UM11607

LPC86x User manual

Rev. 3 — April 2023 User manual

Document information

Info	Content
Keywords	LPC860, LPC86x, LPC860 UM, LPC86x UM, LPC860 user manual, LPC86x user manual
Abstract	LPC86x user manual

Revision history

Rev	Date	Description
Rev. 3	April, 2023	Initial release

Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

UM11607

Chapter 1: Introductory information

Rev. 3 — April 2023 User manual

1.1 Introduction

The LPC86x is an Arm Cortex-M0+ based, low-cost 32-bit MCU family operating at CPU frequencies of up to 60 MHz. The LPC86x supports up to 64 KB of flash memory and 8 KB of SRAM.

The peripheral complement of the LPC86x includes a CRC engine, one I2C-bus interface, one I3C-MIPI bus interface, up to three USARTs, up to two SPI interfaces, one multi-rate timer, self-wake-up timer, two FlexTimers (one with full motor control feature), a DMA, one 12-bit ADC, one analog comparator, function-configurable I/O ports through a switch matrix, an input pattern match engine, and up to 54 general-purpose I/O pins.

Remark: For additional documentation, see Section 31.2 “References”.

1.2 Features

- System:
  - Arm Cortex-M0+ processor (revision r0p1), running at frequencies of up to 60 MHz with single-cycle multiplier and fast single-cycle I/O port.
  - Arm Cortex-M0+ built-in Nested Vectored Interrupt Controller (NVIC).
  - System tick timer.
  - AHB multilayer matrix.
  - Serial Wire Debug (SWD) with four break points and two watch points. JTAG boundary scan (BSDL) supported.
- Memory:
  - Up to 64 KB on-chip flash programming memory with 64 Byte page write and erase.
  - Code Read Protection (CRP).
  - Up to 8 KB SRAM consisting of contiguous SRAM banks.
  - Bit-band addressing supported to permit atomic operations to modify a single bit.
- ROM API support:
  - Bootloader.
  - Supports Flash In-Application Programming (IAP).
  - Supports In-System Programming (ISP) through USART.
  - FRO API.
  - Flash In-Application Programming (IAP) and In-System Programming (ISP).
- Digital peripherals:
  - High-speed GPIO interface connected to the ARM Cortex-M0+ IO bus with up to 54 General-Purpose I/O (GPIO) pins with configurable pull-up/pull-down resistors, programmable open-drain mode, input inverter, and glitch filter. GPIO direction control supports independent set/clear/toggle of individual bits. The GPIO pin is tristate when power in on.

  - High-current source output driver (20 mA) on four pins.
  - High-current sink driver (20 mA) on two true open-drain pins.
  - GPIO interrupt generation capability with boolean pattern-matching feature on eight GPIO inputs.
  - Switch matrix for flexible configuration of each I/O pin function.
  - CRC engine.
  - DMA with 16 channels and 13 trigger inputs.
- Timers:
  - Two FlexTimers with DMA support and a selection of hardware triggers. The first FlexTimer has six channels and includes support for motor control (including Fault Control). The second FlexTimer has four channels. This timer does not include Fault Control but includes a Quadrature Decoder interface. Both FlexTimers are clocked up to 60 MHz.
  - Four channel Multi-Rate Timer (MRT) for repetitive interrupt generation at up to four programmable, fixed rates.
  - Self-Wake-up Timer (WKT) clocked from either Free Running Oscillator (FRO), a low-power, low-frequency internal oscillator, or an external clock input in the always-on power domain.
  - Windowed Watchdog timer (WWDT).
- Analog peripherals:
  - One 12-bit ADC with up to 12 input channels with multiple internal and external trigger inputs and with sample rates of up to 1.9 Msamples/s. The ADC supports two independent conversion sequences.
  - Comparator with five input pins and external or internal reference voltage.
- Serial peripherals:
  - Three USART interfaces with pin functions assigned through the switch matrix, support receive idle interrupt, and two fractional baud rate generators.
  - Two SPI controller with pin functions assigned through the switch matrix.
  - One I^{2}C-bus interface. I^{2}C supports Fast-mode Plus with 1 Mbit/s data rate on two true open-drain pins and listen mode.
  - One master/slave I3C-MIPI bus interface. The I3C support DDR. It is supported by the general purpose DMA controller.

