR 3
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH3											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH3	11:0	r	**ADC or filter output value channel 3** For ADC output set ADC1_FILTUP_3_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF3	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH3. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH3 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH3 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.29 ADC1 or filter output channel 4 register

ADC1_FILT_OUT4 Offset address: 0080_H

ADC1 or filter output channel 4 register RESET_TYPE_3 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				FILT_OUT_CH4											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH4	11:0	r	**ADC or filter output value channel 4** For ADC output set ADC1_FILTUP_4_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF4	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH4. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH4 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH4 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.30 ADC1 or filter output channel 5 register

ADC1_FILT_OUT5 Offset address: 0084_H

ADC1 or filter output channel 5 register RESET_TYPE_3 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				FILT_OUT_CH5											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH5	11:0	r	**ADC or filter output value channel 5** For ADC output set ADC1_FILTUP_5_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF5	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH5. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH5 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH5 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.31 ADC1 or filter output channel 6 register

ADC1_FILT_OUT6 Offset address: 0088_H

ADC1 or filter output channel 6 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF6	VF6	WFR 6
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH6											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH6	11:0	r	**ADC or filter output value channel 6** For ADC output set ADC1_FILTUP_6_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF6	17	rh	**Valid flag** Indicates valid contents in result register-bit field ADC1_OUT_CH6. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH6 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH6 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.32 ADC1 or filter output channel 7 register

ADC1_FILT_OUT7 Offset address: $008C_H$

ADC1 or filter output channel 7 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF7	VF7	WFR 7
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH7											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH7	11:0	r	**ADC or filter output value channel 7** For ADC output set ADC1_FILTUP_7_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF7	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH7. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH7 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH7 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.33 ADC1 or filter output channel 8 register

ADC1_FILT_OUT8 Offset address: 0090_H

ADC1 or filter output channel 8 register RESET_TYPE_3 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				FILT_OUT_CH8											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH8	11:0	r	**ADC or filter output value channel 8** For ADC output set ADC1_FILTUP_8_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
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
VF8	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH8. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH8 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
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 FILT_OUT_CH8 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.34 ADC1 or filter output channel 9 register

ADC1_FILT_OUT9 Offset address: 0094_H

ADC1 or filter output channel 9 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF9	VF9	WFR 9
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH9											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH9	11:0	r	**ADC or filter output value channel 9** For ADC output set ADC1_FILTUP_9_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
WFR9	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
VF9	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH9. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH9 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
OF9	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 FILT_OUT_CH9 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.35 ADC1 or filter output channel 10 register

ADC1_FILT_OUT10 Offset address: 0098_H

ADC1 or filter output channel 10 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF10	VF10	WFR 10
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH10											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH10	11:0	r	**ADC or filter output value channel 10** For ADC output set ADC1_FILTUP_10_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
WFR10	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
VF10	17	rh	**Valid flag** Indicates valid contents in result register bit field ADC1_OUT_CH10. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH10 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
OF10	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 FILT_OUT_CH10 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.36 ADC1 or filter output channel 11 register

ADC1_FILT_OUT11 Offset address: $009C_H$

ADC1 or filter output channel 11 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF11	VF11	WFR 11
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH11											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH11	11:0	r	**ADC or filter output value channel 11** For ADC output set ADC1_FILTUP_11_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
WFR11	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
VF11	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH11. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH11 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
OF11	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 FILT_OUT_CH11 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.37 ADC1 differential channel output 1 register

ADC1_DIFFCH_OUT1 Offset address: $00A0_H$

ADC1 differential channel output 1 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													DOF1	DVF1	DWFR1
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				DCH1											
r				r											

Field	Bits	Type	Description
DCH1	11:0	r	**ADC differential output value 1**
RES	15:12, 31:19	r	**Reserved** Always read as 0
DWFR1	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
DVF1	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_DOUT1. *Note: Bit is set by hardware on update of result register and it is cleared by software once the DCH1 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
DOF1	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 DCH1 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.38 ADC1 differential channel output 2 register

ADC1_DIFFCH_OUT2 Offset address: $00A4_H$

ADC1 differential channel output 2 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													DOF2	DVF2	DWFR2
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				DCH2											
r				r											

Field	Bits	Type	Description
DCH2	11:0	r	**ADC differential output value 2**
RES	15:12, 31:19	r	**Reserved** Always read as 0
DWFR2	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
DVF2	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_DOUT2. *Note: Bit is set by hardware on update of result register and it is cleared by software once the DCH2 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
DOF2	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 DCH2 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.39 ADC1 differential channel output 3 register

ADC1_DIFFCH_OUT3 Offset address: $00A8_H$

ADC1 differential channel output 3 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													DOF3	DVF3	DWFR3
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				DCH3											
r				r											

Field	Bits	Type	Description
DCH3	11:0	r	**ADC differential output value 3**
RES	15:12, 31:19	r	**Reserved** Always read as 0
DWFR3	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
DVF3	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_DOUT3. *Note: Bit is set by hardware on update of result register and it is cleared by software once the DCH3 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
DOF3	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 DCH3 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.40 ADC1 differential channel output 4 register

ADC1_DIFFCH_OUT4 Offset address: $00AC_H$

ADC1 differential channel output 4 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													DOF4	DVF4	DWFR4
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				DCH4											
r				r											

Field	Bits	Type	Description
DCH4	11:0	r	**ADC differential output value 4**
RES	15:12, 31:19	r	**Reserved** Always read as 0
DWFR4	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
DVF4	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_DOUT4. *Note: Bit is set by hardware on update of result register and it is cleared by software once the DCH4 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
DOF4	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 DCH4 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.41 ADC1 or filter output channel 12 register

ADC1_FILT_OUT12 Offset address: 0110_H

ADC1 or filter output channel 12 register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF12	VF12	WFR 12
r													r	rh	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				FILT_OUT_CH12											
r				r											

Field	Bits	Type	Description
FILT_OUT_CH12	11:0	r	**ADC or filter output value channel 12** For ADC output set ADC1_FILTUP_12_EN = 0
RES	15:12, 31:19	r	**Reserved** Always read as 0
WFR12	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
VF12	17	rh	**Valid flag** Indicates valid contents in result register bit-field ADC1_OUT_CH12. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_CH12 register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
OF12	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 FILT_OUT_CH12 register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.42 ADC1 or filter output of EIM register

Note: This channel is not included in the sequencer. EIM Mode uses the postprocessing chain of the selected EIM channel.

ADC1_FILT_OUTEIM Offset address: 0120_H

ADC1 or filter output of EIM register RESET_TYPE_3 value: $0000\ 0XXX_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES													OF_EIM	VF_EIM	WFR_EIM
r													r	rh	rw

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

Field	Bits	Type	Description
FILT_OUT_EIM	11:0	r	**ADC or filter output value for last EIM measurement**
RES	15:12, 31:19	r	**Reserved** Always read as 0
WFR_EIM	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
VF_EIM	17	rh	**Valid flag** Indicates valid contents in result register bit-field of last EIM measurement. *Note: Bit is set by hardware on update of result register and it is cleared by software once the FILT_OUT_EIM register is read. The hardware update has priority than the software read in case the event occurs at the same cycle.* 0_B **NOT_VALID**: No new valid data available 1_B **VALID**: Result register contains valid data and has not yet been read
OF_EIM	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 FILT_OUT_EIM register.* 0_B **NO_OVERRUN**: Result register not overwritten 1_B **OVERRUN**: Result register overwritten

23.14.43 Lower comparator trigger level channel 0-3

ADC1_TH0_3_LOWER Offset address: 0040_H

Lower comparator trigger level channel 0-3 RESET_TYPE_4 value: $1D2F\ 423A_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3_LOW								CH2_LOW							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1_LOW								CH0_LOW							
rw								rw							

Field	Bits	Type	Description
CH0_LOW	7:0	rw	**Channel 0 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH1_LOW	15:8	rw	**Channel 1 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH2_LOW	23:16	rw	**Channel 2 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH3_LOW	31:24	rw	**Channel 3 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value

23.14.44 Lower comparator trigger level channel 4-7

ADC1_TH4_7_LOWER Offset address: 0044_H

Lower comparator trigger level channel 4-7 RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH7_LOW								CH6_LOW							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH5_LOW								CH4_LOW							
rw								rw							

Field	Bits	Type	Description
CH4_LOW	7:0	rw	**Channel 4 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH5_LOW	15:8	rw	**Channel 5 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH6_LOW	23:16	rw	**Channel 6 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH7_LOW	31:24	rw	**Channel 7 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value

23.14.45 Upper threshold filter enable

ADC1_FILT_UP_CTRL Offset address: $00B0_H$

Upper threshold filter enable RESET_TYPE_4 value: $0000\ FFFF_H$

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
RES1				FU_CH11_EN	FU_CH10_EN	FU_CH9_EN	FU_CH8_EN	FU_CH7_EN	FU_CH6_EN	FU_CH5_EN	FU_CH4_EN	FU_CH3_EN	FU_CH2_EN	FU_CH1_EN	FU_CH0_EN
r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
FU_CH0_EN	0	rw	**Upper threshold IIR filter enable channel 0** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH1_EN	1	rw	**Upper threshold IIR filter enable channel 1** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH2_EN	2	rw	**Upper threshold IIR filter enable channel 2** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH3_EN	3	rw	**Upper threshold IIR filter enable channel 3** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH4_EN	4	rw	**Upper threshold IIR filter enable channel 4** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH5_EN	5	rw	**Upper threshold IIR filter enable channel 5** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH6_EN	6	rw	**Upper threshold IIR filter enable channel 6** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH7_EN	7	rw	**Upper threshold IIR filter enable channel 7** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH8_EN	8	rw	**Upper threshold IIR filter enable channel 8** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable

(table continues...)

(continued)

Field	Bits	Type	Description
FU_CH9_EN	9	rw	**Upper threshold IIR filter enable channel 9** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH10_EN	10	rw	**Upper threshold IIR filter enable channel 10** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FU_CH11_EN	11	rw	**Upper threshold IIR filter enable channel 11** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
RES1	15:12	r	**Reserved** Always read as 1
RES	31:16	r	**Reserved** Always read as 0

23.14.46 Lower Threshold filter enable

Setting this register enables the IIR filter structure for the postprocessing of the lower threshold. This can be used for example as shutdown signal for the system, in case of supply loss.

ADC1_FILT_LO_CTRL Offset address: $00B4_H$

Lower Threshold filter enable RESET_TYPE_4 value: $0000\ FFFF_H$

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				FL_CH11_EN	FL_CH10_EN	FL_CH9_EN	FL_CH8_EN	FL_CH7_EN	FL_CH6_EN	FL_CH5_EN	FL_CH4_EN	FL_CH3_EN	FL_CH2_EN	FL_CH1_EN	FL_CH0_EN
r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
FL_CH0_EN	0	rw	**Lower threshold IIR filter enable channel 0** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH1_EN	1	rw	**Lower threshold IIR filter enable channel 1** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH2_EN	2	rw	**Lower threshold IIR filter enable channel 2** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH3_EN	3	rw	**Lower threshold IIR filter enable channel 3** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH4_EN	4	rw	**Lower threshold IIR filter enable channel 4** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH5_EN	5	rw	**Lower threshold IIR filter enable channel 5** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH6_EN	6	rw	**Lower threshold IIR filter enable channel 6** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH7_EN	7	rw	**Lower threshold IIR filter enable channel 7** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH8_EN	8	rw	**Lower threshold IIR filter enable channel 8**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH9_EN	9	rw	**Lower threshold IIR filter enable channel 9** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH10_EN	10	rw	**Lower threshold IIR filter enable channel 10** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
FL_CH11_EN	11	rw	**Lower threshold IIR filter enable channel 11** 0_B **DISABLE**: Disable 1_B **ENABLE**: Enable
RES	31:12	r	**Reserved** Always read as 0

23.14.47 Lower comparator trigger level channel 8-11

ADC1_TH8_11_LOWER Offset address: $00C0_H$

Lower comparator trigger level channel 8-11 RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH11_LOW								**CH10_LOW**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH9_LOW								**CH8_LOW**							
rw								rw							

Field	Bits	Type	Description
CH8_LOW	7:0	rw	**Channel 8 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH9_LOW	15:8	rw	**Channel 9 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH10_LOW	23:16	rw	**Channel 10 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
CH11_LOW	31:24	rw	**Channel 11 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value

23.14.48 Lower comparator trigger level differential channel 1-4 register

ADC1_DCHTH1_4_LOWER

Lower comparator trigger level differential channel 1-4 register

Offset address: $00C4_H$

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCH4_LOW								DCH3_LOW							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCH2_LOW								DCH1_LOW							
rw								rw							

Field	Bits	Type	Description
DCH1_LOW	7:0	rw	**Differential channel 1 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
DCH2_LOW	15:8	rw	**Differential channel 2 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
DCH3_LOW	23:16	rw	**Differential channel 3 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value
DCH4_LOW	31:24	rw	**Differential channel 4 lower trigger level** 00_H **MIN**: Min. threshold value FF_H **MAX**: Max. threshold value

23.14.49 Upper comparator trigger level channel 0-3 register

ADC1_TH0_3_UPPER Offset address: $00C8_H$

Upper comparator trigger level channel 0-3 register RESET_TYPE_4 value: $AB8D\ C5C0_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH3_UP								**CH2_UP**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH1_UP								**CH0_UP**							
rw								rw							

Field	Bits	Type	Description
CH0_UP	7:0	rw	**Channel 0 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH1_UP	15:8	rw	**Channel 1 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH2_UP	23:16	rw	**Channel 2 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH3_UP	31:24	rw	**Channel 3 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255

23.14.50 Upper comparator trigger level channel 4-7 register

ADC1_TH4_7_UPPER Offset address: $00CC_H$

Upper comparator trigger level channel 4-7 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH7_UP								CH6_UP							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH5_UP								CH4_UP							
rw								rw							

Field	Bits	Type	Description
CH4_UP	7:0	rw	**Channel 4 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH5_UP	15:8	rw	**Channel 5 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH6_UP	23:16	rw	**Channel 6 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH7_UP	31:24	rw	**Channel 7 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255

23.14.51 Upper comparator trigger level channel 8-11 register

ADC1_TH8_11_UPPER Offset address: $00D0_H$

Upper comparator trigger level channel 8-11 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CH11_UP								**CH10_UP**							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CH9_UP								**CH8_UP**							
rw								rw							

Field	Bits	Type	Description
CH8_UP	7:0	rw	**Channel 8 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH9_UP	15:8	rw	**Channel 9 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH10_UP	23:16	rw	**Channel 10 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
CH11_UP	31:24	rw	**Channel 11 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255

23.14.52 Upper comparator trigger level differential channel 1-4 register

ADC1_DCHTH1_4_UPPER Offset address: $00D4_H$

Upper comparator trigger level differential channel 1-4 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCH4_UP								DCH3_UP							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCH2_UP								DCH1_UP							
rw								rw							

Field	Bits	Type	Description
DCH1_UP	7:0	rw	**Differential channel 1 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
DCH2_UP	15:8	rw	**Differential channel 2 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
DCH3_UP	23:16	rw	**Differential channel 3 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255
DCH4_UP	31:24	rw	**Differential channel 4 upper trigger level** 00_H **MIN**: Min. threshold value = 0 FF_H **MAX**: Max. threshold value = 255

23.14.53 Lower counter trigger level channel 0-3 register

ADC1_CNT0_3_LOWER Offset address: $00D8_H$

Lower counter trigger level channel 0-3 register RESET_TYPE_4 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
CNT_LO_CH0	2:0	rw	**Lower timer trigger threshold channel 0** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH0	4:3	rw	**Channel 0 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_LO_CH1	10:8	rw	**Lower timer trigger threshold channel 1** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH1	12:11	rw	**Channel 1 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH2	18:16	rw	**Lower timer trigger threshold channel 2** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH2	20:19	rw	**Channel 2 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH3	26:24	rw	**Lower timer trigger threshold channel 3** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH3	28:27	rw	**Channel 3 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.54 Lower counter trigger level channel 4-7 register

ADC1_CNT4_7_LOWER Offset address: $00DC_H$

Lower counter trigger level channel 4-7 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_LO_CH7		CNT_LO_CH7			RES			HYST_LO_CH6		CNT_LO_CH6		
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_CH5		CNT_LO_CH5			RES			HYST_LO_CH4		CNT_LO_CH4		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_LO_CH4	2:0	rw	**Lower timer trigger threshold channel 4** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH4	4:3	rw	**Channel 4 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_LO_CH5	10:8	rw	**Lower timer trigger threshold channel 5** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH5	12:11	rw	**Channel 5 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH6	18:16	rw	**Lower timer trigger threshold channel 6** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH6	20:19	rw	**Channel 6 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH7	26:24	rw	**Lower timer trigger threshold channel 7** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH7	28:27	rw	**Channel 7 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.55 Lower counter trigger level channel 8-11 register

ADC1_CNT8_11_LOWER Offset address: $00E0_H$

Lower counter trigger level channel 8-11 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_LO_CH11		CNT_LO_CH11			RES			HYST_LO_CH10		CNT_LO_CH10		
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_CH9		CNT_LO_CH9			RES			HYST_LO_CH8		CNT_LO_CH8		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_LO_CH8	2:0	rw	**Lower timer trigger threshold channel 8** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH8	4:3	rw	**Channel 8 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_LO_CH9	10:8	rw	**Lower timer trigger threshold channel 9** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH9	12:11	rw	**Channel 9 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH10	18:16	rw	**Lower timer trigger threshold channel 10** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH10	20:19	rw	**Channel 10 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_CH11	26:24	rw	**Lower timer trigger threshold channel 11** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_CH11	28:27	rw	**Channel 11 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.56 Lower counter trigger level differential channel 1-4 register

ADC1_DCHCNT1_4_LOWER

Lower counter trigger level differential channel 1-4 register

Offset address: $00E4_H$

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_LO_DCH4		CNT_LO_DCH4			RES			HYST_LO_DCH3		CNT_LO_DCH3		
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_DCH2		CNT_LO_DCH2			RES			HYST_LO_DCH1		CNT_LO_DCH1		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_LO_DCH1	2:0	rw	**Lower timer trigger threshold differential channel 1** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_DCH1	4:3	rw	**Differential Channel 1 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_LO_DCH2	10:8	rw	**Lower timer trigger threshold differential channel 2** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_DCH2	12:11	rw	**Differential Channel 2 lower hysteresis**

(table continues...)

(continued)

Field	Bits	Type	Description
			00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_DCH3	18:16	rw	**Lower timer trigger threshold differential channel 3** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_DCH3	20:19	rw	**Differential Channel 3 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_LO_DCH4	26:24	rw	**Lower timer trigger threshold differential channel 4** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_LO_DCH4	28:27	rw	**Differential Channel 4 lower hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.57 Upper counter trigger level channel 0-3 register

ADC1_CNT0_3_UPPER Offset address: $00E8_H$

Upper counter trigger level channel 0-3 register RESET_TYPE_4 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
CNT_UP_CH0	2:0	rw	**Upper timer trigger threshold channel 0** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH0	4:3	rw	**Channel 0 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_UP_CH1	10:8	rw	**Upper timer trigger threshold channel 1** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH1	12:11	rw	**Channel 1 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH2	18:16	rw	**Upper timer trigger threshold channel 2** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH2	20:19	rw	**Channel 2 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH3	26:24	rw	**Upper timer trigger threshold channel 3** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH3	28:27	rw	**Channel 3 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.58 Upper counter trigger level channel 4-7 register

ADC1_CNT4_7_UPPER Offset address: $00EC_H$

Upper counter trigger level channel 4-7 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_UP_CH7		CNT_UP_CH7			RES			HYST_UP_CH6		CNT_UP_CH6		
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_CH5		CNT_UP_CH5			RES			HYST_UP_CH4		CNT_UP_CH4		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_UP_CH4	2:0	rw	**Upper timer trigger threshold channel 4** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH4	4:3	rw	**Channel 4 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_UP_CH5	10:8	rw	**Upper timer trigger threshold channel 5** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH5	12:11	rw	**Channel 5 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH6	18:16	rw	**Upper timer trigger threshold channel 6** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH6	20:19	rw	**Channel 6 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH7	26:24	rw	**Upper timer trigger threshold channel 7** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH7	28:27	rw	**Channel 7 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.59 Upper counter trigger level channel 8-11 register

ADC1_CNT8_11_UPPER Offset address: $00F0_H$

Upper counter trigger level channel 8-11 register RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_UP_CH11		CNT_UP_CH11			RES			HYST_UP_CH10		CNT_UP_CH10		
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_CH9		CNT_UP_CH9			RES			HYST_UP_CH8		CNT_UP_CH8		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_UP_CH8	2:0	rw	**Upper timer trigger threshold channel 8** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH8	4:3	rw	**Channel 8 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_UP_CH9	10:8	rw	**Upper timer trigger threshold channel 9** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH9	12:11	rw	**Channel 9 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off

(table continues...)

(continued)

Field	Bits	Type	Description
			01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH10	18:16	rw	**Upper timer trigger threshold channel 10** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH10	20:19	rw	**Channel 10 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_CH11	26:24	rw	**Upper timer trigger threshold channel 11** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **16**: 16 measurements 101_B **32**: 32 measurements 110_B **63**: 63 measurements 111_B **63**: 63 measurements
HYST_UP_CH11	28:27	rw	**Channel 11 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.60 Upper counter trigger level differential channel 1-4 register

ADC1_DCHCNT1_4_UPPER

Upper counter trigger level differential channel 1-4 register

Offset address: $00F4_H$

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			HYST_UP_DCH4		CNT_UP_DCH4			RES			HYST_UP_DCH3		CNT_UP_DCH3		
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_DCH2		CNT_UP_DCH2			RES			HYST_UP_DCH1		CNT_UP_DCH1		
r			rw		rw			r			rw		rw		

Field	Bits	Type	Description
CNT_UP_DCH1	2:0	rw	**Upper timer trigger threshold differential channel 1** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **15**: 15 measurements … 111_B **15**: 15 measurements
HYST_UP_DCH1	4:3	rw	**Differential channel 1 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
RES	7:5, 15:13, 23:21, 31:29	r	**Reserved** Always read as 0
CNT_UP_DCH2	10:8	rw	**Upper timer trigger threshold differential channel 2** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **15**: 15 measurements … 111_B **15**: 15 measurements
HYST_UP_DCH2	12:11	rw	**Differential channel 2 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4

(table continues...)

(continued)

Field	Bits	Type	Description
			10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_DCH3	18:16	rw	**Upper timer trigger threshold differential channel 3** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **15**: 15 measurements … 111_B **15**: 15 measurements
HYST_UP_DCH3	20:19	rw	**Differential channel 3 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16
CNT_UP_DCH4	26:24	rw	**Upper timer trigger threshold differential channel 4** 000_B **1**: 1 measurement 001_B **2**: 2 measurements 010_B **4**: 4 measurements 011_B **8**: 8 measurements 100_B **15**: 15 measurements … 111_B **15**: 15 measurements
HYST_UP_DCH4	28:27	rw	**Differential channel 4 upper hysteresis** 00_B **HYSTOFF**: Hysteresis switched off 01_B **HYST4**: Hysteresis = 4 10_B **HYST8**: Hysteresis = 8 11_B **HYST16**: Hysteresis = 16

23.14.61 Overvoltage measurement mode of channel 0-11 register

ADC1_MMODE0_11

Overvoltage measurement mode of channel 0-11 register

Offset address: $00F8_H$

RESET_TYPE_4 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MMODE_D4		MMODE_D3		MMODE_D2		MMODE_D1		MMODE_11		MMODE_10		MMODE_9		MMODE_8	
rw		rw		rw		rw		rw		rw		rw		rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMODE_7		MMODE_6		MMODE_5		MMODE_4		MMODE_3		MMODE_2		MMODE_1		MMODE_0	
rw		rw		rw		rw		rw		rw		rw		rw	

Field	Bits	Type	Description
MMODE_0	1:0	rw	**Measurement mode channel 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
MMODE_1	3:2	rw	**Measurement mode channel 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
MMODE_2	5:4	rw	**Measurement mode channel 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
MMODE_3	7:6	rw	**Measurement mode channel 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
MMODE_4	9:8	rw	**Measurement mode channel 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
MMODE_5	11:10	rw	**Measurement mode channel 5** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement

(table continues...)

(continued)

Field	Bits	Type	Description
			11_B **RESERVED**: Reserved
MMODE_6	13:12	rw	**Measurement mode channel 6** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved
MMODE_7	15:14	rw	**Measurement mode channel 7** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved
MMODE_8	17:16	rw	**Measurement mode channel 8** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved
MMODE_9	19:18	rw	**Measurement mode channel 9** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved
MMODE_10	21:20	rw	**Measurement mode channel 10** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved
MMODE_11	23:22	rw	**Measurement mode channel 11** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement 11_B **RESERVED**: Reserved
MMODE_D1	25:24	rw	**Measurement mode differential channel 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
MMODE_D2	27:26	rw	**Measurement mode differential channel 2** 00_B **MMODE0**: Upper & lower voltage/limit measurement 01_B **MMODEUV**: Undervoltage/-limit measurement 10_B **MMODEOV**: Overvoltage/-limit measurement

(table continues...)

(continued)

Field	Bits	Type	Description
			11_B **RESERVED**: Reserved
MMODE_D3	29:28	rw	**Measurement mode differential channel 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
MMODE_D4	31:30	rw	**Measurement mode differential channel 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

23.14.62 ADC1 interrupt status 1 register

ADC1_IRQS_1 Offset address: 0064_H

ADC1 interrupt status 1 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DU4UP_IS	DU4LO_IS	DU3UP_IS	DU3LO_IS	DU2UP_IS	DU2LO_IS	DU1UP_IS	DU1LO_IS	RES						ESM_IS	EIM_IS
rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	r						rwhxre	rwhxre

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			P2_0_IS	P2_7_IS	P2_6_IS	P2_3_IS	P2_2_IS	P2_1_IS	MON5_IS	MON4_IS	MON3_IS	MON2_IS	MON1_IS	VS_IS	VBATSEN_IS
r			rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre

Field	Bits	Type	Description
VBATSEN_IS	0	rwhxre	**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
VS_IS	1	rwhxre	**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
MON1_IS	2	rwhxre	**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
MON2_IS	3	rwhxre	**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
MON3_IS	4	rwhxre	**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
MON4_IS	5	rwhxre	**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
MON5_IS	6	rwhxre	**ADC1 channel 6 interrupt status** Conversion of channel has finished

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **INACTIVE**: No channel 6 interrupt has occurred 1_B **ACTIVE**: Channel 6 interrupt has occurred
P2_1_IS	7	rwhxre	**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
P2_2_IS	8	rwhxre	**ADC1 channel 8 interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No channel 8 interrupt has occurred 1_B **ACTIVE**: Channel 8 interrupt has occurred
P2_3_IS	9	rwhxre	**ADC1 channel 9 interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No channel 9 interrupt has occurred 1_B **ACTIVE**: Channel 9 interrupt has occurred
P2_6_IS	10	rwhxre	**ADC1 channel 10 interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No channel 10 interrupt has occurred 1_B **ACTIVE**: Channel 10 interrupt has occurred
P2_7_IS	11	rwhxre	**ADC1 channel 11 interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No channel 11 interrupt has occurred 1_B **ACTIVE**: Channel 11 interrupt has occurred
P2_0_IS	12	rwhxre	**ADC1 channel 12 interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No channel 12 interrupt has occurred 1_B **ACTIVE**: Channel 12 interrupt has occurred
RES	15:13, 23:18	r	**Reserved** Always read as 0
EIM_IS	16	rwhxre	**Exceptional interrupt measurement (EIM) status** 0_B **INACTIVE**: No EIM occurred 1_B **ACTIVE**: EIM occurred
ESM_IS	17	rwhxre	**Exceptional sequence measurement (ESM) status** 0_B **INACTIVE**: No ESM has occurred 1_B **ACTIVE**: ESM occurred
DU1LO_IS	24	rwhxre	**ADC1 Differential Unit 1 (DU1) lower channel interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No DU lower channel Interrupt has occurred 1_B **ACTIVE**: DU lower channel interrupt has occurred
DU1UP_IS	25	rwhxre	**ADC1 differential unit 1 (DU1) upper channel interrupt status** Conversion of channel has finished

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **INACTIVE**: No DU upper Channel Interrupt has occurred 1_B **ACTIVE**: DU upper Channel Interrupt has occurred
DU2LO_IS	26	rwhxre	**ADC1 differential unit 2 (DU2) lower channel interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No DU lower channel interrupt has occurred 1_B **ACTIVE**: DU lower channel interrupt has occurred
DU2UP_IS	27	rwhxre	**ADC1 differential unit 2 (DU2) upper channel interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No DU upper channel interrupt has occurred 1_B **ACTIVE**: DU upper channel interrupt has occurred
DU3LO_IS	28	rwhxre	**ADC1 differential unit 3 (DU3) lower Channel interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No DU lower channel interrupt has occurred 1_B **ACTIVE**: DU lower channel interrupt has occurred
DU3UP_IS	29	rwhxre	**ADC1 differential unit 3 (DU3) upper channel interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No DU upper channel interrupt has occurred 1_B **ACTIVE**: DU upper channel interrupt has occurred
DU4LO_IS	30	rwhxre	**ADC1 differential unit 4 (DU4) lower channel interrupt status** Conversion of channel has finished 0_B **INACTIVE**: No DU lower channel interrupt has occurred 1_B **ACTIVE**: DU lower channel interrupt has occurred
DU4UP_IS	31	rwhxre	**ADC1 differential unit 4 (DU4) upper channel interrupt dtatus** Conversion of channel has finished 0_B **INACTIVE**: No DU upper channel interrupt has occurred 1_B **ACTIVE**: DU upper channel interrupt has occurred

23.14.63 ADC1 interrupt enable 1 register

ADC1_IRQEN_1 Offset address: 0068_H

ADC1 interrupt enable 1 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DU4UP_IEN	DU4LO_IEN	DU3UP_IEN	DU3LO_IEN	DU2UP_IEN	DU2LO_IEN	DU1UP_IEN	DU1LO_IEN	RES						ESM_IEN	EIM_IEN
rw	rw	rw	rw	rw	rw	rw	rw	r						rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			P2_0_IEN	P2_7_IEN	P2_6_IEN	P2_3_IEN	P2_2_IEN	P2_1_IEN	MON5_IEN	MON4_IEN	MON3_IEN	MON2_IEN	MON1_IEN	VS_IEN	VBATSEN_IEN
r			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
VBATSEN_IEN	0	rw	**ADC1 VBAT_SENSE interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
VS_IEN	1	rw	**ADC1 VS interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON1_IEN	2	rw	**ADC1 MON 1 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON2_IEN	3	rw	**ADC1 MON 2 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON3_IEN	4	rw	**ADC1 MON 3 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON4_IEN	5	rw	**ADC1 MON 4 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON5_IEN	6	rw	**ADC1 MON 5 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_1_IEN	7	rw	**ADC1 Port 2.1 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_2_IEN	8	rw	**ADC1 Port 2.2 interrupt enable**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_3_IEN	9	rw	**ADC1 Port 2.3 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_6_IEN	10	rw	**ADC1 Port 2.6 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_7_IEN	11	rw	**ADC1 Port 2.7 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_0_IEN	12	rw	**ADC1 Port 2.0 interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
RES	15:13, 23:18	r	**Reserved** Always read as 0
EIM_IEN	16	rw	**Exceptional interrupt measurement (EIM) interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
ESM_IEN	17	rw	**Exceptional sequence measurement (ESM) interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU1LO_IEN	24	rw	**Differential unit 1 lower interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU1UP_IEN	25	rw	**Differential unit 1 upper interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU2LO_IEN	26	rw	**Differential unit 2 lower interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU2UP_IEN	27	rw	**Differential unit 2 upper interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU3LO_IEN	28	rw	**Differential unit 3 lower interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU3UP_IEN	29	rw	**Differential unit 3 upper interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled

(table continues...)

(continued)

Field	Bits	Type	Description
DU4LO_IEN	30	rw	**Differential unit 4 lower interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
DU4UP_IEN	31	rw	**Differential unit 4 upper interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled

23.14.64 ADC1 interrupt status clear 1 register

ADC1_IRQCLR_1 Offset address: $006C_H$

ADC1 interrupt status clear 1 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DU4UP_ISC	DU4LO_ISC	DU3UP_ISC	DU3LO_ISC	DU2UP_ISC	DU2LO_ISC	DU1UP_ISC	DU1LO_ISC	RES						ESM_ISC	EIM_ISC
w	w	w	w	w	w	w	w	r						w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			P2_0_ISC	P2_7_ISC	P2_6_ISC	P2_3_ISC	P2_2_ISC	P2_1_ISC	MON5_ISC	MON4_ISC	MON3_ISC	MON2_ISC	MON1_ISC	VS_ISC	VBATSEN_ISC
r			w	w	w	w	w	w	w	w	w	w	w	w	w

Field	Bits	Type	Description
VBATSEN_ISC	0	w	**ADC1 VBAT_SENSE interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
VS_ISC	1	w	**ADC1 VS interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON1_ISC	2	w	**ADC1 MON 1 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON2_ISC	3	w	**ADC1 MON 2 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON3_ISC	4	w	**ADC1 MON 3 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON4_ISC	5	w	**ADC1 MON 4 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON5_ISC	6	w	**ADC1 MON 5 interrupt status clear** Interrupt status is cleared

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_1_ISC	7	w	**ADC1 Port 2.1 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_2_ISC	8	w	**ADC1 Port 2.2 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_3_ISC	9	w	**ADC1 Port 2.3 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_6_ISC	10	w	**ADC1 Port 2.6 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_7_ISC	11	w	**ADC1 Port 2.7 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_0_ISC	12	w	**ADC1 Port 2.0 interrupt status clear** Interrupt status is cleared 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
RES	15:13, 23:18	r	**Reserved** Always read as 0
EIM_ISC	16	w	**Exceptional interrupt measurement (EIM) status clear** 0_B **INACTIVE**: No EIM cleared 1_B **ACTIVE**: EIM cleared
ESM_ISC	17	w	**Exceptional sequence measurement (ESM) status clear** 0_B **INACTIVE**: No ESM has cleared 1_B **ACTIVE**: ESM cleared
DU1LO_ISC	24	w	**Differential unit 1 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
DU1UP_ISC	25	w	**Differential unit 1 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared

(table continues...)

(continued)

Field	Bits	Type	Description
DU2LO_ISC	26	w	**Differential unit 2 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
DU2UP_ISC	27	w	**Differential unit 2 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
DU3LO_ISC	28	w	**Differential unit 3 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
DU3UP_ISC	29	w	**Differential unit 3 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
DU4LO_ISC	30	w	**Differential unit 4 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
DU4UP_ISC	31	w	**Differential unit 4 lower interrupt status clear** 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared

23.14.65 ADC1 interrupt status 2 register

Hint: VBATSENSE supervision goes to NMI "supply prewarning", therefore bit 0 and 16 are "reserved" here.

ADC1_IRQS_2 Offset address: 0100_H

ADC1 interrupt status 2 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				P2_7_UP_IS	P2_6_UP_IS	P2_3_UP_IS	P2_2_UP_IS	P2_1_UP_IS	MON5_UP_IS	MON4_UP_IS	MON3_UP_IS	MON2_UP_IS	MON1_UP_IS	VS_UP_IS	RES
r				rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				P2_7_LO_IS	P2_6_LO_IS	P2_3_LO_IS	P2_2_LO_IS	P2_1_LO_IS	MON5_LO_IS	MON4_LO_IS	MON3_LO_IS	MON2_LO_IS	MON1_LO_IS	VS_LO_IS	RES
r				rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	rwhxre	r

Field	Bits	Type	Description
RES	0, 16:12, 31:28	r	**Reserved** Always read as 0
VS_LO_IS	1	rwhxre	**ADC1 VS lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON1_LO_IS	2	rwhxre	**ADC1 MON 1 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON2_LO_IS	3	rwhxre	**ADC1 MON 2 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON3_LO_IS	4	rwhxre	**ADC1 MON 3 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON4_LO_IS	5	rwhxre	**ADC1 MON 4 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON5_LO_IS	6	rwhxre	**ADC1 MON 5 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_1_LO_IS	7	rwhxre	**ADC1 port 2.1 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_2_LO_IS	8	rwhxre	**ADC1 port 2.2 lower threshold interrupt status**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_3_LO_IS	9	rwhxre	**ADC1 port 2.3 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_6_LO_IS	10	rwhxre	**ADC1 port 2.6 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_7_LO_IS	11	rwhxre	**ADC1 port 2.7 lower threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
VS_UP_IS	17	rwhxre	**ADC1 VS upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON1_UP_IS	18	rwhxre	**ADC1 MON 1 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON2_UP_IS	19	rwhxre	**ADC1 MON 2 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON3_UP_IS	20	rwhxre	**ADC1 MON 3 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON4_UP_IS	21	rwhxre	**ADC1 MON 4 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
MON5_UP_IS	22	rwhxre	**ADC1 MON 5 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_1_UP_IS	23	rwhxre	**ADC1 port 2.1 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_2_UP_IS	24	rwhxre	**ADC1 port 2.2 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_3_UP_IS	25	rwhxre	**ADC1 port 2.3 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred
P2_6_UP_IS	26	rwhxre	**ADC1 port 2.6 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ACTIVE**: Interrupt has occurred
P2_7_UP_IS	27	rwhxre	**ADC1 port 2.7 upper threshold interrupt status** 0_B **INACTIVE**: No interrupt has occurred 1_B **ACTIVE**: Interrupt has occurred

23.14.66 ADC1 status 2 register

Hint: VBATSENSE supervision goes to NMI "supply prewarning", therefore bit 0 and 16 are "reserved" here.