- Clock generation:
  - Free Running Oscillator (FRO). This oscillator provides a selectable 60 MHz, 48 MHz, and 36 MHz outputs that can be used as a system clock. Also, these outputs can be divided down to 30 MHz, 24 MHz, and 18 MHz for system clock. The FRO is trimmed to ±1 % accuracy over the entire voltage and temperature range of 0 °C to 70 °C.
  - External clock input for clock frequencies of up to 25 MHz.
  - Crystal oscillator with an operating range of 1 MHz to 25 MHz.
  - 1 MHz (±3%) low power oscillator(LPOSC) can be used as a clock source to the watchdog timer.
  - PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. May be run from the system oscillator, the external clock input, or the internal FRO.
  - Clock output function with divider that can reflect all internal clock sources.
- Power control:
  - Reduced power modes: sleep mode, deep-sleep mode, power-down mode, and deep power-down mode.
  - Wake-up from deep-sleep and power-down modes on activity on USART, SPI, I2C, and I3C peripherals.
  - Timer-controlled self wake-up from deep power-down mode.
  - Power-On Reset (POR).
  - Brownout detect (BOD).
- Unique device serial number for identification.
- Single power supply (1.8 V to 3.6 V).
- Operating temperature range -40 °C to +105 °C.
- Available in LQFP64, HVQFN48, and HVQFN32 packages.

1.3 Ordering options

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
LPC865M201JBD64	LQFP64	Plastic low profile quad flat package; 64 leads; body 10× 10 × 1.4 mm	SOT314-2
LPC865M201JHI48	HVQFN48	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 48 terminals; body 7× 7 × 0.85 mm	SOT619-1
LPC865M201JHI33	HVQFN32	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 32 terminals; body 5× 5 × 0.85 mm	SOT617-11

Table 2. Ordering options

Type number	CPU/MHz	Flash/KB	SRAM/KB	USART	I^2C	I^3C	SPI	GPIO	Package
LPC865M201JBD64	60	64	8	3	1	1	2	54	LQFP64
LPC865M201JHI48	60	64	8	3	1	1	2	42	HVQFN48
LPC865M201JHI33	60	64	8	3	1	1	2	29	HVQFN32

1.4 General description

1.4.1 ARM Cortex-M0+ core configuration

The ARM Cortex-M0+ core runs at an operating frequency of up to 60 MHz. Integrated in the core are the NVIC and Serial Wire Debug with four breakpoints and two watch points. The ARM Cortex-M0+ core supports a single-cycle I/O enabled port (IOP) for fast GPIO access at address 0xA000 0000. The ARM Cortex M0+ core version is r0p1.

The core includes a single-cycle multiplier and a system tick timer (SysTick).