ADC1_STS_2 Offset address: 0104_H

ADC1 status 2 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				P2_7_UP_STS	P2_6_UP_STS	P2_3_UP_STS	P2_2_UP_STS	P2_1_UP_STS	MON5_UP_STS	MON4_UP_STS	MON3_UP_STS	MON2_UP_STS	MON1_UP_STS	VS_UP_STS	RES
r				rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				P2_7_LO_STS	P2_6_LO_STS	P2_3_LO_STS	P2_2_LO_STS	P2_1_LO_STS	MON5_LO_STS	MON4_LO_STS	MON3_LO_STS	MON2_LO_STS	MON1_LO_STS	VS_LO_STS	RES
r				rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	rc	r

Field	Bits	Type	Description
RES	0, 16:12, 31:28	r	**Reserved** Always read as 0
VS_LO_STS	1	rc	**ADC1 VS lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON1_LO_STS	2	rc	**ADC1 MON 1 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON2_LO_STS	3	rc	**ADC1 MON 2 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON3_LO_STS	4	rc	**ADC1 MON 3 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON4_LO_STS	5	rc	**ADC1 MON 4 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON5_LO_STS	6	rc	**ADC1 MON 5 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_1_LO_STS	7	rc	**ADC1 port 2.1 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_2_LO_STS	8	rc	**ADC1 port 2.2 lower threshold status**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_3_LO_STS	9	rc	**ADC1 port 2.3 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_6_LO_STS	10	rc	**ADC1 port 2.6 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_7_LO_STS	11	rc	**ADC1 port 2.7 lower threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
VS_UP_STS	17	rc	**ADC1 VS upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON1_UP_STS	18	rc	**ADC1 MON 1 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON2_UP_STS	19	rc	**ADC1 MON 2 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON3_UP_STS	20	rc	**ADC1 MON 3 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON4_UP_STS	21	rc	**ADC1 MON 4 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
MON5_UP_STS	22	rc	**ADC1 MON 5 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_1_UP_STS	23	rc	**ADC1 port 2.1 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_2_UP_STS	24	rc	**ADC1 port 2.2 upper threshold Status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_3_UP_STS	25	rc	**ADC1 port 2.3 upper threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_6_UP_STS	26	rc	**ADC1 port 2.6 upper threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ABOVE_LIMIT**: Upper threshold exceeded
P2_7_UP_STS	27	rc	**ADC1 port 2.7 upper threshold status** 0_B **BELOW_LIMIT**: Status below upper threshold 1_B **ABOVE_LIMIT**: Upper threshold exceeded

23.14.67 ADC1 interrupt status clear 2 register

ADC1_IRQCLR_2 Offset address: 0108_H

ADC1 interrupt status clear 2 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				P2_7_UP_ISC	P2_6_UP_ISC	P2_3_UP_ISC	P2_2_UP_ISC	P2_1_UP_ISC	MON5_UP_ISC	MON4_UP_ISC	MON3_UP_ISC	MON2_UP_ISC	MON1_UP_ISC	VS_UP_ISC	RES
r				w	w	w	w	w	w	w	w	w	w	w	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				P2_7_LO_ISC	P2_6_LO_ISC	P2_3_LO_ISC	P2_2_LO_ISC	P2_1_LO_ISC	MON5_LO_ISC	MON4_LO_ISC	MON3_LO_ISC	MON2_LO_ISC	MON1_LO_ISC	VS_LO_ISC	RES
r				w	w	w	w	w	w	w	w	w	w	w	r

Field	Bits	Type	Description
RES	0, 16:12, 31:28	r	**Reserved** Always read as 0
VS_LO_ISC	1	w	**ADC1 VS lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON1_LO_ISC	2	w	**ADC1 MON 1 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON2_LO_ISC	3	w	**ADC1 MON 2 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON3_LO_ISC	4	w	**ADC1 MON 3 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON4_LO_ISC	5	w	**ADC1 MON 4 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON5_LO_ISC	6	w	**ADC1 MON 5 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ACTIVE**: Interrupt status is cleared
P2_1_LO_ISC	7	w	**ADC1 port 2.1 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_2_LO_ISC	8	w	**ADC1 port 2.2 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_3_LO_ISC	9	w	**ADC1 port 2.3 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_6_LO_ISC	10	w	**ADC1 port 2.6 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_7_LO_ISC	11	w	**ADC1 port 2.7 lower threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
VS_UP_ISC	17	w	**ADC1 VS upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON1_UP_ISC	18	w	**ADC1 MON 1 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON2_UP_ISC	19	w	**ADC1 MON 2 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON3_UP_ISC	20	w	**ADC1 MON 3 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
MON4_UP_ISC	21	w	**ADC1 MON 4 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared

(table continues...)

(continued)

Field	Bits	Type	Description
MON5_UP_ISC	22	w	**ADC1 MON 5 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_1_UP_ISC	23	w	**ADC1 port 2.1 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_2_UP_ISC	24	w	**ADC1 port 2.2 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_3_UP_ISC	25	w	**ADC1 port 2.3 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_6_UP_ISC	26	w	**ADC1 port 2.6 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared
P2_7_UP_ISC	27	w	**ADC1 port 2.7 upper threshold interrupt status clear** Interrupt status is cleared. 0_B **INACTIVE**: Interrupt status is not cleared 1_B **ACTIVE**: Interrupt status is cleared

23.14.68 ADC1 interrupt enable 2 register

Hint: VBATSENSE supervision goes to NMI "supply prewarning", therefore bit 0 and 16 are "reserved" here.

ADC1_IRQEN_2 Offset address: $010C_H$

ADC1 interrupt enable 2 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES				P2_7_UP_IEN	P2_6_UP_IEN	P2_3_UP_IEN	P2_2_UP_IEN	P2_1_UP_IEN	MON5_UP_IEN	MON4_UP_IEN	MON3_UP_IEN	MON2_UP_IEN	MON1_UP_IEN	VS_UP_IEN	RES
r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				P2_7_LO_IEN	P2_6_LO_IEN	P2_3_LO_IEN	P2_2_LO_IEN	P2_1_LO_IEN	MON5_LO_IEN	MON4_LO_IEN	MON3_LO_IEN	MON2_LO_IEN	MON1_LO_IEN	VS_LO_IEN	RES
r				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r

Field	Bits	Type	Description
RES	0, 16:12, 31:28	r	**Reserved** Always read as 0
VS_LO_IEN	1	rw	**ADC1 VS lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON1_LO_IEN	2	rw	**ADC1 MON 1 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON2_LO_IEN	3	rw	**ADC1 MON 2 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON3_LO_IEN	4	rw	**ADC1 MON 3 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON4_LO_IEN	5	rw	**ADC1 MON 4 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON5_LO_IEN	6	rw	**ADC1 MON 5 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_1_LO_IEN	7	rw	**ADC1 port 2.1 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_2_LO_IEN	8	rw	**ADC1 port 2.2 lower threshold interrupt enable**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_3_LO_IEN	9	rw	**ADC1 port 2.3 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_6_LO_IEN	10	rw	**ADC1 port 2.6 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_7_LO_IEN	11	rw	**ADC1 port 2.7 lower threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
VS_UP_IEN	17	rw	**ADC1 VS upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON1_UP_IEN	18	rw	**ADC1 MON 1 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON2_UP_IEN	19	rw	**ADC1 MON 2 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON3_UP_IEN	20	rw	**ADC1 MON 3 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON4_UP_IEN	21	rw	**ADC1 MON 4 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
MON5_UP_IEN	22	rw	**ADC1 MON 5 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_1_UP_IEN	23	rw	**ADC1 port 2.1 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_2_UP_IEN	24	rw	**ADC1 port 2.2 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_3_UP_IEN	25	rw	**ADC1 port 2.3 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled
P2_6_UP_IEN	26	rw	**ADC1 port 2.6 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLED**: Interrupt enabled
P2_7_UP_IEN	27	rw	**ADC1 port 2.7 upper threshold interrupt enable** 0_B **DISABLED**: Interrupt disabled 1_B **ENABLED**: Interrupt enabled

23.14.69 ADC1 status 1 register

ADC1_STS_1 Offset address: 0124_H

ADC1 status 1 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DU4UP_STS	**DU4LO_STS**	**DU3UP_STS**	**DU3LO_STS**	**DU2UP_STS**	**DU2LO_STS**	**DU1UP_STS**	**DU1LO_STS**	**RES**							
rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	rwhxr	r							

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

Field	Bits	Type	Description
RES	23:0	r	**Reserved** Always read as 0
DU1LO_STS	24	rwhxr	**ADC1 differential unit 1 (DU1) lower channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU1UP_STS	25	rwhxr	**ADC1 differential unit 1 (DU1) upper channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred
DU2LO_STS	26	rwhxr	**ADC1 differential unit 2 (DU2) lower channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU2UP_STS	27	rwhxr	**ADC1 differential unit 2 (DU2) upper channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred
DU3LO_STS	28	rwhxr	**ADC1 differential unit 3 (DU3) lower channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU3UP_STS	29	rwhxr	**ADC1 differential unit 3 (DU3) upper channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred
DU4LO_STS	30	rwhxr	**ADC1 differential unit 4 (DU4) lower channel status**

(table continues...)

(continued)

Field	Bits	Type	Description
			Conversion of Channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU4UP_STS	31	rwhxr	**ADC1 differential unit 4 (DU4) upper channel status** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred

23.14.70 ADC1 status clear 1 register

ADC1_STSCLR_1 Offset address: 0128_H

ADC1 status clear 1 register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DU4UP_SC	**DU4LO_SC**	**DU3UP_SC**	**DU3LO_SC**	**DU2UP_SC**	**DU2LO_SC**	**DU1UP_SC**	**DU1LO_SC**	**RES**							
w	w	w	w	w	w	w	w	r							

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

Field	Bits	Type	Description
RES	23:0	r	**Reserved** Always read as 0
DU1LO_SC	24	w	**ADC1 differential unit 1 (DU1) lower channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU1UP_SC	25	w	**ADC1 differential unit 1 (DU1) upper channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred
DU2LO_SC	26	w	**ADC1 differential unit 2 (DU2) lower channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU2UP_SC	27	w	**ADC1 differential unit 2 (DU2) upper channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred
DU3LO_SC	28	w	**ADC1 differential unit 3 (DU3) lower channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU3UP_SC	29	w	**ADC1 differential unit 3 (DU3) upper channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred
DU4LO_SC	30	w	**ADC1 differential unit 4 (DU4) lower channel status clear**

(table continues...)

(continued)

Field	Bits	Type	Description
			Conversion of channel has finished. 0_B **INACTIVE**: No DU lower channel status has occurred 1_B **ACTIVE**: DU lower channel status has occurred
DU4UP_SC	31	w	**ADC1 differential unit 4 (DU4) upper channel status clear** Conversion of channel has finished. 0_B **INACTIVE**: No DU upper channel status has occurred 1_B **ACTIVE**: DU upper channel status has occurred

23.14.71 Measurement unit 1 - Differential unit input selection register

ADC1_DUIN_SEL

Measurement unit 1 - Differential unit input selection register

Offset address: $00FC_H$

RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES			DU4 RES_ NEG	RES			DU4 _EN	RES			DU3 RES_ NEG	RES			DU3 _EN
r			rc	r			rw	r			rc	r			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES			DU2 RES_ NEG	RES			DU2 _EN	RES			DU1 RES_ NEG	RES			DU1 _EN
r			rc	r			rw	r			rc	r			rw

Field	Bits	Type	Description
DU1_EN	0	rw	**Differential unit 1 enable** 0_B **DU1_DISABLE**: Differential unit 1 is disabled 1_B **DU1_ENABLE**: Differential unit 1 is enabled
RES	3:1, 7:5, 11:9, 15:13, 19:17, 23:21, 27:25, 31:29	r	**Reserved** Always read as 0
DU1RES_NEG	4	rc	**Differential unit 1 result negative** *Note: If the calculated result is negative.* 0_B **DU1_RESULT_POSITIVE**: Differential unit 1 result positive after calculation 1_B **DU1_RESULT_NEGATIVE**: Differential unit 1 result negative after calculation
DU2_EN	8	rw	**Differential unit 2 enable** 0_B **DU2_DISABLE**: Differential unit 2 is disabled 1_B **DU2_ENABLE**: Differential unit 2 is enabled
DU2RES_NEG	12	rc	**Differential unit 2 result negative** 0_B **DU2_RESULT_POSITIVE**: Differential unit 2 result positive after calculation 1_B **DU2_RESULT_NEGATIVE**: Differential unit 2 result negative after calculation
DU3_EN	16	rw	**Differential unit 3 enable**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **DU3_DISABLE**: Differential unit 3 is disabled 1_B **DU3_ENABLE**: Differential unit 3 is enabled
DU3RES_NEG	20	rc	**Differential unit 3 result negative** 0_B **DU3_RESULT_POSITIVE**: Differential unit 3 result positive after calculation 1_B **DU3_RESULT_NEGATIVE**: Differential unit 3 result negative after calculation
DU4_EN	24	rw	**Differential unit 4 enable** 0_B **DU4_DISABLE**: Differential unit 4 is disabled 1_B **DU4_ENABLE**: Differential unit 4 is enabled DU4 enable, differential unit 4 is enabled
DU4RES_NEG	28	rc	**Differential unit 4 result negative** 0_B **DU4_RESULT_POSITIVE**: Differential unit 4 result positive after calculation 1_B **DU4_RESULT_NEGATIVE**: Differential unit 4 result negative after calculation

24 High-voltage monitor input

24.1 Features

Features

- 4 or 5 (product variant dependent) high-voltage monitor inputs with VS/2 threshold voltage
- Wake capability for system stop mode and system sleep mode
- Edge sensitive wake-up feature configurable for transitions from low to high, high to low or both directions
- MON inputs can also be evaluated with ADC in active mode, using adjustable threshold values (see also Chapter 23)
- Selectable pull-up and pull-down current sources available

24.2 Introduction

This module is dedicated to monitor external voltage levels above or below a specified threshold. Each MONx pin can further be used to detect a wake-up event by detecting a level change by crossing the selected threshold. This applies to any power mode. Furthermore, each MONx pin can be sampled by the ADC as analog input.

24.2.1 Block diagram

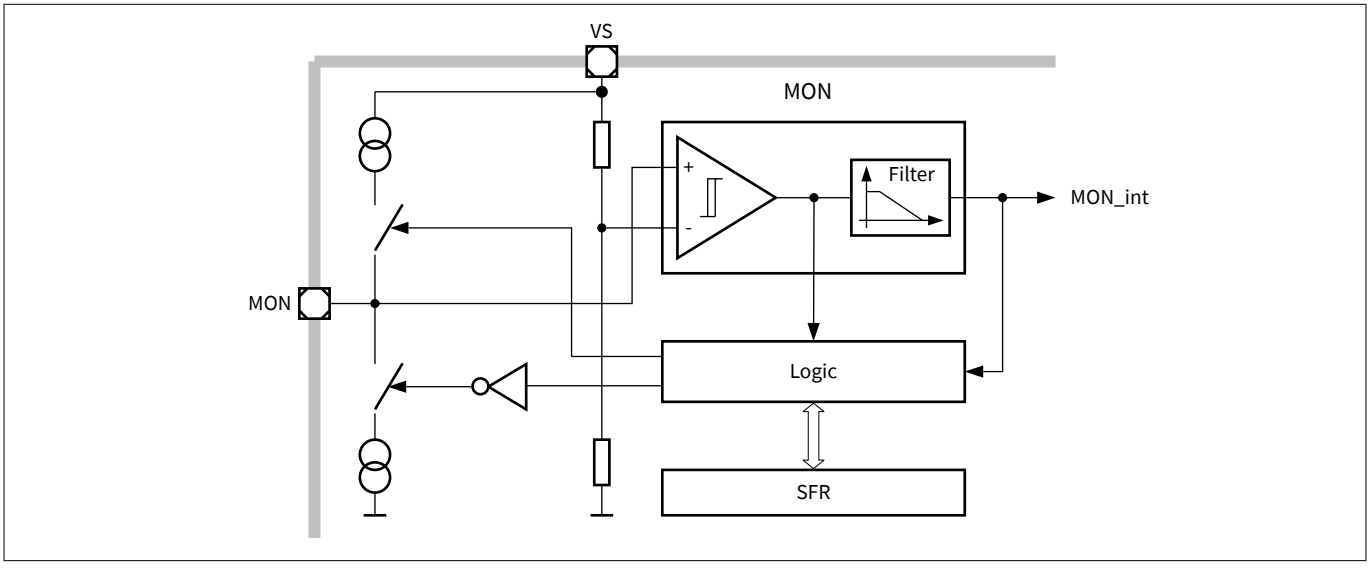


Figure 219 Monitoring input block diagram

24.2.2 Functional description

For a wake-up on a positive voltage transition, the MONx_RISE bit has to be configured. For a wake-up on a negative voltage transition, the corresponding bit MONx_FALL has to be set. This configuration can also be used for an edge detection in active mode.

As the system provides the functionality of cyclic sense, the MONx can be configured as a wake-up source for this mode. This is done by setting the bit MONx_CYC.

The MONx also includes an input circuit with pull-up (can be activated by MONx_PU bit) and pull-down (can be activated by MONx_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 MONx input of $V_{MON\_th} < V_{MONx} < V_S - 1\ V$ activates the pull-up current source. If the MONx voltage is between $1\ V < V_{MONx} < V_{MON\_th}$ the pull-down sink is activated, providing stable levels at the monitor 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 MONx_PU and MONx_PD bit to one at the same time.

Note: In case a monitoring input is deactivated by setting bit MONx_EN to 0, it can neither be used as a wake-up source nor to detect logic levels.

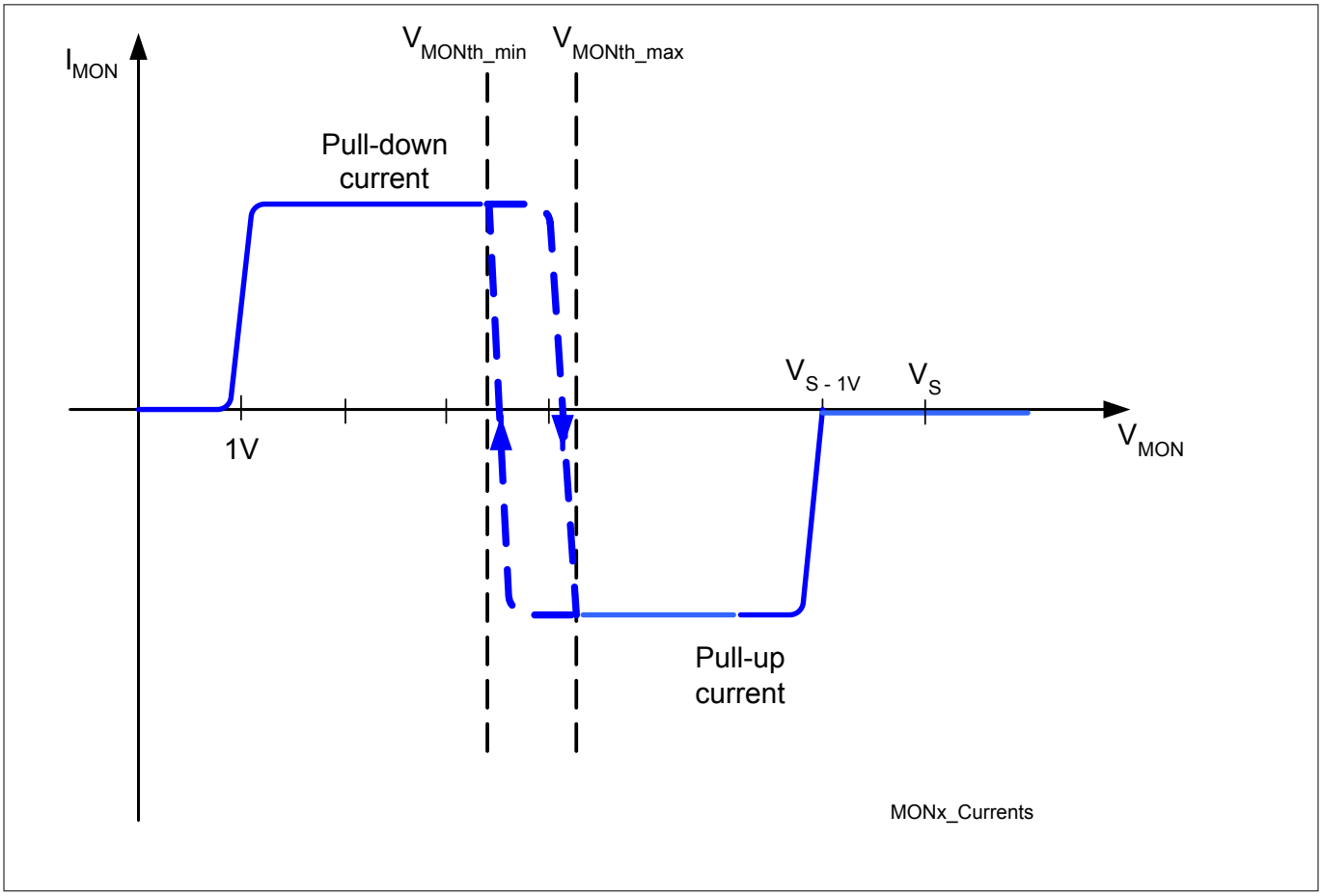


Figure 220 Module - HV_MON input characteristics for switchable pull current and static pull-down (on top) or pull-up

The following tables provide an overview of the configuration possibilities on the MON_INs via XSFR.

Table 178 includes all pull-up and pull-down setup scenarios which can be chosen for one MONx. Table 179 shows an overview of the available states of a MONx.

Table 178 Pull-up/down input current

MONx_PU	MONx_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 MONx circuit depending on level of input signal.

Table 179 MONx_EN MON mode definition

MONx_EN	Mode	Description
0	Disabled	Monitoring input is disabled (no wake-up possible!)
1	Normal/power saving mode	Monitoring input is active during device normal mode monitoring input automatically enters power saving mode in device sleep mode and stop mode

24.3 High-voltage monitor input registers

The monitor input registers PMU_MON_CNF1 and PMU_MON_CNF2 are part of the Power management unit (PMU) module. This is due to the fact that this circuit requires supply (VDD1V5_PD_A) and clock, (LP_CLK) during system wide sleep and stop modes.

Note: MON1-5 are device variant specific. In devices featuring only MON1-4 the PMU_MON_CFG2 register can be ignored. Writing to these register has no effect.

The registers are addressed bytewise.

24.3.1 Register overview - High-voltage monitor input registers (ascending offset address)

Table 180 Register overview - High-voltage monitor input registers (ascending offset address)

Short name	Long name	Offset address	Page number
PMU_MON_CNF1	Settings monitor 1-4 register	0034_H	103
PMU_MON_CNF2	Settings monitor 5 register	0038_H	107

25 High-side switch

25.1 Features

The high-side switch is optimized for driving resistive loads. Only small line inductance are allowed. Typical applications are single or multiple LEDs of a dashboard, switch illumination or other loads that require a high-side switch.

A cyclic switch activation during sleep mode or stop mode of the system is also available.

Functional features

- Multi-purpose high-side switch for resistive load connections (only small line inductances are allowed)
- Overcurrent limitation
- Overcurrent detection with thresholds: 25 mA, 50 mA, 100 mA, 150 mA and automatic shutdown
- Overtemperature detection and automatic shutdown
- Open load detection in on mode with open load current of max. 1.5 mA
- Interrupt signalling of overcurrent, overtemperature and open load condition
- Cyclic switch activation in sleep mode and stop mode with cyclic sense support and reduced driver capability: max. 40 mA
- PWM capability up to 25 kHz
- Internal connection to system-PWM generator (CCU6)
- Slew rate control for low EMI characteristic

Applications hints

- The voltage at HSx must not exceed the supply voltage by more than 0.3 V to prevent a reverse current from HSx to VS

25.2 Introduction

25.2.1 Block diagram

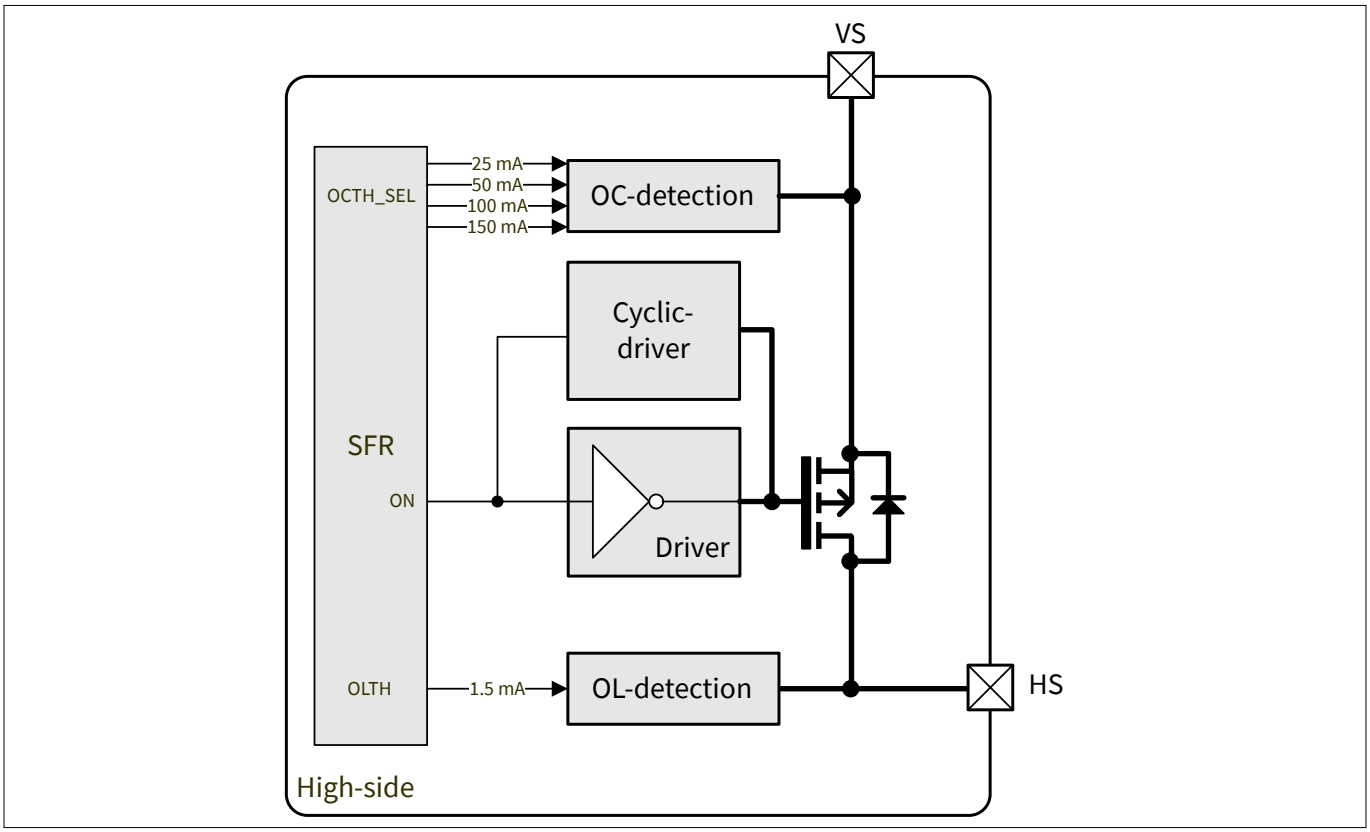


Figure 221 High-side module block diagram (incl. subblocks)

25.2.2 General

The high-side switch can generally be controlled in three different ways:

- In normal mode the output stage is fully controllable through the SFR registers HSx_CTRL. Protection functions as overcurrent, overtemperature and open load detection are available
- The PWM mode can also be enabled by a HSx_CTRL - SFR bit. The PWM configuration has to be done in the corresponding PWM module. All protection functions are also available in this mode. The maximum PWM frequency must not exceed 25 kHz (disabled slew rate control only)
- The high-side switch provides also the possibility of cyclic switch activation in all low power modes (sleep mode and stop mode). In this configuration it has limited functionality with limited current capability. Diagnostic functions are not available in this mode

25.3 Functional description

25.3.1 Normal operation

In normal operation mode (CPU normal mode, CPU slow down mode), the high-side switch provides functionalities and protection functions which are:

- Selectable slew rate control for improved EMI behavior
- Overcurrent detection with four different thresholds (min.): 25 mA, 50 mA, 100 mA and 150 mA
- Overtemperature protection, to protect the switch against overtemperature
- On-state open load detection with threshold lower than 1.5 mA typ

In device stop mode and device sleep mode the high-side driver is switched off and disabled. The user software does not need to take care about the proper power down sequence of this module. This is done by hardware.

In stop mode, the configuration of the driver is kept inside the corresponding sfrs. If the driver was switched on before entering stop mode, after a wake-up its status is restored automatically.

25.3.1.1 Slew rate configuration

The high-side switch provides two slew rate configuration possibilities:

- 10 V/µs (recommended up to 5 kHz PWM frequency)
- 30 V/µs (recommended above 10 kHz PWM frequency)

The intermediate range has to be evaluated in the application. The setting depends on load current demand. Power dissipation analysis is recommended.

The configuration can be done by flag HSx_SRCTL_SEL. The slew rate configuration is also taken for the PWM mode.

25.3.1.2 Overcurrent detection

To configure the proper overcurrent threshold the corresponding bits HSx_OC_SEL in the HSx_CTRL - SFR have to be set. If an overcurrent condition is present, the high-side switch will be automatically turned off. In parallel the flag HSx_OC_IS is set and the HSx_ON flag and HSx_PWM flag is cleared. To enable the high-side switch again, it is recommended to clear the HSx_OC_IS flag and then set the HSx_ON bit to reactivate the switch. Clearing only the HSx_OC_IS flag, would not turn the switch automatically on. If the overcurrent condition is still present, the switch will be disabled once again.

25.3.1.3 Overtemperature detection

If overtemperature condition appears, the switch will shutdown and the corresponding bit HSx_OT_STS is set. To re-enable the high-side switch, the same procedure as for the overcurrent condition has to be applied. Due to the fact that overtemperature condition is removed very slowly (device has to cool down) in comparison to the CPU time base, it is recommended to clear the status flag and to check if it is set again immediately after clearing, before trying to switch the driver on again.

25.3.1.4 ON-state open load detection

The high-side open load detection in ON State is mainly performed by the overcurrent detection and its fixed threshold of typ. 1.5 mA. If the current flowing through the output stage of the high-side switch falls below the value of typ. 1.5 mA, the corresponding status flag OL_STS is set. The open load detection has no influence on operation of the high-side switch.

The open load condition will cause an interrupt if enabled by the user.

25.3.1.5 Low-VS feature

The default behaviour of the high-side switch is the following:

The high-side is switched on using Bits HS1_EN and HS1_ON.

Note: This description describes HS1 only, but applies to HS2 as well accordingly.

In case of VS-undervoltage (detected by ADC2 ch0, at V_S < ~4.43 V), the high-side is switched off.

The high-side remains switched off, even when VS is increased above the threshold again, until the VS_UV interrupt and status bits are cleared in SCUPM_SYS_SUPPLY_IRQ_CLR (VS_UV_ISC and VS_UV_SC).

In case this switch-off of the high-side is not wanted, the high-side low-VS feature can be enabled:

The high-side is switched on using Bits HS1_EN, HS1_ON and HS1_CYC_ON_ACTIVE. (HS1_CYC_ON_ACTIVE should only be set together with HS1_ON). In that case, the high-side remains switched on down to V_S = 3 V.

25.3.2 PWM operation

In PWM mode the high-side switch has to be first enabled by the corresponding bits in the HSx_CTRL register. The related bits are described below. PWM_CHx in the following figure can be set in register HS_PWMSRCSEL.

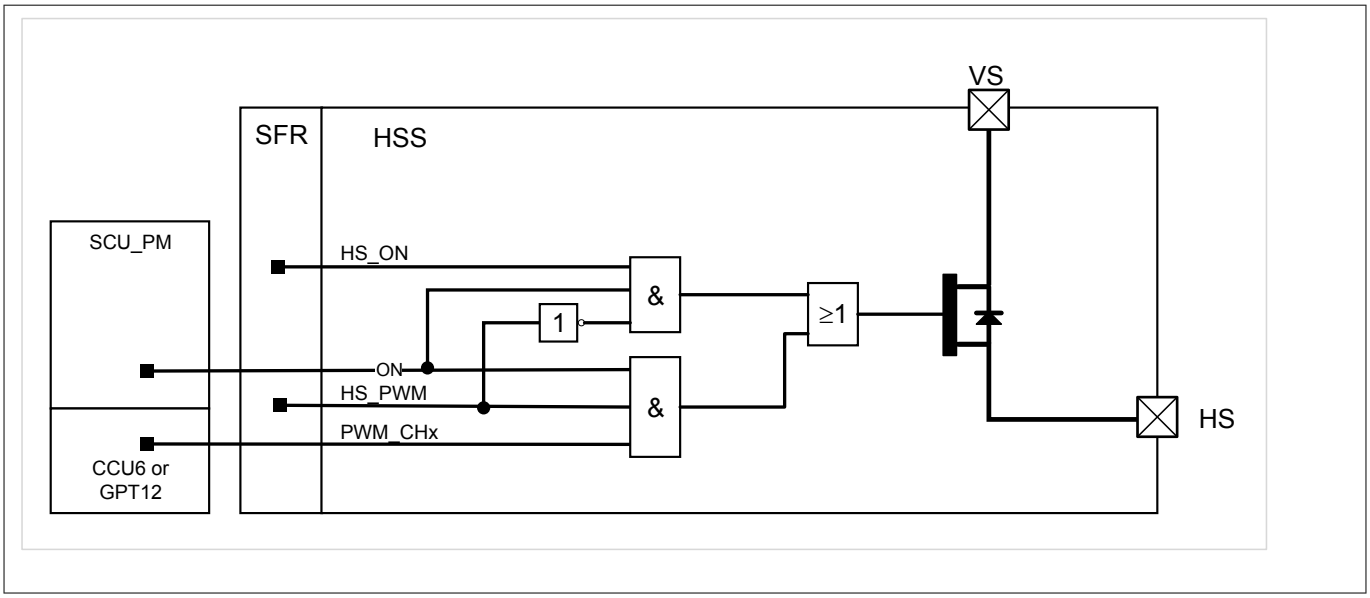


Figure 222 Combinatorial control of high-side switch in PWM mode

To avoid any output glitches on the HSx output, the HSx_PWM bit should be set first. After the function is enabled for PWM operation the corresponding PWM unit can be enabled.

For frequencies higher than 10 kHz, the slew rate setting has to be set to 30 V/µs. Otherwise the internal power dissipation of the switch might damage the device.

In PWM mode all protection functions are available.

25.3.3 Cyclic switching in low power mode

In the cyclic sense power-saving mode the high-side switch cyclically supplies an external switch arrangement for a short time, just long enough to detect the position of the switch. The configuration procedure to use the high-side switch for cyclic sense operation, is described in the chapter Power Management Unit.

25.4 High-side switch (HS) register definition

This chapter describes all necessary registers to control the high-side (HS) module and monitor its operation status.

Note: High-side switches are device variant specific. In devices featuring only one high-side switch HS2_XXX bitfields can be ignored. Writing to these bitfields has no effect.

The registers are addressed bytewise.

25.4.1 Register address space - HS

Table 181 Registers address space - HS

Module	Base address	End address	Note
HS	40024000_H	$40027FFF_H$	High-side switch registers

25.4.2 Register overview - HS (ascending offset address)

Table 182 Register overview - HS (ascending offset address)

Short name	Long name	Offset address	Page number
HS_CTRL	High-side driver control register	0004_H	890
HS_IRQS	High-side driver interrupt status register	0008_H	892
HS_IRQCLR	High-side driver interrupt status clear register	$000C_H$	894
HS_IRQEN	High-side driver interrupt enable register	0010_H	896
HS_HS1_TRIM	High-side driver 1 TRIM register	$001C_H$	897
HS_HS2_TRIM	High-side driver 2 TRIM register	0020_H	898
HS_PWMSRCSEL	High-side PWM source selection register	0024_H	899

25.4.3 High-side driver control register

HS_CTRL Offset address: 0004_H

High-side driver control register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES		**HS2_OC_SEL**		**RES**			**HS2_SRCTL_SEL**	**HS2_CYC_ON_ACTIVE**	**RES**			**HS2_OL_EN**	**HS2_ON**	**HS2_PWM**	**HS2_EN**
r		rw		r			rw	rwhir	r			rw	rwhrs	rwhir	rwhrs

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		**HS1_OC_SEL**		**RES**			**HS1_SRCTL_SEL**	**HS1_CYC_ON_ACTIVE**	**RES**			**HS1_OL_EN**	**HS1_ON**	**HS1_PWM**	**HS1_EN**
r		rw		r			rw	rwhir	r			rw	rwhrs	rwhir	rwhrs

Field	Bits	Type	Description
HS1_EN	0	rwhrs	**High-side 1 enable** 0_B **DISABLE**: HS circuit power off 1_B **ENABLE**: HS circuit power on
HS1_PWM	1	rwhir	**High-side 1 PWM enable** *Note: This flag has higher priority than HS1_ON.* 0_B **DISABLE**: Disables control by PWM input 1_B **ENABLE**: Enables control by PWM input
HS1_ON	2	rwhrs	**High-side 1 on** 0_B **OFF**: HS driver off 1_B **ON**: HS driver on
HS1_OL_EN	3	rw	**High-side 1 open load detection enable** 0_B **DISABLE**: Disable open load detection 1_B **ENABLE**: Enable open load detection
RES	6:4, 11:9, 15:14, 22:20, 27:25, 31:30	r	**Reserved** Always read as 0.
HS1_CYC_ON_ACTIVE	7	rwhir	**High-side 1 cyclic ON driver** 0_B **OFF**: Cyclic ON driver OFF 1_B **ON**: Cyclic ON driver ON

(table continues...)

(continued)

Field	Bits	Type	Description
HS1_SRCTL_SEL	8	rw	**High-side 1 slew rate control select** 0_B **SLEW_RATE_1**: Slew rate 10 V/µs is enabled 1_B **SLEW_RATE_2**: Slew rate 30 V/µs is enabled
HS1_OC_SEL	13:12	rw	**High-side 1 overcurrent threshold selection** 00_B **IOCTH0**: 25 mA min. 01_B **IOCTH1**: 50 mA min. 10_B **IOCTH2**: 100 mA min. 11_B **IOCTH3**: 150 mA min.
HS2_EN	16	rwhrs	**High-side 2 enable** 1) 0_B **DISABLE**: HS circuit power off 1_B **ENABLE**: HS circuit power on
HS2_PWM	17	rwhir	**High-side 2 PWM enable** *Note: 1) This flag has higher priority than HS2_ON.* 0_B **DISABLE**: Disables control by PWM input 1_B **ENABLE**: Enables control by PWM input
HS2_ON	18	rwhrs	**High-side 2 on** 1) 0_B **OFF**: HS driver off 1_B **ON**: HS driver on
HS2_OL_EN	19	rw	**High-side 2 open load detection enable** 1) 0_B **DISABLE**: Disable open load detection 1_B **ENABLE**: Enable open load detection
HS2_CYC_ON_ACTIVE	23	rwhir	**High-side 2 cyclic ON driver** 1) 0_B **OFF**: Cyclic ON driver OFF 1_B **ON**: Cyclic ON driver ON
HS2_SRCTL_SEL	24	rw	**High-side 2 slew rate control select** 1) 0_B **SLEW_RATE_1**: Slew rate 10 V/µs is enabled 1_B **SLEW_RATE_2**: Slew rate 30 V/µs is enabled
HS2_OC_SEL	29:28	rw	**High-side 2 overcurrent threshold selection** 1) 00_B **IOCTH0**: 25 mA min. 01_B **IOCTH1**: 50 mA min. 10_B **IOCTH2**: 100 mA min. 11_B **IOCTH3**: 150 mA min.