1.5 Block diagram

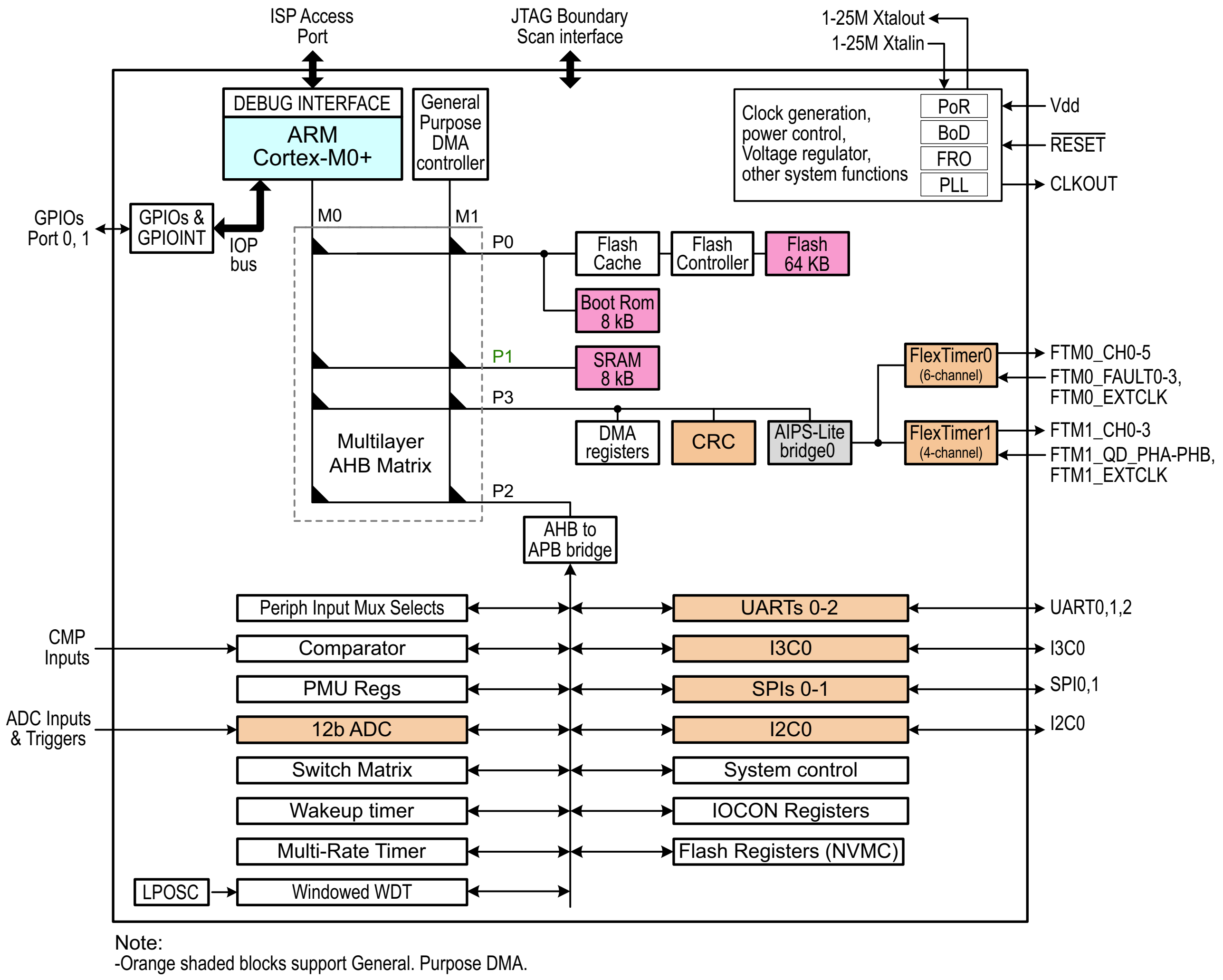


Fig 1. LPC86x block diagram

Chapter 2: Memory mapping

2.1 How to read this chapter

The memory mapping is identical for all LPC86x parts. Different LPC86x parts support different flash and SRAM memory sizes.

2.2 General description

The LPC86x incorporates several distinct memory regions. Figure 2 shows the overall map of the entire address space from the user program viewpoint following reset.

The APB peripheral area is 512 KB in size and is divided to allow for up to 32 peripherals. Each peripheral is allocated 16 KB of space simplifying the address decoding.

The registers incorporated into the ARM Cortex-M0+ core, such as NVIC, SysTick, and sleep mode control, are located on the private peripheral bus.

The GPIO port and pin interrupt/pattern match registers are accessed by the ARM Cortex-M0+ single-cycle I/O enabled port (IOP).