1) This flag is device variant specific. In devices featuring only one high-side switch writing to this bitfield has no effect.

25.4.4 High-side driver interrupt status register

HS_IRQS Offset address: 0008_H

High-side driver interrupt status register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**HS2_OL_STS**	**HS2_OT_STS**	**RES**					**HS2_OC_IS**	**HS2_OL_IS**	**HS2_OT_IS**	**RES**				
r	rwhxr	rwhxr	r					rwhxr	rwhxre	rwhxre	r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**HS1_OL_STS**	**HS1_OT_STS**	**RES**					**HS1_OC_IS**	**HS1_OL_IS**	**HS1_OT_IS**	**RES**				
r	rwhxr	rwhxr	r					rwhxr	rwhxre	rwhxre	r				

Field	Bits	Type	Description
RES	4:0, 12:8, 20:15, 28:24, 31	r	**Reserved** Always read as 0.
HS1_OT_IS	5	rwhxre	**High-side 1 overtemperature interrupt status** 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
HS1_OL_IS	6	rwhxre	**High-side 1 open load interrupt status** 0_B **NORMAL**: Normal load 1_B **OPEN_LOAD**: Open load detected, write sets status
HS1_OC_IS	7	rwhxr	**High-side 1 overcurrent interrupt status** 0_B **NO_OVERCURRENT**: No overcurrent condition occurred 1_B **OVERCURRENT**: Overcurrent occurred; switch is automatically shut down. Write sets status
HS1_OT_STS	13	rwhxr	**High-side 1 overtemperature status** 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
HS1_OL_STS	14	rwhxr	**High-side 1 open load interrupt status** 0_B **NO_OPEN_LOAD**: No open load condition occurred 1_B **OPEN_LOAD**: Open load occurred; switch is not automatically shut down. Write sets status
HS2_OT_IS	21	rwhxre	**High-side 2 overtemperature interrupt status** *1)* 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
HS2_OL_IS	22	rwhxre	**High-side 2 open load interrupt status** *1)* 0_B **NORMAL**: Normal load 1_B **OPEN_LOAD**: Open load detected, write sets status
HS2_OC_IS	23	rwhxr	**High-side 2 overcurrent interrupt status** *1)* 0_B **NO_OVERCURRENT**: No overcurrent condition occurred 1_B **OVERCURRENT**: Overcurrent occurred; switch is automatically shut down. Write sets status
HS2_OT_STS	29	rwhxr	**High-side 2 overtemperature status** *1)* 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
HS2_OL_STS	30	rwhxr	**High-side 2 open load interrupt status** *1)* 0_B **NO_OPEN_LOAD**: No open load condition occurred 1_B **OPEN_LOAD**: Open load occurred; switch is not automatically shut down. Write sets status

1) This flag is device variant specific. In devices featuring only one high-side switch writing to this bitfield has no effect.

25.4.5 High-side driver interrupt status clear register

HS_IRQCLR Offset address: $000C_H$

High-side driver interrupt status clear register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**HS2_OL_SC**	**HS2_OT_SC**	**RES**					**HS2_OC_ISC**	**HS2_OL_ISC**	**HS2_OT_ISC**	**RES**				
r	w	w	r					w	w	w	r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**HS1_OL_SC**	**HS1_OT_SC**	**RES**					**HS1_OC_ISC**	**HS1_OL_ISC**	**HS1_OT_ISC**	**RES**				
r	w	w	r					w	w	w	r				

Field	Bits	Type	Description
RES	4:0, 12:8, 20:15, 28:24, 31	r	**Reserved** Always read as 0.
HS1_OT_ISC	5	w	**High-side 1 overtemperature interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS1_OL_ISC	6	w	**High-side 1 open load interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS1_OC_ISC	7	w	**High-side 1 overcurrent interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS1_OT_SC	13	w	**High-side 1 overtemperature status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS1_OL_SC	14	w	**High-side 1 open load status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS2_OT_ISC	21	w	**High-side 2 overtemperature interrupt status clear** *1)* 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS2_OL_ISC	22	w	**High-side 2 open load interrupt status clear** *1)*

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS2_OC_ISC	23	w	**High-side 2 overcurrent interrupt status clear** *1)* 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS2_OT_SC	29	w	**High-side 2 overtemperature status clear** *1)* 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
HS2_OL_SC	30	w	**High-side 2 open load status clear** *1)* 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear

1) This flag is device variant specific. In devices featuring only one high-side switch writing to this bitfield has no effect.

25.4.6 High-side driver interrupt enable register

HS_IRQEN Offset address: 0010_H

High-side driver interrupt enable register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								HS2_OC_IEN	HS2_OL_IEN	HS2_OT_IEN	RES				
r								rw	rw	rw	r				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								HS1_OC_IEN	HS1_OL_IEN	HS1_OT_IEN	RES				
r								rw	rw	rw	r				

Field	Bits	Type	Description
RES	4:0, 20:8, 31:24	r	**Reserved** Always read as 0.
HS1_OT_IEN	5	rw	**High-side 1 overtemperature interrupt enable** 0_B **DISABLED**: Disabled 1_B **ENABLE**: Enable
HS1_OL_IEN	6	rw	**High-side 1 open load interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enable
HS1_OC_IEN	7	rw	**High-side 1 overcurrent interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enable
HS2_OT_IEN	21	rw	**High-side 2 overtemperature interrupt enable** *1)* 0_B **DISABLED**: Disabled 1_B **ENABLE**: Enable
HS2_OL_IEN	22	rw	**High-side 2 open load interrupt enable** *1)* 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enable
HS2_OC_IEN	23	rw	**High-side 2 overcurrent interrupt enable** *1)* 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enable

1) This flag is device variant specific. In devices featuring only one high-side switch writing to this bitfield has no effect.

25.4.7 High-side driver 1 TRIM register

HS_HS1_TRIM Offset address: $001C_H$

High-side driver 1 TRIM register RESET_TYPE_4 value: $0000\ 0000_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		RES				HS1_OC_OT_BTFILT_SEL		RES				RES		HS1_OL_BTFILT_SEL	
r		r				rw		r				r		rw	

Field	Bits	Type	Description
HS1_OL_BTFILT_SEL	1:0	rw	**Blanking time filter select for HS1 open load detection** 00_B **2_us**: 4 µs filter time 01_B **4_us**: 8 µs filter time 10_B **8_us**: 16 µs filter time 11_B **16_us**: 32 µs filter time
RES	3:2, 7:4, 13:10, 15:14, 18:16, 23:19, 27:24, 31:28	r	**Reserved** Always read as 0.
HS1_OC_OT_BTFILT_SEL	9:8	rw	**Blanking time filter select for HS1 overcurrent/overtemperature detection** 00_B **4_us**: 4 µs filter time 01_B **8_us**: 8 µs filter time 10_B **16_us**: 16 µs filter time 11_B **32_us**: 32 µs filter time

25.4.8 High-side driver 2 TRIM register

Note: This register is device variant specific. In devices featuring only one high-side switch writing to these bitfields has no effect.

HS_HS2_TRIM Offset address: 0020_H

High-side driver 2 TRIM register RESET_TYPE_4 value: $0000\ 0000_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES		RES				HS2_OC_OT_BTFILT_SEL		RES				RES		HS2_OL_BTFILT_SEL	
r		r				rw		r				r		rw	

Field	Bits	Type	Description
HS2_OL_BTFILT_SEL	1:0	rw	**Blanking time filter select for HS2 open load detection** 00_B **2_us**: 4 µs filter time 01_B **4_us**: 8 µs filter time 10_B **8_us**: 16 µs filter time 11_B **16_us**: 32 µs filter time
RES	3:2, 7:4, 13:10, 15:14, 18:16, 23:19, 27:24, 31:28	r	**Reserved** Always read as 0.
HS2_OC_OT_BTFILT_SEL	9:8	rw	**Blanking time/filter select for HS2 overcurrent/overtemperature detection** 00_B **4_us**: 4 µs filter time 01_B **8_us**: 8 µs filter time 10_B **16_us**: 16 µs filter time 11_B **32_us**: 32 µs filter time

25.4.9 High-side PWM source selection register

HS_PWMSRCSEL Offset address: 0024_H

High-side PWM source selection register RESET_TYPE_3 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										**HS1_SRC_SEL**			**HS2_SRC_SEL**		
r										rw			rw		

Field	Bits	Type	Description
HS2_SRC_SEL	2:0	rw	**HS2 PWM source selection** *Note:* [1] *Can be only written when HS_CTRL.HS2_PWM = 0.* 000_B **CC60**: PWM output of CCU6 (CC60) 001_B **CC61**: PWM output of CCU6 (CC61) 010_B **CC62**: PWM output of CCU6 (CC62) 011_B **COUT60**: PWM output of CCU6 (COUT60) 100_B **COUT61**: PWM output of CCU6 (COUT61) 101_B **COUT62**: PWM output of CCU6 (COUT62) 110_B **T3OUT**: PWM output of GPT12
HS1_SRC_SEL	5:3	rw	**HS1 PWM source selection** *Note: Can be only written when HS_CTRL.HS1_PWM = 0.* 000_B **CC60**: PWM output of CCU6 (CC60) 001_B **CC61**: PWM output of CCU6 (CC61) 010_B **CC62**: PWM output of CCU6 (CC62) 011_B **COUT60**: PWM output of CCU6 (COUT60) 100_B **COUT61**: PWM output of CCU6 (COUT61) 101_B **COUT62**: PWM output of CCU6 (COUT62) 110_B **T3OUT**: PWM output of GPT12
RES	31:6	r	**Reserved** Always read as 0.

1) This bitfield is device variant specific. In devices featuring only one high-side switch writing to this bitfield has no effect.

25.5 Interrupt generation – and status bit logic

The interrupt flags of the high-side module are having the following behaviour:

Overcurrent detection: The overcurrent detection interrupt flag is a level sensitive interrupt flag. This flag is set when the overcurrent condition occurs and stays persistent until the condition is removed.

Overtemperature detection: The overtemperature detection interrupt flag is a level sensitive interrupt flag. This flag is set when the overtemperature condition occurs, but can be cleared immediately. The overtemperature status of the overtemperature condition can then still be monitored in the dedicated status register, which is placed in the same interrupt status register.

Open load detection: The open load detection interrupt flag is a level sensitive interrupt flag. This flag is set when the open condition occurs, but can be cleared immediately. The open load status of the open load condition can then still be monitored in the dedicated status register, which is placed in the same interrupt status register.

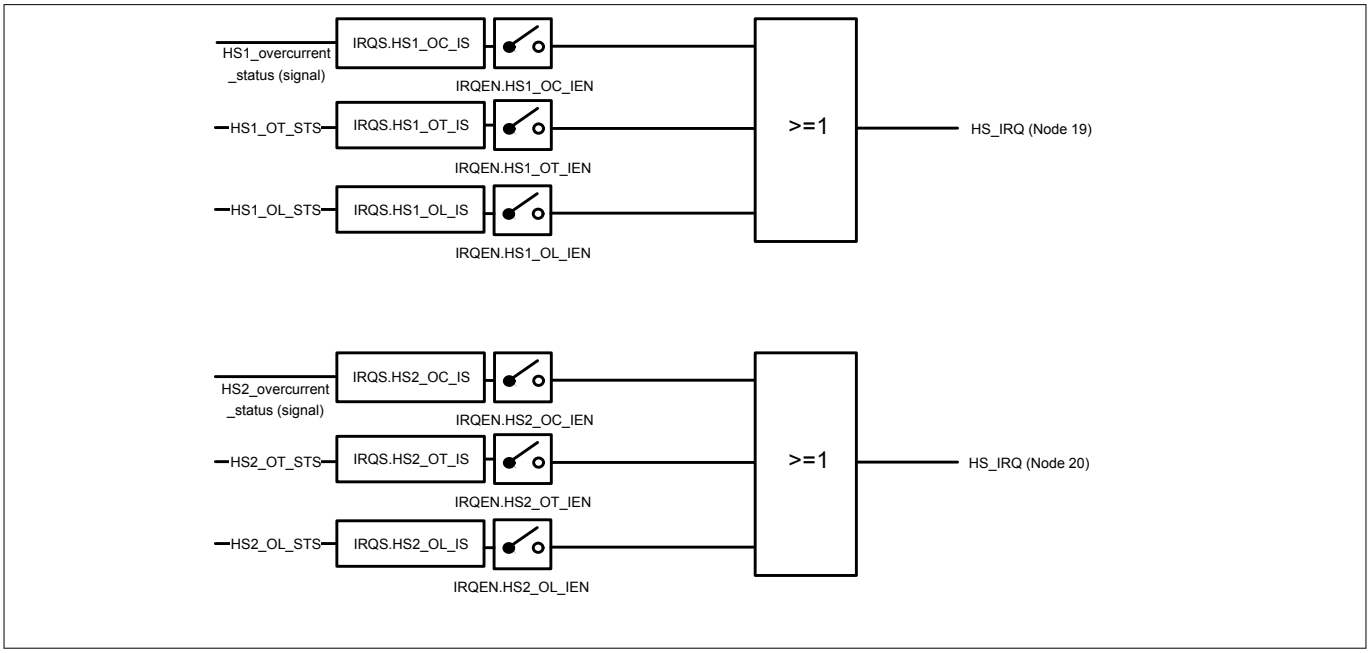


Figure 223 High-side 1/2 switch interrupt generation

25.6 Application information

If the high-side module is used as off board pin the following external circuitry is mandatory:

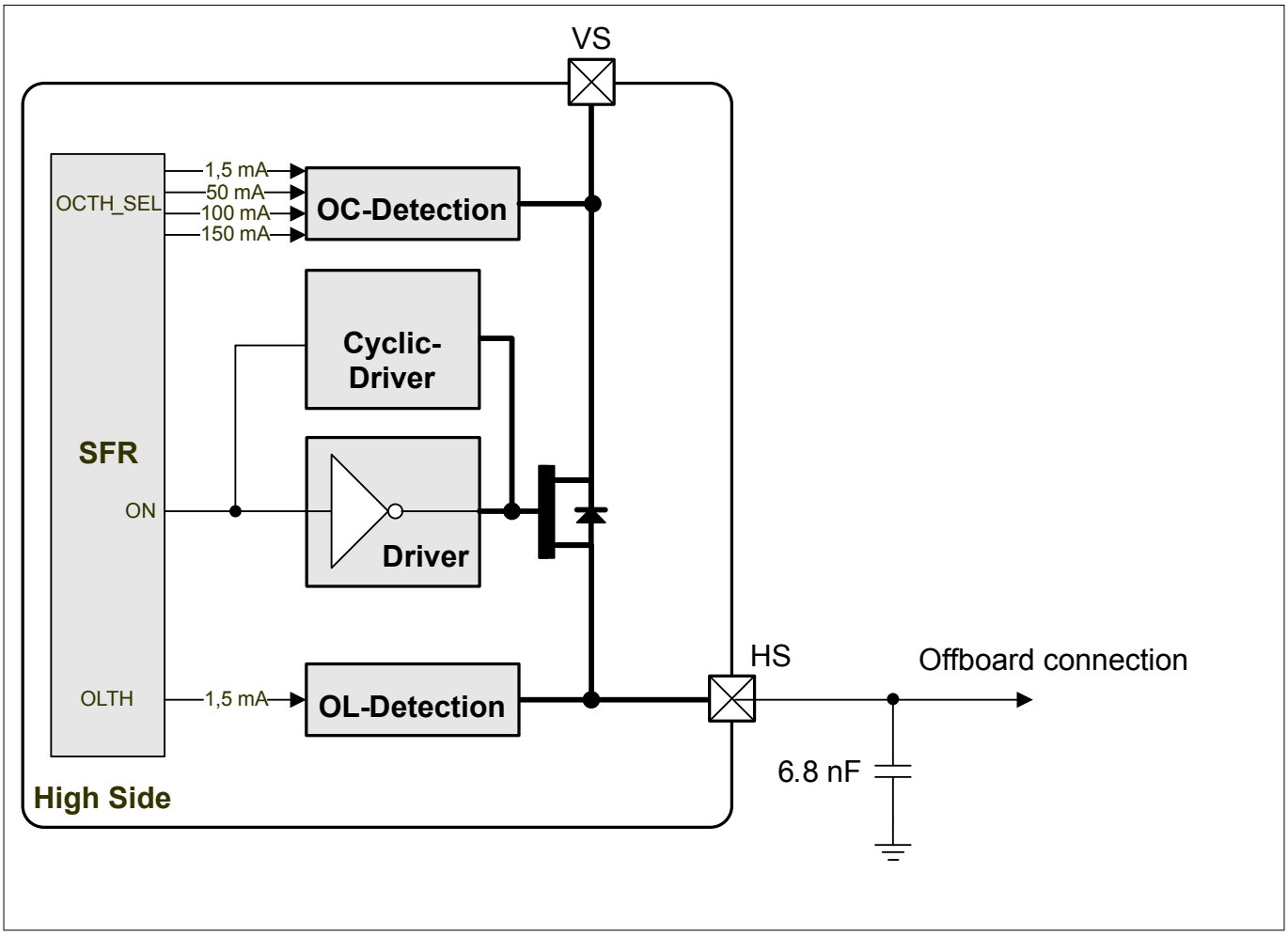


Figure 224 Circuitry mandatory for use as off board pin

If the high-side module is used as off board pin a 6.8 nF is needed as buffer capacitor.

26 Low-side switch

26.1 Features

The general purpose low-side switch is optimized to control an on-board relay. The low-side switch provides embedded protection functions including overcurrent and overtemperature detection. The module is designed for on-board connections.

Measures for standard ESD (HBM) and EMC robustness are implemented.

Functional features

- Multipurpose low-side switch optimized for driving relays:
  - Simple relay driver
  - PWM relay driver
- Integrated clamping for usage as a simple relay driver
- Overcurrent detection and automatic shutdown
- Overtemperature detection and automatic shutdown
- Interrupt signalling of overcurrent and overtemperature condition
- Open load detection with interrupt signaling
- PWM capability up to 25 kHz (for inductive loads with external clamping circuitry only)
- Selectable PWM source: Dedicated CCU6 channels
- Current drive capability up to min. 270 mA

Applications hints

- It is not recommended to use the switch in PWM mode without external free wheeling diode. See Chapter 26.3.2.1.