2.2.1 Memory mapping

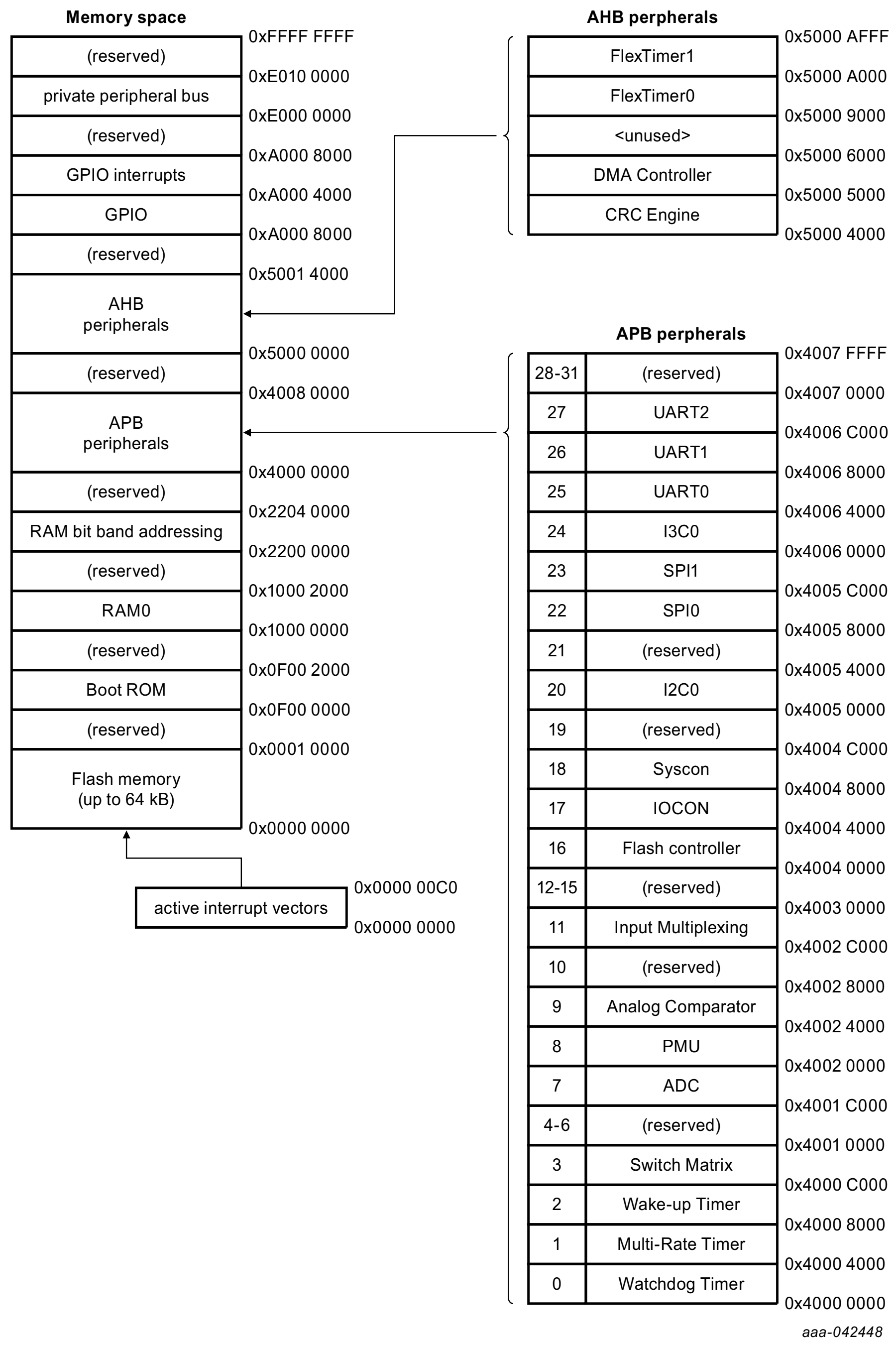


Fig 2. Memory mapping

UM11607

Chapter 3: Boot Process

Rev. 3 — April 2023 User manual

3.1 How to read this chapter

The bootloader is identical for all parts.

3.2 Features

- On-chip boot ROM
- Contains the bootloader with In-System Programming (ISP) facility over UART communication and the following API:
  - In-Application Programming (IAP) of flash memory.
  - FRO API.

3.3 Basic configuration

Depending on the configuration, the boot ROM can set the FRO control register to select the operating frequency accordingly.

Remark: At default, ROM runs from 24 MHz FRO clock.

3.4 Pin description

When the ISP entry pin (PIO0_12) is pulled LOW on reset, the part enters ISP mode and the ISP command handler starts up.

PIO0_25 is UART0 TX, and PIO0_24 is UART0 RX, are used for USART ISP mode.

3.5 General description

3.5.1 Bootloader

The bootloader executes every time the device is powered on or reset. Based on the chip configuration information, the bootloader controls initial operation after reset, including setting internal voltage regulator, system clock, flash controller, miscellaneous factory trimming value, and then allows programming and reprogramming of internal flash via a set of commands on USART. The LPC86x device must be connected to a host system that provides the UART master connections.

During the boot process, a LOW level after reset on the ISP pin is considered as an external hardware request to start the ISP command handler via USART. Otherwise, the bootloader checks if there is valid user code in flash. If the valid user code is not found, ROM enters the UART ISP mode.

Remark: The sampling of pin the ISP entry pin can be disabled through programming flash location 0x0000 02FC (see Section 4.3.6 “Code Read Protection (CRP)”).

See Chapter 4 “ISP and IAP” for more details.

3.5.2 ROM-based APIs

Once the part has booted, the user can access several APIs located in the boot ROM. The ROM API supports:

- Flash In-Application Programming (IAP).
- FRO API.

The structure of the boot ROM APIs is shown in Figure 3.

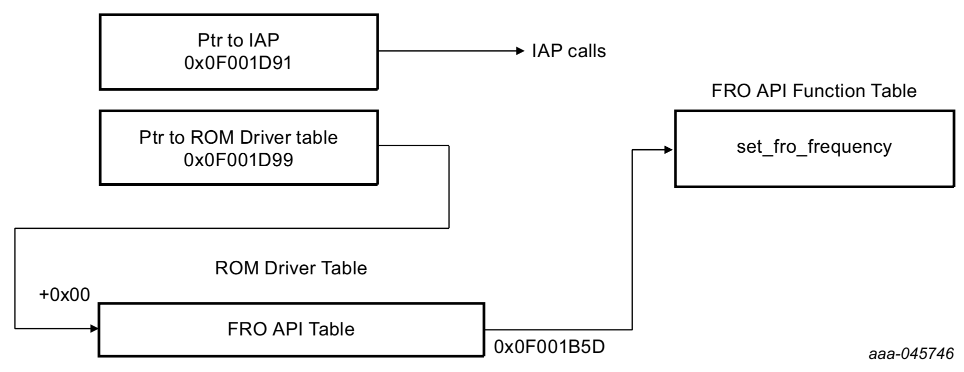


Fig 3. Boot ROM structure

The boot ROM structure should be included as follows:

```
#define IAP_API_BASE (0x0F001D90)
typedef struct {
     const uint32_t FRO_API_T *FROApiBase;
}LPC_ROM_API_T

     const FRO_API_T fro_api=
{
     &set_fro_frequency
};

#define ROM_DRIVER_BASE (0x0F001D98)
```

Table 3. API calls

API	Description	Reference
Flash IAP	Flash In-Application programming. Base address is 0x0F001D90.	Table 27
LPC ROM API	Base address is 0x0F001D98.	
FRO API	Free Running Oscillator configuration	10.4

3.6 Functional description

3.6.1 Memory map after any reset

The boot ROM block is 8 KB in size. The boot block is located in the memory region starting from address 0x0F00 0000. The bootloader is designed to run from this memory area, but both the ISP and IAP software use parts of the on-chip RAM. The RAM usage is described in Section 4.3.7 “ISP interrupt and SRAM use”. The interrupt vectors residing in the boot block of the on-chip flash memory also become active after reset, that is the bottom 512 bytes of the boot block are also visible in the memory region starting from the address 0x0000 0000.