26.2 Introduction

26.2.1 Block diagram

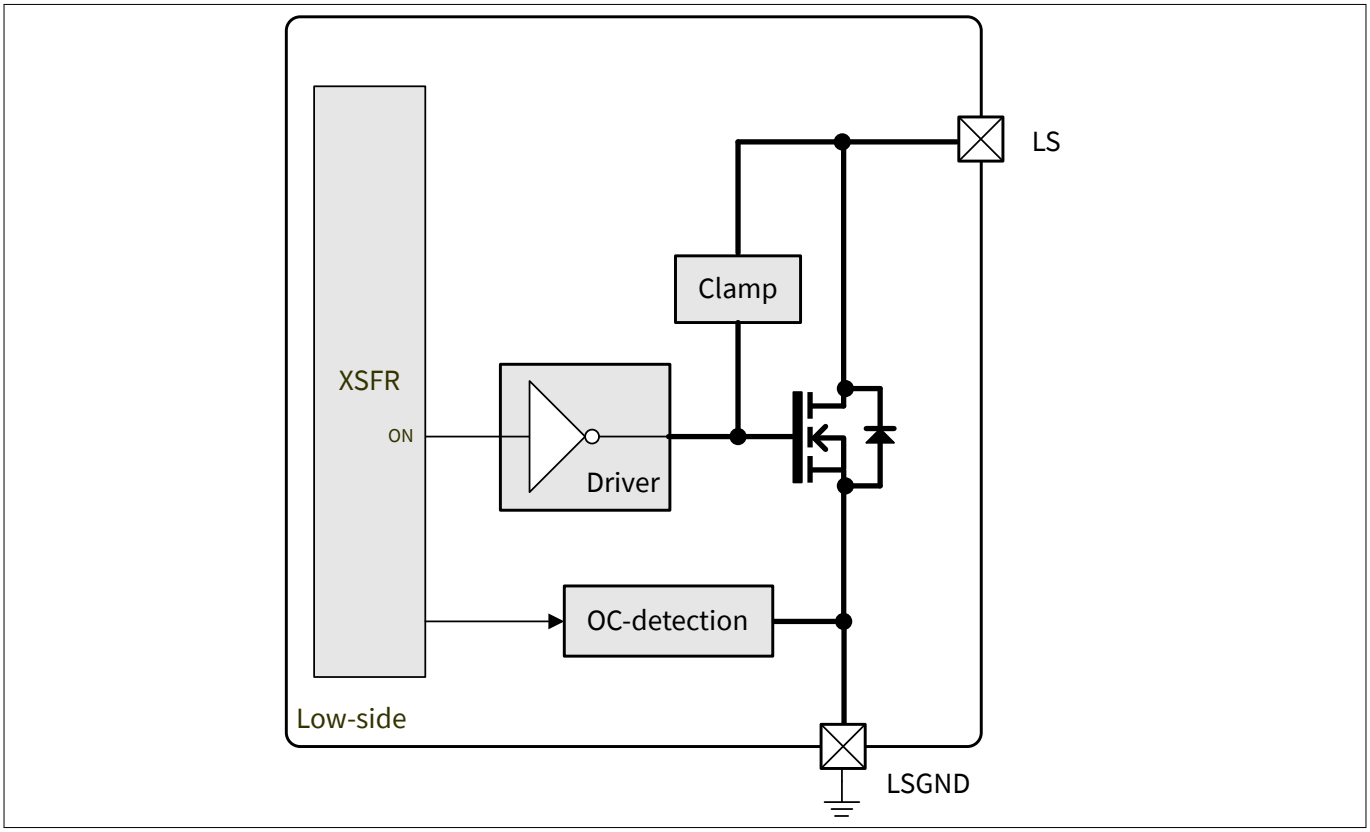


Figure 225 Low-side module block diagram

26.2.2 General

The low-side switches can be generally controlled in two different ways:

- In normal mode the output stage is fully controllable through the SFR registers LSx_CTRL. Protection functions as overcurrent and overtemperature are available
- The PWM mode can also be enabled by a LSx_CTRL - SFR bit. The PWM configuration has to be done in the corresponding PWM module (CCU6). All protection functions are also available in this mode. The maximum PWM frequency must not exceed 25 kHz (fast slew rate only)

26.3 Functional description

26.3.1 Normal operation

In normal operation mode (CPU normal mode, CPU slow-down mode) the low-side switch provides functionalities and protection functions which are:

- Selectable slew rate control for improved EMI behavior
- Overcurrent detection with threshold (min.) 270 mA
- Overtemperature protection, to protect the switch against overtemperature

In device stop mode and device sleep mode the low-side driver is switched off and disabled. The user software does not need to take care about the proper power-down sequence of this module. This is done by hardware.

In stop mode the configuration of the driver is kept inside the corresponding sfrs. If the driver was switched on before entering stop mode, after a wake up its status is restored automatically.

26.3.1.1 Slew rate configuration

The low-side switch provides two slew rate configuration possibilities:

- 10 V/µs (up to 5 kHz PWM frequency)
- 30 V/µs (above 5 kHz PWM frequency)

The configuration can be done by flag LSx_SRCTL_SEL. The slew rate configuration is also taken for the PWM mode.

26.3.1.2 Overcurrent detection

If an overcurrent condition is present, the low-side switches will be automatically turned off. In parallel the flag LSx_OC_IS is set and the LSx_ON flag is cleared. To enable the low-side switch again, it is recommended to clear the LSx_OC_IS flag and then set the LSx_ON bit to reactivate the switch. Clearing only the LSx_OC_IS flag, would not turn the switch automatically on. If the overcurrent condition is still present, the switch will be disabled once again.

26.3.1.3 Overtemperature detection

If overtemperature condition appears, the switch will shutdown and the corresponding bit LSx_OT_STS is set. To re-enable the low-side switches, the same procedure as for the overcurrent condition has to be applied. Due to the fact that overtemperature condition is removed very slowly (device has to cool down) in comparison to the CPU time base, it is recommended to clear the status flag and to check if it is set again immediately after clearing, before trying to switch the driver on again.

26.3.2 Operation of low-side switch in PWM mode

The low-side switch can also be operated in PWM mode. To enable the PWM mode of the low-side switch, the corresponding bits LSx_PWM and LSx_ON in the control register LSx_CTRL have to be set. The implemented combinatorial logic is shown in the next figure. PWM_CHx in the following figure can be set in register LS_PWMSRCSEL.

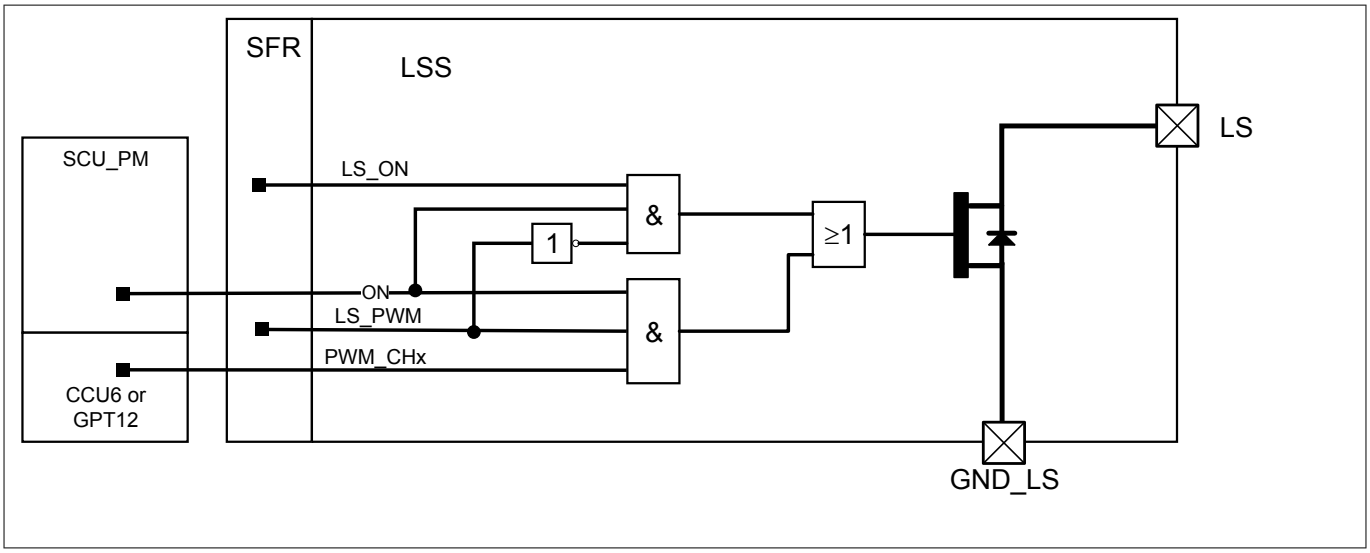


Figure 226 Module PWM usage of low-side switch

26.3.2.1 Application requirement for low-side switch in PWM mode

The low-side switch is not designed to handle the amount of energy which is generated by switching an inductive load in PWM mode. Therefore an external freewheeling diode is required to absorb the generated energy. The picture below shows the possible application diagram for this case.

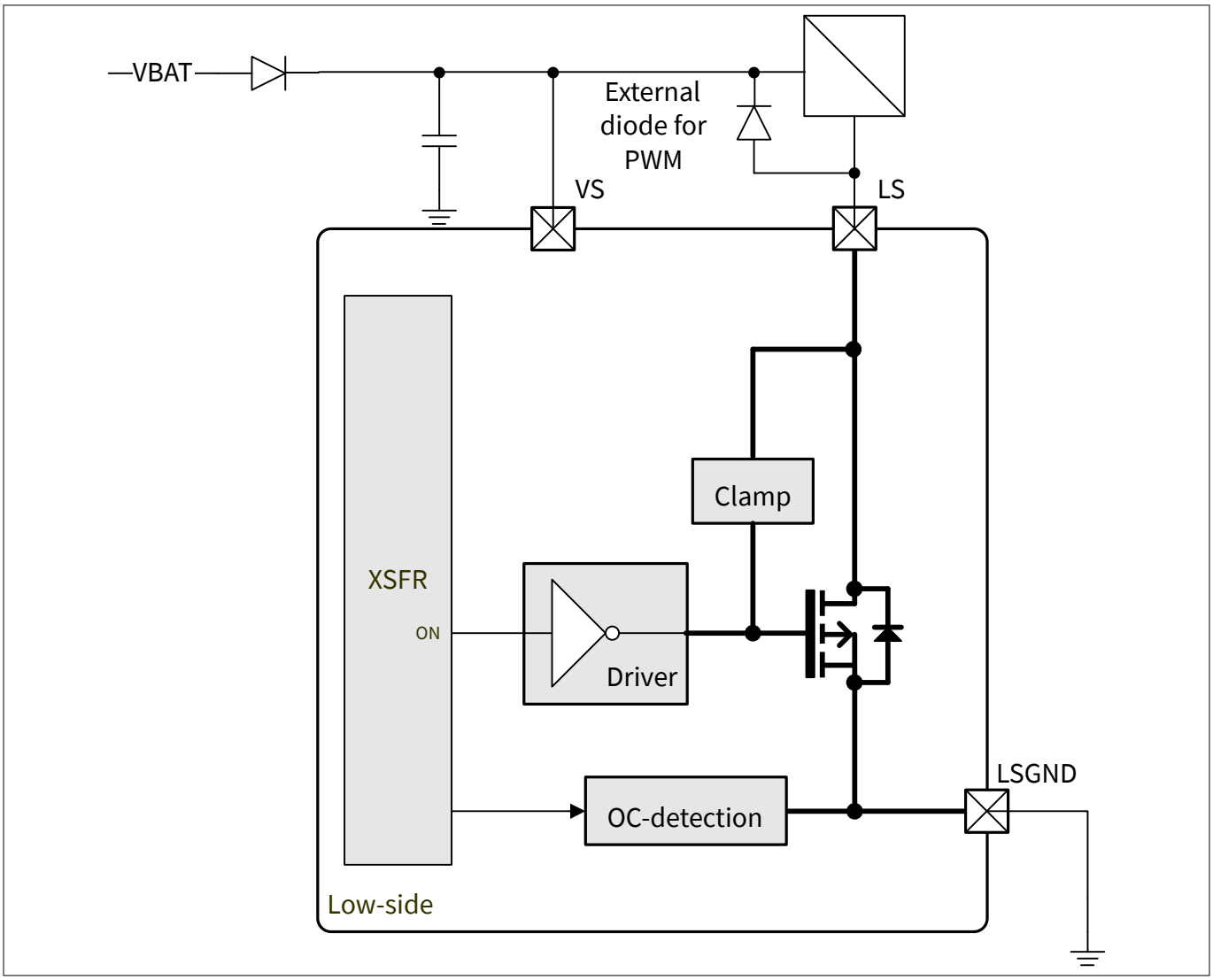


Figure 227 Module block diagram (with interconnects and external components)

26.4 Low-side switch (LS) register definition

This chapter describes all necessary registers to control the low-side (LS) module and monitor its operation status.

The registers are addressed bytewise.

26.4.1 Register address space - LS

Table 183 Registers address space - LS

Module	Base address	End address	Note
LS	$4001C000_H$	$4001FFFF_H$	Low-side switch registers

26.4.2 Register overview - LS (ascending offset address)

Table 184 Register overview - LS (ascending offset address)

Short name	Long name	Offset address	Page number
LS_CTRL	Low-side driver control register	0004_H	908
LS_IRQS	Low-side driver interrupt status register	0008_H	910
LS_IRQCLR	Low-side driver interrupt status clear register	$000C_H$	912
LS_IRQEN	Low-side driver interrupt enable register	0010_H	914
LS_LS1_TRIM	Low-side 1 reference current trimming register	0018_H	916
LS_PWMSRCSEL	Low-side PWM source selection register	$001C_H$	917
LS_LS2_TRIM	Low-side 2 reference current trimming register	0020_H	918

26.4.3 Low-side driver control register

LS_CTRL Offset address: 0004_H

Low-side driver control register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES							LS2_SRCTL_SEL	RES				LS2_OL_EN	LS2_ON	LS2_PWM	LS2_EN
r							rw	r				rw	rwhir	rwhir	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES							LS1_SRCTL_SEL	RES				LS1_OL_EN	LS1_ON	LS1_PWM	LS1_EN
r							rw	r				rw	rwhir	rwhir	rw

Field	Bits	Type	Description
LS1_EN	0	rw	**Low-side switch 1 enable** 0_B **DISABLE**: Disables LS1 1_B **ENABLE**: Enables LS1
LS1_PWM	1	rwhir	**Low-side switch 1 PWM enable** *Note: This flag has higher priority than LS1_ON.* 0_B **DISABLE**: Normal mode controlled by LS1_ON 1_B **ENABLE**: Enables LS1 for PWM mode
LS1_ON	2	rwhir	**Low-side switch 1 on/off** 0_B **OFF**: Switches LS1 off 1_B **ON**: Turns LS1 on
LS1_OL_EN	3	rw	**Open load detection enable** 0_B **DISABLE**: Open load detection 1_B **ENABLE**: Open load detection
RES	7:4, 15:9, 23:20, 31:25	r	**Reserved** Always read as 0.
LS1_SRCTL_SEL	8	rw	**Low-side switch 1 slew rate selection** 0_B **SLOW**: Slow slew rate is selected 1_B **FAST**: Fast slew rate is selected
LS2_EN	16	rw	**Low-side switch 2 enable** 0_B **DISABLE**: Disables LS2 1_B **ENABLE**: Enables LS2
LS2_PWM	17	rwhir	**Low-side switch 2 PWM enable**

(table continues...)

(continued)

Field	Bits	Type	Description
			Note: This flag has higher priority than LS2_ON. 0_B **DISABLE**: Normal mode controlled by LS2_ON 1_B **ENABLE**: Enables LS2 for PWM mode
LS2_ON	18	rwhir	**Low-Side switch 2 on/off** 0_B **OFF**: Switches LS2 off 1_B **ON**: Turns LS2 on
LS2_OL_EN	19	rw	**Open load detection enable** 0_B **DISABLE**: Open load detection 1_B **ENABLE**: Open load detection
LS2_SRCTL_SEL	24	rw	**Low-side switch 2 slew rate selection** 0_B **SLOW**: Slow slew rate is selected 1_B **FAST**: Fast slew rate is selected

26.4.4 Low-side driver interrupt status register

LS_IRQS Offset address: 0008_H

Low-side driver interrupt status register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**LS2_OL_STS**	**LS2_OT_STS**	**LS2_OT_PREWARN_STS**	**RES**				**LS2_OC_IS**	**LS2_OL_IS**	**LS2_OT_IS**	**LS2_OT_PREWARN_IS**	**RES**			
r	rwhxr	rwhxr	rwhxr	r				rwhxr	rwhxre	rwhxre	rwhxre	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**LS1_OL_STS**	**LS1_OT_STS**	**LS1_OT_PREWARN_STS**	**RES**				**LS1_OC_IS**	**LS1_OL_IS**	**LS1_OT_IS**	**LS1_OT_PREWARN_IS**	**RES**			
r	rwhxr	rwhxr	rwhxr	r				rwhxr	rwhxre	rwhxre	rwhxre	r			

Field	Bits	Type	Description
RES	3:0, 11:8, 19:15, 27:24, 31	r	**Reserved** Always read as 0.
LS1_OT_PREWARN_IS	4	rwhxre	**Low-Side 1 overtemperature prewarning interrupt status** 0_B **NO_OVERTEMPERATURE_PREWARN**: No overtemperature prewarn occurred. 1_B **OVERTEMPERATURE_PREWARN**: Overtemperature prewarn occurred. Write sets status
LS1_OT_IS	5	rwhxre	**Low-Side 1 overtemperature interrupt status** 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
LS1_OL_IS	6	rwhxre	**Low-Side 1 open load interrupt status** 0_B **NO_OPEN_LOAD**: No open load condition occurred 1_B **OPEN_LOAD**: Open load occurred; switch is not automatically shut down. Write sets status
LS1_OC_IS	7	rwhxr	**Low-Side 1 overcurrent interrupt status** 0_B **NO_OVERCURRENT**: No overcurrent condition occurred 1_B **OVERCURRENT**: Overcurrent occurred; switch is automatically shut down. Write sets status
LS1_OT_PREWARN_STS	12	rwhxr	**Low-Side 1 overtemperature prewarning status**

(table continues...)

(continued)

Field	Bits	Type	Description
			0_B **NO_OVERTEMPERATURE_PREWARN**: No overtemperature prewarn occurred 1_B **OVERTEMPERATURE**: Overtemperature prewarn occurred; Write sets status
LS1_OT_STS	13	rwhxr	**Low-Side 1 overtemperature status** 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
LS1_OL_STS	14	rwhxr	**Low-Side 1 open load status** 0_B **NO_OPEN_LOAD**: No open load condition occurred 1_B **OPEN_LOAD**: Open load occurred; switch is not automatically shut down. Write sets status
LS2_OT_PREWARN_IS	20	rwhxre	**Low-Side 2 overtemperature prewarning interrupt status** 0_B **NO_OVERTEMPERATURE_PREWARN**: No overtemperature prewarn occurred 1_B **OVERTEMPERATURE_PREWARN**: Overtemperature prewarn occurred. Write sets status
LS2_OT_IS	21	rwhxre	**Low-Side 2 overtemperature interrupt status** 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
LS2_OL_IS	22	rwhxre	**Low-Side 2 open load interrupt status** 0_B **NO_OPEN_LOAD**: No open load condition occurred 1_B **OPEN_LOAD**: Open load occurred; switch is not automatically shut down. Write sets status
LS2_OC_IS	23	rwhxr	**Low-Side 2 overcurrent interrupt status** 0_B **NO_OVERCURRENT**: No overcurrent condition occurred 1_B **OVERCURRENT**: Overcurrent occurred; switch is automatically shut down. Write sets status
LS2_OT_PREWARN_STS	28	rwhxr	**Low-Side 2 overtemperature prewarning status** 0_B **NO_OVERTEMPERATURE_PREWARN**: No overtemperature prewarn occurred 1_B **OVERTEMPERATURE_PREWARN**: Overtemperature prewarn occurred. Write sets status
LS2_OT_STS	29	rwhxr	**Low-Side 2 overtemperature status** 0_B **NO_OVERTEMPERATURE**: No overtemperature occurred 1_B **OVERTEMPERATURE**: Overtemperature occurred; switch is automatically shut down. Write sets status
LS2_OL_STS	30	rwhxr	**Low-side 2 open load status** 0_B **NO_OPEN_LOAD**: No open load condition occurred 1_B **OPEN_LOAD**: Open load occurred; switch is not automatically shut down. Write sets status

26.4.5 Low-side driver interrupt status clear register

LS_IRQCLR Offset address: $000C_H$

Low-side driver interrupt status clear register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES	**LS2_OL_SC**	**LS2_OT_SC**	**LS2_OT_PREWARN_SC**	**RES**				**LS2_OC_ISC**	**LS2_OL_ISC**	**LS2_OT_ISC**	**LS2_OT_PREWARN_ISC**	**RES**			
r	w	w	w	r				w	w	w	w	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES	**LS1_OL_SC**	**LS1_OT_SC**	**LS1_OT_PREWARN_SC**	**RES**				**LS1_OC_ISC**	**LS1_OL_ISC**	**LS1_OT_ISC**	**LS1_OT_PREWARN_ISC**	**RES**			
r	w	w	w	r				w	w	w	w	r			

Field	Bits	Type	Description
RES	3:0, 11:8, 19:15, 27:24, 31	r	**Reserved** Always read as 0.
LS1_OT_PREWARN_ISC	4	w	**Low-side 1 overtemperature prewarn interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS1_OT_ISC	5	w	**Low-side 1 overtemperature interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS1_OL_ISC	6	w	**Low-side 1 open load interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS1_OC_ISC	7	w	**Low-side 1 overcurrent interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS1_OT_PREWARN_SC	12	w	**Low-side 1 overtemperature prewarn status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS1_OT_SC	13	w	**Low-side 1 overtemperature status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear

(table continues...)