3.6.2 Boot process

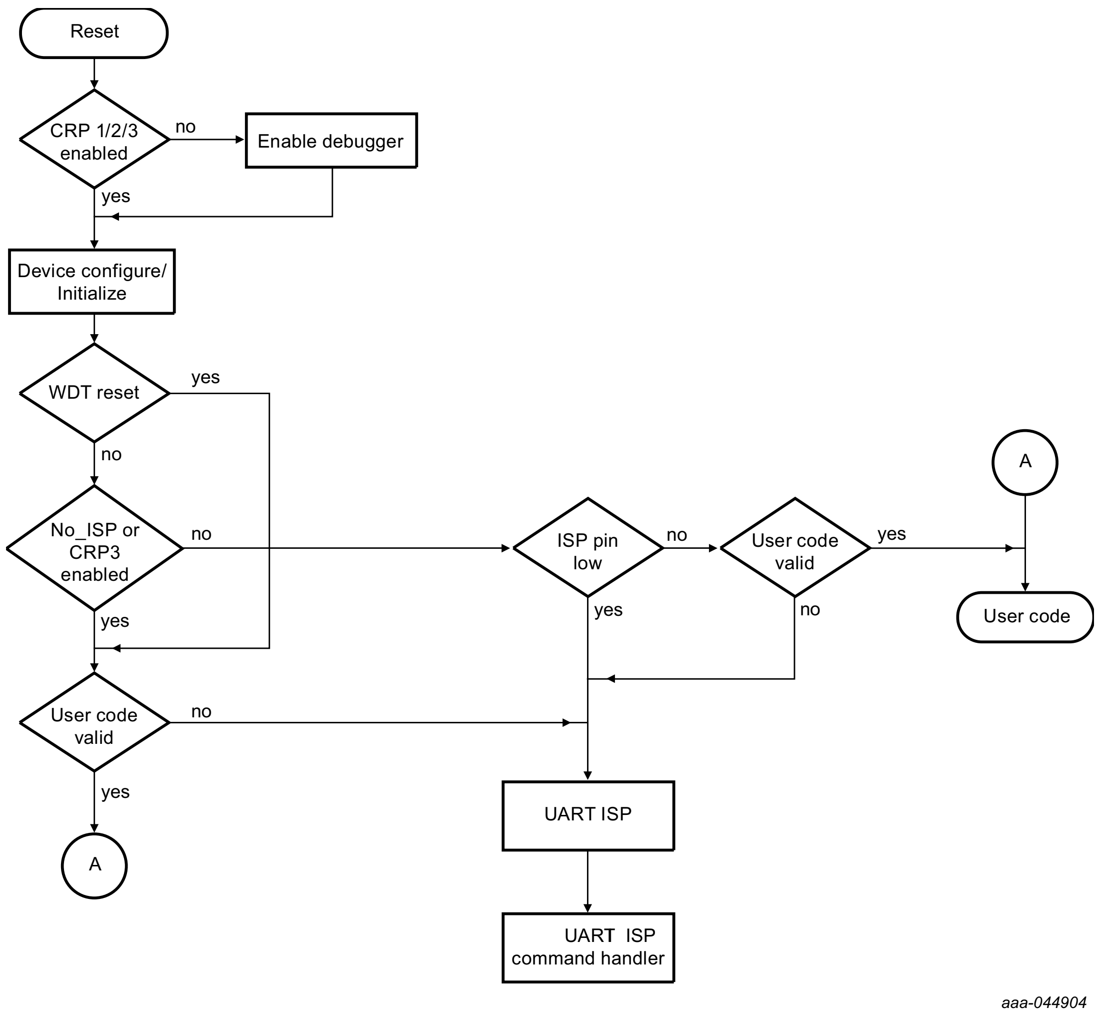


Fig 4. Boot process flowchart

UM11607

Chapter 4: ISP and IAP

Rev. 3 — April 2023 User manual

4.1 How to read this chapter

All LPC86x devices include ROM-based services for programming and reading the flash memory in addition to other functions. In-System Programming works on an unprogrammed or previously programmed device using one from a selection of hardware interfaces. In-Application Programming allows application software to do the same kinds of operations.

See specific device data sheets for different flash configurations.

Remark: In addition to the ISP and IAP commands, the flash configuration register (FLASHCFG) can be accessed in the flash controller block to configure flash memory access times, see Section 5.4.1.

4.2 Features

- In-System Programming: In-System programming (ISP) is programming or reprogramming the on-chip flash memory, using the boot loader software and USART. This can be done when the part resides in the end-user board.
- In Application Programming: In-Application (IAP) programming is performing erase and write operation on the on-chip flash memory, as directed by the end-user application code.
- Small size (64 byte) page erase programming.

4.3 General description

4.3.1 Boot loader

For the boot loader operation and boot pin, see Chapter 3 “Boot Process”.

The boot loader version can be read by ISP/IAP calls (see Section 4.5.13 or Section 4.6.6).

4.3.2 Memory map after any reset

The boot ROM is located in the memory region starting from the address 0x0F00 0000. The boot loader is designed to run from this memory area, but both the ISP and IAP software use parts of the on-chip RAM. The RAM usage is described later in Section 4.3.7.

4.3.3 Flash content protection mechanism

The LPC86x is equipped with the Error Correction Code (ECC) capable Flash memory. The purpose of an error correction module is twofold. Firstly, it decodes data words read from the memory into output data words. Secondly, it encodes data words to be written to the memory. The error correction capability consists of single bit error correction with Hamming code.

The operation of the ECC is transparent to the running application. The ECC content itself is stored in a flash memory not accessible by the user's code to either read from it or write into it on its own. Six bits of ECC corresponds to every consecutive 32 bit of the user accessible Flash. Consequently, Flash bytes from 0x0000 0000 to 0x0000 0003 are protected by the first 6-bit ECC, Flash bytes from 0x0000 0004 to 0x0000 0007 are protected by the second 6-bit ECC byte, etc.

Whenever the CPU requests a read from user's Flash, both 32 bits of raw data containing the specified memory location and the matching ECC byte are evaluated. If the ECC mechanism detects a single error in the fetched data, a correction will be applied before data are provided to the CPU. When a write request into the user's Flash is made, write of user specified content is accompanied by a matching ECC value calculated and stored in the ECC memory.

When a sector of Flash memory is erased, the corresponding ECC bytes are also erased. Once an ECC byte is written, it can not be updated unless it is erased first. Therefore, for the implemented ECC mechanism to perform properly, data must be written into the flash memory in groups of 4 bytes (or multiples of 4), aligned as described above.