(continued)

Field	Bits	Type	Description
LS1_OL_SC	14	w	**Low-side 1 open load status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OT_PREWARN_ISC	20	w	**Low-side 2 overtemperature prewarn interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OT_ISC	21	w	**Low-side 2 overtemperature interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OL_ISC	22	w	**Low-side 2 open load interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OC_ISC	23	w	**Low-side 2 overcurrent interrupt status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OT_PREWARN_SC	28	w	**Low-side 2 overtemperature prewarn status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OT_SC	29	w	**Low-side switch 2 overtemperature status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear
LS2_OL_SC	30	w	**Low-side 2 open load status clear** 0_B **NO_CLEAR**: No clear 1_B **CLEAR**: Clear

26.4.6 Low-side driver interrupt enable register

LS_IRQEN Offset address: 0010_H

Low-side driver interrupt enable register RESET_TYPE_3 value: $0000\ 0000_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES								LS2_OC_IEN	LS2_OL_IEN	LS2_OT_IEN	LS2_OT_PREWARN_IEN	RES			
r								rw	rw	rw	rw	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES								LS1_OC_IEN	LS1_OL_IEN	LS1_OT_IEN	LS1_OT_PREWARN_IEN	RES			
r								rw	rw	rw	rw	r			

Field	Bits	Type	Description
RES	3:0, 19:8, 31:24	r	**Reserved** Always read as 0.
LS1_OT_PREWARN_IEN	4	rw	**Low-side 1 overtemperature prewarn interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
LS1_OT_IEN	5	rw	**Low-side 1 overtemperature interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
LS1_OL_IEN	6	rw	**Low-side 1 open load interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
LS1_OC_IEN	7	rw	**Low-side 1 overcurrent interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
LS2_OT_PREWARN_IEN	20	rw	**Low-side 2 overtemperature prewarn interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
LS2_OT_IEN	21	rw	**Low-side 2 overtemperature interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled
LS2_OL_IEN	22	rw	**Low-side 2 open load interrupt enable** 0_B **DISABLE**: Disabled

(table continues...)

(continued)

Field	Bits	Type	Description
			1_B **ENABLE**: Enabled
LS2_OC_IEN	23	rw	**Low-side 2 overcurrent interrupt enable** 0_B **DISABLE**: Disabled 1_B **ENABLE**: Enabled

26.4.7 Low-side 1 reference current trimming register

LS_LS1_TRIM Offset address: 0018_H

Low-side 1 reference current trimming register RESET_TYPE_4 value: $0000\ 0000_H$

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

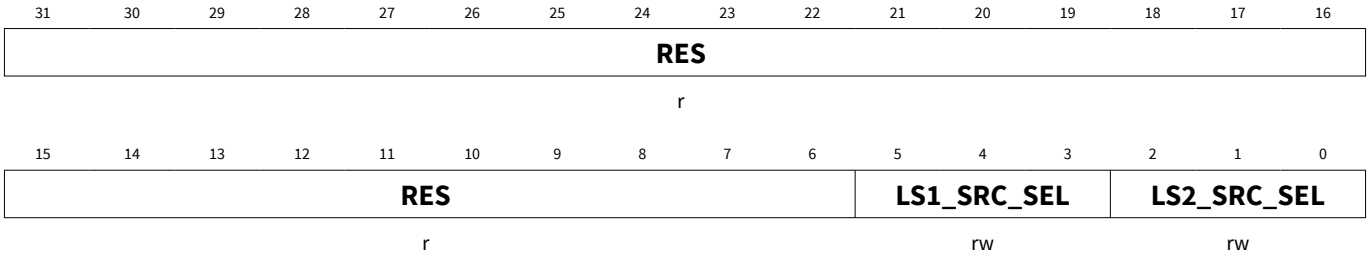
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				RES		LS1_OC_BTFILT_SEL		RES				RES		LS1_OL_BTFILT_SEL	
r				r		rw		r				r		rw	

Field	Bits	Type	Description
LS1_OL_BTFILT_SEL	1:0	rw	**Open load blank time select for LS1** 00_B **4_us**: 4 µs filter time 01_B **8_us**: 8 µs filter time 10_B **16_us**: 16 µs filter time 11_B **32_us**: 32 µs filter time
RES	3:2, 7:4, 11:10, 15:12, 18:16, 23:19, 29:24, 31:30	r	**Reserved** Always read as 0.
LS1_OC_BTFILT_SEL	9:8	rw	**Overcurrent blanktime select for LS1** 00_B **4_us**: 4 µs filter time 01_B **8_us**: 8 µs filter time 10_B **16_us**: 16 µs filter time 11_B **32_us**: 32 µs filter time

26.4.8 Low-side PWM source selection register

LS_PWMSRCSEL Offset address: $001C_H$

Low-side PWM source selection register RESET_TYPE_3 value: $0000\ 0000_H$



Field	Bits	Type	Description
LS2_SRC_SEL	2:0	rw	**LS2 PWM source selection** Note: Can be only written when LS_CTRL.LS1_PWM = 0 000_B **CC60**: PWM output of CCU6 (CC) … 010_B **CC62**: PWM output of CCU6 (CC) 011_B **COUT60**: PWM output of CCU6 (COUT) … 101_B **COUT62**: PWM output of CCU6 (COUT) 110_B **T3OUT**: PWM output of GPT12
LS1_SRC_SEL	5:3	rw	**LS1 PWM source selection** Note: Can be only written when LS_CTRL.LS1_PWM = 0 000_B **CC60**: PWM output of CCU6 (CC) … 010_B **CC62**: PWM output of CCU6 (CC) 011_B **COUT60**: PWM output of CCU6 (COUT) … 101_B **COUT62**: PWM output of CCU6 (COUT) 110_B **T3OUT**: PWM output of GPT12
RES	31:6	r	**Reserved** Always read as 0.

26.4.9 Low-side 2 reference current trimming register

LS_LS2_TRIM

Low-side 2 reference current trimming register

Offset address: 0020_H

RESET_TYPE_4 value: $0000\ 0000_H$

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES				**RES**		**LS2_OC_BTFILT_SEL**		**RES0**				**RES0**		**LS2_OL_BTFILT_SEL**	
r				r		rw		r				r		rw	

Field	Bits	Type	Description
LS2_OL_BTFILT_SEL	1:0	rw	**Open load blank time select for LS2** 00_B **4_us**: 4 µs filter time 01_B **8_us**: 8 µs filter time 10_B **16_us**: 16 µs filter time 11_B **32_us**: 32 µs filter time
RES0	3:2, 7:4	r	**Reserved**
LS2_OC_BTFILT_SEL	9:8	rw	**Overcurrent blank time select for LS2** 00_B **4_us**: 4 µs filter time 01_B **8_us**: 8 µs filter time 10_B **16_us**: 16 µs filter time 11_B **32_us**: 32 µs filter time
RES	11:10, 15:12, 18:16, 23:19, 29:24, 31:30	r	**Reserved** Always read as 0.

26.5 Interrupt generation and status bit logic

The interrupt flags of the low-side module show the following behavior:

Overcurrent detection: The overcurrent detection interrupt flag is a level sensitive interrupt flag. This flag is set when the overcurrent condition occurs and stays persistent until the condition is removed.

Overtemperature detection: The overtemperature detection interrupt flag is a level sensitive interrupt flag. This flag is set when the overtemperature condition occurs, but can be cleared immediately. The overtemperature status of the overtemperature condition can then still be monitored in the dedicated status register, which is placed in the same interrupt status register.

Open Load detection: The open load detection interrupt flag is a level sensitive interrupt flag. This flag is set when the open condition occurs, but can be cleared immediately. The open load status of the open load condition can then still be monitored in the dedicated status register, which is placed in the same interrupt status register.

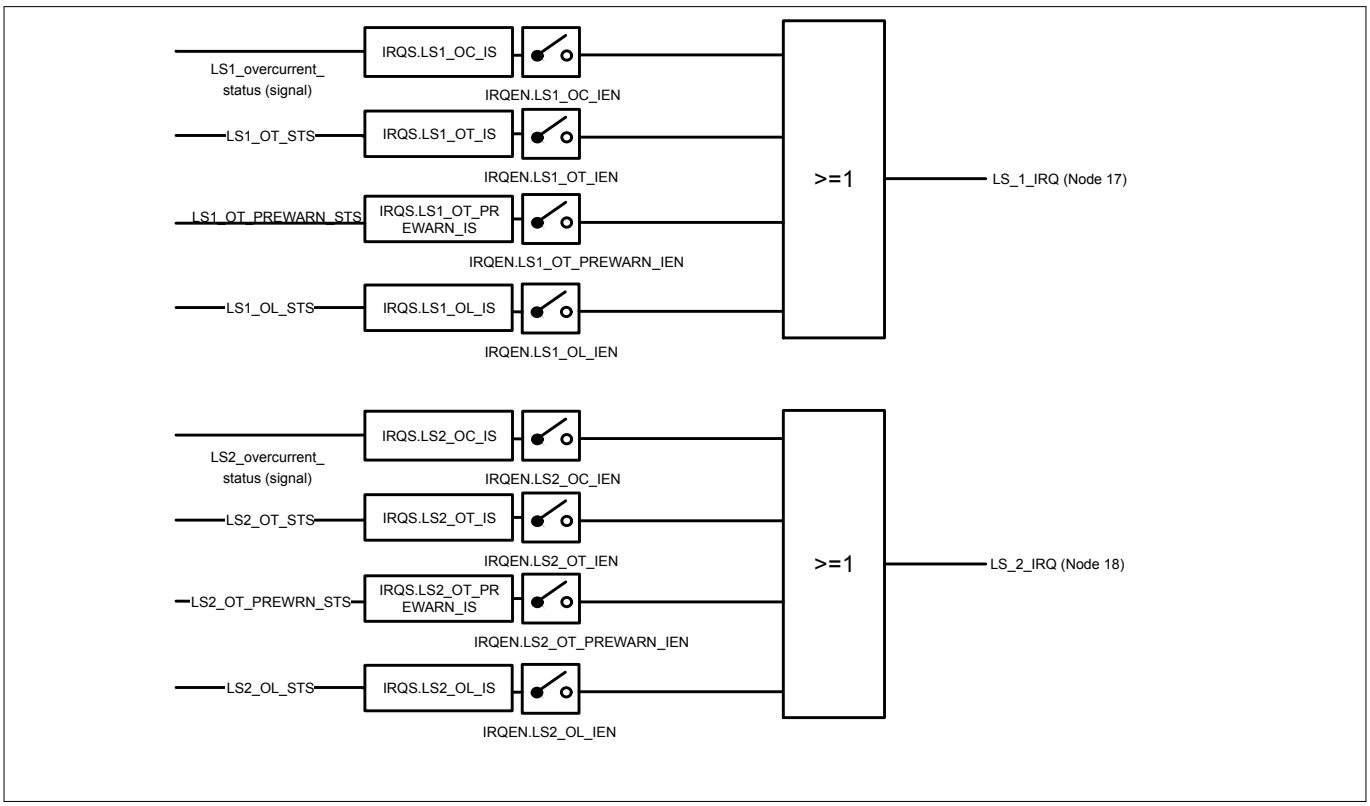


Figure 228 Low-side 1/2 switch interrupt generation

27 Application information

Note: The following information is given as a hint for the implementation of the device only and shall not be regarded as a description or warranty of a certain functionality, condition or quality of the device.

27.1 Relay window lift application diagram

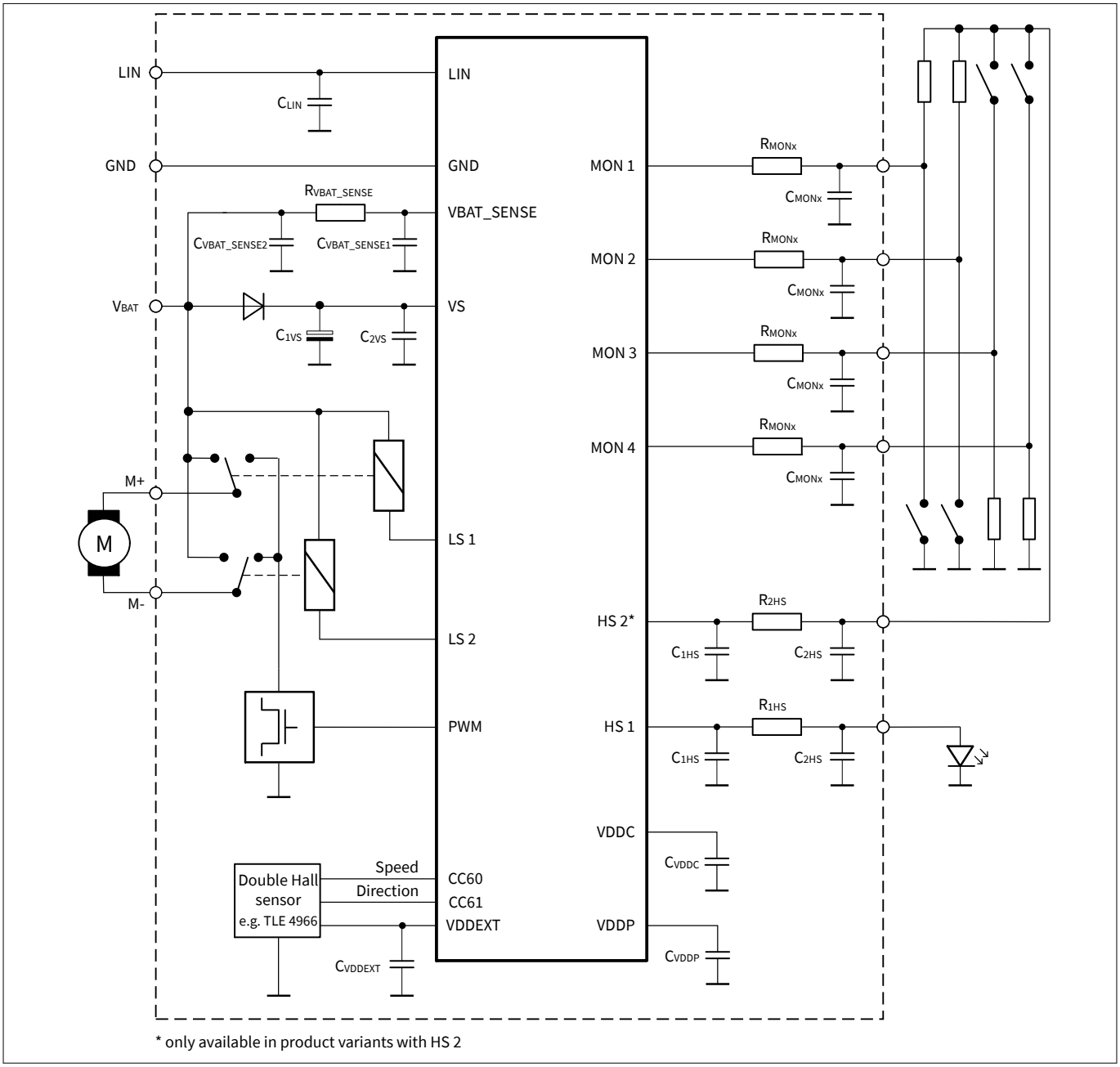


Figure 229 Simplified application diagram example

Note: This is a very simplified example of an application circuit and bill of material. The function must be verified in the actual application.

Table 185 External component (BOM)

Symbol	Function	Component
C_{1VS}	Capacitor 1 at VS pin	22 µF[1]
C_{2VS}	Capacitor 2 at VS pin	100 nF[2] [3]
C_{VDDEXT}	Capacitor at VDDEXT pin	330 nF[2]
C_{VDDC}	Capacitor at VDDC pin	100 nF[2] [3] + 330 nF[2]
C_{VDDP}	Capacitor at VDDP pin	470 nF[2] [3] + 470 nF[2]
R_{MONx}	Resistor at MONx pin	3.9 kΩ
C_{MONx}	Capacitor at MONx connector	6.8 nF[4]
R_{VBAT_SENSE}	Resistor at VBAT_SENSE pin	3.9 kΩ
C_{VBAT_SENSE1}	Capacitor 1at VBAT_SENSE pin	10 nF[2]
C_{VBAT_SENSE2}	Capacitor 2 at VBAT_SENSE connector	6.8 nF[4]
C_{LIN}	Capacitor at LIN pin	220 pF
R_{1HS}	Resistor at HS pin for LED	e.g. 2.7 kΩ
R_{2HS}	Resistor at HS pin	160 Ω[5]
C_{1HS}	Capacitor at HS pin	6.8 nF[2]
C_{2HS}	Capacitor at HS connector	33 nF[4]

1) To be dimensioned according to application requirements.
2) To reduce the effect of fast voltage transients of V_S, these capacitors should be placed close to the device pin.
3) Ceramic capacitor.
4) For ESD GUN.
5) Optional, for short to battery protection, calculated for 24 V (jump start).

27.2 Motor drive with P/N-channel power MOSFET half bridge application (TLE9845QX only)

Note: The following information is given as a hint for the implementation of the device only and shall not be regarded as a description or warranty of a certain functionality, condition or quality of the device.

27.2.1 P/N-channel half bridge application diagram

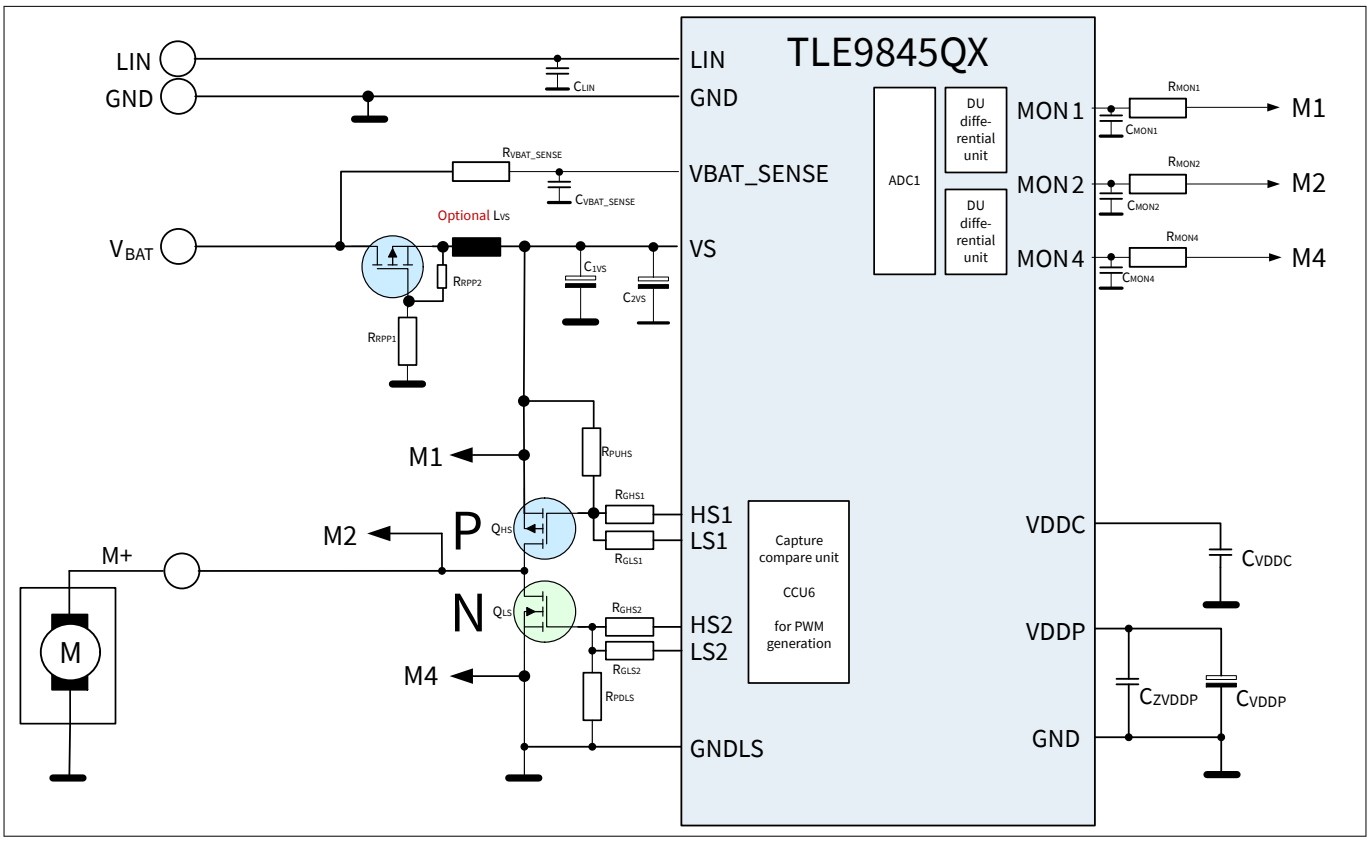


Figure 230 Scheme of relevant blocks used for single phase brushed-DC-motor control

27.2.2 Functional description

27.2.2.1 Gate driver stages

Figure 230 shows a simplified diagram with MOTIX™ TLE984xQX in an electric drive application setup controlling an uni-directional brushed-DC-motor. The driver stages support two variants to drive the motor: The non controlled motor contact can be connected to battery or to ground potential.

The half bridge is built up with a P-channel power MOSFET in high-side and a N-channel power MOSFET in low-side position. The driver stage of the P- and N-channel MOSFET is provided by the outputs HS1/LS1 and HS2/LS2, respectively. Each pair of outputs builds up a push-pull gate driver stage with fixed supply to VS and ground. The driver stages are not limited to a maximum voltage. Therefore an additional zener diode between the gate and source of each MOSFET is recommended for protection.

For adjusting the required slope of the output voltage the resistors between the outputs HSx/LSx and the power MOSFET gate have to be dimensioned in a specific way according to the application requirements. It is further recommended to additionally connect the respective gate potentials to battery (P-channel) or ground

(N-channel) by external resistors. This prevents the power half bridge from unwanted cross currents in case of a MOTIX™ TLE984xQX reset condition (driver stage is high impedance).

27.2.2.2 PWM operation

MOTIX™ TLE984xQX supports PWM controlled motor drive with active free-wheeling (that is synchronous switching of the half bridge MOSFETs to avoid body diode losses during PWM off phase) by using the CCU6 module.

The CCU6 can be configured to use the internal deadtime generation to control the switching delay between the external P- and N-channel MOSFETs of the half-bridge. In this case HSx and LSx are switched without any deadtime. The gate pre-resistors has to be dimensioned expecting the max values.

27.2.2.3 MOSFET protection with integrated differential units for drain-source-monitoring

For emergency shut-off in case of short-to-GND or short-to-VBAT, the following protection scheme can be used.

For this feature, 3 of the MON inputs (e.g. MON1, MON2, MON4) are used in combination with 2 differential measurement units (MON1-MON2, MON2-MON4), that are located in ADC1.

The differential measurement units are sampled by the ADC1 and use the post processing for threshold supervision, interrupt generation and trap handling.

The ADC measurements are triggered from CCU6, that is aligned to the PWM signals.

27.3 Connection of N.C. / N.U. pins

The device contains several N.C. (not connected, no bond wire) and possibly N.U. (not used, but bonded) pins.

Table 186 Recommendation for connecting N.C. / N.U. pins

Type	Pin number	Recommendation 1	Recommendation 2	Comment
N.C.	27, 28, 29, 38, 40, 41	GND		
N.C.	10, 46	open	GND	Neighboring high-voltage pins
N.U.	4	VS	open	In product variants with one high-side only, no HS2
N.U.	9	GND		In product variants with four MON only, no MON5

27.4 Connection of unused pins

The following table shows recommendations how to connect pins, in case they are not needed by the application.

Table 187 Recommendation for connecting unused pins (product variant dependent)

Type	Pin number	Recommendation 1 (if unused)	Recommendation 2 (if unused)
LIN	1	open	
HS1, HS2	3, 4	VS	open
MON	5, 6, 7, 8, 9	GND	open + configure internal PU/PD
LS1, LS2	11, 12	GNDLS	open
GPIO	14, 15, 16, 17, 20, 22, 23, 24, 25, 26, 33, 34, 35, 36, 37, 39	GND	External PU/PD or Open + configure internal PU/PD
TMS	18	GND	
Reset	21	open	
P2/XTAL out	31	open	
P2/XTAL in	32	GND	
VDDEXT	45	Open	
VBAT_SENSE	48	VS	

27.5 Connection of P0.2 for SWD debug mode

To enter the SWD debug mode, P0.2 needs to be 0 at the rising edge of the reset signal.

P0.2 has an internal pull-down, so it just needs to be ensured that there is no external 1 at P0.2 when the debug mode is entered.

27.6 Connection of TMS

For the debug mode, the TMS pin needs to be 1 at the rising edge of the reset signal. This is controlled by the debugger. The TMS pin has an internal PD.

To avoid the device entering the debug mode unintendedly in the final application, adding an external pull-down additionally is recommended.

27.7 ESD immunity according to IEC61000-4-2

Note: Tests for ESD robustness according to IEC61000-4-2 “gun test” (150 pF, 330 Ω) were performed. The results and test condition are available in a test report. The achieved values for the test are listed in the table below.

Table 188 ESD “Gun test”

Performed test	Result	Unit	Remarks
ESD at pin *LIN*, versus *GND*	≥6	kV	1) Positive pulse
ESD at pin *LIN*, versus *GND*	≤-6	kV	1) Negative pulse
ESD at pin *VS*, *VBAT_SENSE*, *MONX*, *HS*, versus *GND*	≥6	kV	1) Positive pulse
ESD at pin *VS*, *VBAT_SENSE*, *MONX*, *HS*, versus *GND*	≤-6	kV	1) Negative pulse

1) ESD susceptibility “ESD GUN”, tested by external test house (IBEE Zwickau, EMC test report nr. 11-01-16), according to "LIN conformance test specification package for LIN 2.1, October 10th, 2008" and "hardware requirements for LIN, CAN and FlexRay interfaces in automotive application – AUDI, BMW, Daimler, Porsche, Volkswagen – Revision 1.3/2012".

Revision history

Document version	Date of release	Description of changes
Rev. 1.2	2021-10-08	**General changes** • Registers restructuring: Added overview tables, moved registers definition to the end of the module chapters • Added notes in registers that HS2 and MON5 are product variant specific • Editorial changes
		Overview • Overview: Updated description
		Device pinout and pin configuration • Pin configuration: Updated description
		Introduction (chapter restructured) • SOC system power modes overview: Added heading: Header "Modes of operation" renamed to "SOC system power modes overview", and moved to chapter "Introduction"; Updated description; Removed figure • Device reset masks: Added description
		Power management unit (PMU) (chapter restructured) • PMU modes overview: Updated description and moved figures into other chapters • Power supply generation (PGU): Updated description • VDDEXT internal diagnosis: Added • Low-V_S operation: Added • PGU ADC2 monitoring: Added • Power control unit (PCU): Updated description; Figure 12 and Table 9 added • Fail safe scenarios: Moved to Fail Sleep mode and updated descriptions • Reset management unit (RMU): Move before chapter Wake-up management unit, updated description and Figure 20 • RESET pin: Added • Figure 21: Updated • Block diagram: Updated Figure 4 • Configuration of cyclic sense mode : Updated Figure 23 • Cyclic sense mode: Updated description and Figure 22 • External voltage regulator 5.0 V (VDDEXT) Updated description • Power supervision function of PCU: Updated description • Wake-up management unit (WMU): Updated description and Figure 24 • PMU_GPIO_WAKE_STATUS: Updated GPIO1_STS_x bitfields • PMU_MON_CNFx: Updated MONx_CYC bitfields • PMU_RESET_STS: Updated SYS_FAIL and PMU_WAKE bitfield • PMU_VDDEXT_CTRL: Updated reset masks • PMU_WAKE_STATUS: Updated MON, GPIO1, FAIL, VDDEXT_OT and VDDEXT_UV bitfields • PMU_WFS: Updated SUPP_SHORT, PMU_5V_OVL, SUPP_TMOUT, SYS_CLK_WDT, SYS_OT and LP_CLKWD bitfields

Document version	Date of release	Description of changes
		System control unit - digital modules (SCU-DM) • Baud-rate generator: Updated description • Bit protection register: Updated description • Clock control unit: Updated #unique_125/ unique_125_Connect_42_fig_th3_xpf_r4b • Clock generation unit: Updated Figure 26 • Extended interrupts: Changed description • External crystal mode: Updated description • External input clock mode: Updated description • Features: Updated description • Free running mode: Updated description • Internal oscillator (OSC_PLL): Updated description • PLL functional description: Updated description and Figure 28 • Providing an input clock to the PLLUpdated description • Sleep mode: Updated description • Switching PLL parameters: Updated description • Wake-up reset: Updated description • SCU_APCLK: Updated APCLK1FAC and APCLK2FAC bitfields • SCU_APCLK_STS: Updated description of APCLK1STS and APCLK2STS bitfields • SCU_APCLK_SCLR: Updated description • SCU_CMCON0: Updated XTAL_ON description • SCU_CMCON1: Updated K2DIV description • SCU_MEMSTAT: Updated SASTATUS bitfield • SCU_MODPISEL: Updated URIOS1 and U_TX_CONDIS description • SCU_MODPISEL3: Updated URIOS2 description • SCU_MODSUSP: Updated RES description • SCU_NVM_PROT_STS: Updated DIS_RDUS and DIS_RDUS_S0 descriptions • SCU_PMCON: Changed "CCU_DIS" to "CCU6_DIS" • SCU_PMCON0: Updated description • Added SCU_ prefix to OSC_CON, PLL_CON, CMCON1, CMCON2, SYSCON0
		System control unit - power modules (SCU-PM) • Structure of PREWARN_SUP_NMI: Updated Figure 40 • SCUPM_SYS_IS: Updated LIN_FAIL_STS and LIN_FAIL_IS bitfields • SCUPM_SYS_ISCLR: Updated bitfield descriptions
		Arm® Cortex®-M0 core • CPU_AIRCR: Updated VECTKEY description • CPU_CPUID: Updated CONSTANT description • CPU_SYSTICK_CSR: Updated CLKSOURCE description
		Address space organization • Address space organization: Updated description
		Memory control unit

Document version	Date of release	Description of changes
		• Functional features for RAM: Updated description • NVM protection modes: Updated description
		NVM module • Hot spot distribution: Updated description • Linearly mapped sectors: Updated description • Timing : Updated description
		Interrupt system (chapter reworked) • Overview: Updated description • Interrupt node assignment: Updated figures in all subchapters • Interrupt flags overview: Updated Table 68 • Interrupt source and vector: Updated Table 69
		LIN transceiver (chapter reworked)
		GPIO ports and peripheral I/O • General port x register description: Removed chapter with all subchapters
		General purpose timer units (GPT12) • Counter mode: Encoding of GPT1 input edge selection: Added Table 89Changed Table 90 • GPT12E_PISEL: Updated description • GPT12E_T2CON: Updated T2I description • GPT12E_T4CON: Updated T4I description
		Capture/compare unit 6 (CCU6) • CCU6_T12MSEL: Updated MSEL60 and MSEL61 bitfields
		UART1/UART2 • Interfaces of the UART module: Updated description, Figure 170 and Figure 171
		Measurement unit • Block diagram: Updated Figure 188 • 10-bit ADC channel allocation: Updated Figure 191 and Figure 192
		Measurement core module (incl. ADC2) • Calibration unit control registers: Updated description • Post processing default values: Updated description • ADC2_CTRL_STS: Updated RES bitfield • ADC2_CHx_EIM: Updated EN and SEL descriptions.
		10-bit analog digital converter (ADC1) • "ADC1 Interrupt Generation for EIM and ESM Mode": Deleted figure • Functional description: Updated Figure 209, Figure 210, Figure 211 and Figure 212 • Implementation of differential measurement unit: Updated description and Figure 215 • Module interfaces: Added chapter • Setup of calibration unit: Updated equation

Document version	Date of release	Description of changes
		• Start-up behavior after reset: Updated description • ADC1_CHx_EIM: Updated EIM_EN description • ADC1_FILT_OUT0: Updated description • ADC1_FILT_UP_CTRL: Updated RES bitfield
		High-voltage monitor input • "High-voltage monitor input registers": Changed description
		High-side switch • HS_CTRL: Updated HS1_OC_SEL and HS2_OC_SEL bitfields • HS_CTRL, HS_IRQS, HS_IRQCLR, HS_IRQEN, HS_HS2_TRIM, HS_PWMSRCSEL: Added note that HS2 is device variant specific
		Low-side switch • Application requirement for low-side switch in PWM mode: Updated Figure 227 • Block diagram: Updated Figure 225 • LS_LS2_TRIM: Updated RES bitfields [7:2]
Rev 1.1	2019-03-18	(Following chapter numbers are only valid for Rev 1.1 of the user manual.) **Chapter 5** updated register types overview **Chapter 6** removed PMU_CPREG_CNF and PMU_WAKE_CNF_GPIO0 registers, as GPIO0 cannot be selected PMU_WAKE_STATUS, Bits MONx_WAKE_STS (x=1..5): added sentence about clearing before entering power saving modes (moved from PMU_MON_CNF1+2: MONx_STS (x=1..5)) **Chapter 7** Clock Tree drawing adapted, esp. ADC1_CLK **Chapter 7.3.4** added TFILT_CLK explanation **Chapter 7,** **Chapter 13** SCU_EXICON0 edge configuration adapted for EXINT0, EXINT1 and EXINT2 removed "interrupt structure 2" references **Chapter 8** SCUPM_SYS_IS: Bit 28 (SYS_SUPPLY_STS) and Bit 12 (SYS_SUPPLY_IS) do not work, but are not needed as NMI is available. Made bits "reserved" **Chapter 11.6.2.2** added "Application hint regarding read-protection" **Chapter 13.4.1** Interrupt Structure 1: added description for MON/EXTINT interrupt behaviour **Chapter 20** Info added: LIN VS-undervoltage needs ADC1 running **Chapter 24** ADC1_CTRL_STS.SOOC removed ADC1_STATUS removed (contained SD_FEEDB_ON, SOC_JITTER, DAC_IN)

Document version	Date of release	Description of changes
		added information for ESM_STS: this bit has to be cleared, additionally to ESM_IS, before further ESM-interrupts can be triggered **Chapter 25** MONx_STS (x=1..5): restrictions regarding update MONx_STS (x=1..5): removed sentence about clearing before entering power saving modes (moved to PMU_WAKE_STATUS, Bits MONx_WAKE_STS) **Chapter 26.3.1.5** added chapter "low-VS Feature" **Chapter 26.3.2** updated figure for HS PWM control **Chapter 27.2.2** updated figure for LS PWM control **Chapter 6** Reset type of VDDEXT_CTRL.VDDEXT_UV_IS changed PMU_WAKE_STATUS: renamed LIN to LIN_WAKE PMU_WAKE_STATUS.GPIO2 removed PMU_WAKE_CNF_GPIO1, removed bits for GPIO1.3 **Chapter 6,** **Chapter 7** VDDC in stop mode: changed 0.9V to more generic "reduced voltage", in text, register bit, and pictures **Chapter 7** SCU_MODPISEL3: adapted URIOS2 Bit description for clarification SCU_MEM_ACC_STS: Reset_type_3 -> 4 **Chapter 7,** **Chapter 13** External Interrupt Control Registers: editorial changes **Chapter 8** editorial changes in SCUPM_SYS_SUPPLY_IRQ_STS and SCUPM_SYS_SUPPLY_IRQ_CLR "VS-Overvoltage System Shutdown" chapter removed (feature cannot be enabled by user), "Overtemperature System Shutdown" chapter adapted (feature cannot be disabled by user **Chapter 9** added reset-types RESET_TYPE_3 changed bitfield-width 8->2, changed access-types, changed reset-value CCR **Chapter 13** editorial: removed "wakeup" at nodes 12 and 22 in "Interrupt Vector Table" editorial: removed "wakeup" at node 12 and "VREF5V" at node 3 in "Interrupt Node Table" editorial: SCU_NMICON.0 is reserved (was already correct in SCU-DM) editorial: SCU_EXICON1 reset value adapted to 0 (was already correct in SCU-DM) **Chapter 13,**

Revision history

Document version	Date of release	Description of changes
		Chapter 26, **Chapter 27** no OC_STS-bit (signal only) for HS and LS overcurrent **Chapter 22** Tables "ADC2 Channel Selection and Voltage Ranges" and "ADC1 Channel Selection and Voltage Ranges" simpliefied and corrected **Chapter 23** corrected description of ADC2_FILT_UP_CTRL: for bypassing IIR filter, not for bypassing threshold counter **Chapter 25** removed NSLEEP description, which does not apply to this device **Chapter 26.3.1.1** chapter "Slew Rate Configuration" adapted for clarification regarding 5..10kHz PWM **all chapters** editorial changes
Rev 1.0	2016-06-20	Initial revision

Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

Edition 2021-10-08

Published by

Infineon Technologies AG

81726 Munich, Germany

Do you have a question about any aspect of this document?

Email: erratum@infineon.com

Document reference
Z8F55202499

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.

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.