4.3.4 Criteria for Valid User Code

The reserved CPU exception vector location 7 (offset 0x0000 001C in the vector table) should contain the 2's complement of the check-sum of table entries 0 through 6. This causes the checksum of the first 8 table entries to be 0. The boot loader code checksums the first 8 locations in sector 0 of the flash.

If the checksum is not zero indicating valid user code is not found, the bootloader enters UART ISP mode directly, regardless the NO_ISP, CRP1, CRP2, CRP3 signature at flash location 0x2FC.

For such a case, when bootloader enters UART ISP mode directly, if the signature at flash location 0x2FC is NO_ISP, CRP3, then the device's CRP level is CRP2.

Once the valid user code is found, boot loader will enable the cache and buffer bits of the Flash Cache Configuration register before jumping into the user code.

4.3.5 Flash partitions

Some IAP and ISP commands operate on sectors and specify sector numbers. In addition, a page erase command is available. The size of a sector is 1 KB and the size of a page is 64 Byte. One sector contains 16 pages.

Table 4. LPC86x flash configuration

Sector number	Sector size [KB]	Page number	Address range	LPC865 flash (KB)	LPC864 flash (KB)
0~3	1	0 -63	0x00000000~0x00000FFF	Yes	Yes
4~7	1	64-127	0x00001000~0x00001FFF	Yes	Yes
8~11	1	128-191	0x00002000~0x00002FFF	Yes	Yes
12~15	1	192-255	0x00003000~0x00003FFF	Yes	Yes
16~19	1	256-319	0x00004000~0x00004FFF	Yes	Yes
20~23	1	320-383	0x00005000~0x00005FFF	Yes	Yes

Table 4. LPC86x flash configuration

Sector number	Sector size [KB]	Page number	Address range	LPC865 flash (KB)	LPC864 flash (KB)
24~27	1	384-447	0x00006000~0x00006FFF	Yes	Yes
28~31	1	448-511	0x00007000~0x00007FFF	Yes	Yes
32~35	1	512-575	0x00008000~0x00008FFF	Yes	No
36~39	1	576-639	0x00009000~0x00009FFF	Yes	No
40~43	1	640-703	0x0000A000~0x0000AFFF	Yes	No
44~47	1	704-767	0x0000B000~0x0000BFFF	Yes	No
48~51	1	768-831	0x0000C000~0x0000CFFF	Yes	No
52~55	1	832-895	0x0000D000~0x0000DFFF	Yes	No
56~59	1	896-959	0x0000E000~0x0000EFFF	Yes	No
60~63	1	960-1023	0x0000F000~0x0000FFFF	Yes	No

4.3.6 Code Read Protection (CRP)

Code Read Protection is a mechanism that allows the user to enable different levels of security in the system so that access to the on-chip flash and use of the ISP can be restricted. When needed, CRP is invoked by programming a specific pattern in the flash image at offset 0x0000 02FC. IAP commands are not affected by the code read protection.

Table 1 shows the limitations of the USART ISP commands when CRP (CRP1, CRP2, or CRP3) is enabled.

Note: Any CRP change becomes effective only after the device has gone through a power cycle.

Table 5. USART ISP command limitations in CRP modes

Name	Pattern programmed in 0x0000 02FC	Description
NO_ISP	0x536A AC95	Access to the chip via the SWD pins is enabled. Prevents sampling of the pins for entering ISP mode. ISP sampling pin is available for other applications.
CRP0/NO_CRP	0xFFFFFFFF	All USART ISP commands are supported.
CRP1	0x5963 A69C	Access to the chip via the SWD pins is disabled. This mode allows partial flash update using the following USART ISP commands and restrictions: • Write to RAM command cannot access RAM below 0x1000 0600. Access to addresses below 0x1000 0600 is disabled. • Copy RAM to flash command cannot write to Sector 0. • Erase command can erase Sector 0 only when all sectors are selected for erase. • Compare command is disabled. • Read Memory command is disabled. This mode is useful when CRP is required and flash field updates are needed but all sectors can not be erased. Since compare command is disabled in case of partial updates the secondary loader should implement checksum mechanism to verify the integrity of the flash.

Table 5. USART ISP command limitations in CRP modes

Name	Pattern programmed in 0x0000 02FC	Description
CRP2	0x9635 69CA	Access to chip via the SWD pins is disabled. The following ISP commands are disabled: • Read Memory • Write to RAM • Go • Copy RAM to flash • Compare When CRP2 is enabled the ISP erase command only allows erasure of all user sectors.
CRP3	0x6359 9CA6	Access to chip via the SWD pins is disabled. ISP entry selected via the ISP entry pin is disabled if a valid user code is present in flash sector 0. This mode effectively disables ISP override using the entry pin. It is up to the application of the user to provide a flash update mechanism using IAP calls or call reinvoke ISP command to enable flash update via USART. **Caution: If CRP3 is selected, no future factory testing can be performed on the device.**
Others		All the value other than mentioned above are treated as CRP2.

In case a CRP mode is enabled and access to the chip is allowed via the ISP, an unsupported or restricted ISP command will be terminated with return code CODE_READ_PROTECTION_ENABLED.

4.3.6.1 ISP entry protection

In addition to the three CRP modes, the user can prevent the sampling of the pin for entering ISP mode and thereby release the pin for other applications. This is called the NO_ISP mode. The NO_ISP mode can be entered by programming the pattern 0x4E69 7370 at location 0x0000 02FC.

The NO_ISP mode is identical to the CRP3 mode except for SWD access, which is allowed in NO_ISP mode but disabled in CRP3 mode. The NO_ISP mode does not offer any code protection.

4.3.6.2 ISP entry configuration and detection

The ISP mode allows programming and reprogramming of the internal FLASH via a set of commands on the UART.

In auto detection mode, the chip enables fixed GPIO port and pins, and enter ISP mode on a successful auto baud detection on USART.

4.3.7 ISP interrupt and SRAM use

4.3.7.1 Interrupts during IAP

The on-chip flash memory is not accessible during erase/write operations. When the user application code starts executing, the interrupt vectors from the user flash area are active. Before making any IAP call, either disable the interrupts or ensure that the user interrupt vectors are active in RAM and that the interrupt handlers reside in RAM. The IAP code does not use or disable interrupts.

4.3.7.2 RAM used by ISP command handlers

The stack of UART ISP commands is located at address 0x1000 0600. The maximum stack usage is 1536 bytes (0x600) and grows downwards.

Memory for the USART ISP commands is allocated dynamically.

4.4 USART ISP communication protocol

All USART ISP commands should be sent as single ASCII strings. Strings should be terminated with Carriage Return (CR) and/or Line Feed (LF) control characters. Extra <CR> and <LF> characters are ignored. All ISP responses are sent as <CR><LF> terminated ASCII strings. Data is sent and received in plain binary format.

4.4.1 USART ISP initialization

Once the USART ISP mode is entered, the auto-baud routine needs to synchronize with the host via the serial port (USART).

The host should send a ’?’ (0x3F) as a synchronization character and wait for a response. The host side serial port settings should be 8 data bits, 1 stop bit and no parity. The auto-baud routine measures the bit time of the received synchronization character in terms of its own frequency and programs the baud rate generator of the serial port. It also sends an ASCII string ("Synchronized<CR><LF>") to the host. In response to this, the host should send back the same string ("Synchronized<CR><LF>").

The auto-baud routine looks at the received characters to verify synchronization. If synchronization is verified then "OK<CR><LF>" string is sent to the host. The host should respond by sending the crystal frequency (in kHz) at which the part is running. The response is required for backward compatibility of the boot loader code and is ignored. "OK<CR><LF>" string is sent to the host after receiving the crystal frequency. If synchronization is not verified then the auto-baud routine waits again for a synchronization character.

Once the crystal frequency is received the part is initialized and the ISP command handler is invoked. For safety reasons an "Unlock" command is required before executing the commands resulting in flash erase/write operations and the "Go" command. The rest of the commands can be executed without the unlock command. The Unlock command is required to be executed once per ISP session. The Unlock command is explained in Section 4.5 “USART ISP commands”.

4.4.2 USART ISP command format

"Command Parameter_0 Parameter_1 ... Parameter_n<CR><LF>" "Data" (Data only for Write commands).

4.4.3 USART ISP response format

"Return_Code<CR><LF>Response_0<CR><LF>Response_1<CR><LF> ... Response_n<CR><LF>" "Data" (Data only for Read commands).

4.4.4 USART ISP data format

The data stream is in plain binary format.

4.5 USART ISP commands

The following commands are accepted by the ISP command handler. Detailed status codes are supported for each command. The command handler sends the return code INVALID_COMMAND when an undefined command is received. Commands and return codes are in ASCII format.

CMD_SUCCESS is sent by ISP command handler only when received ISP command has been completely executed and the new ISP command can be given by the host. Exceptions from this rule are "Set Baud Rate", "Write to RAM", "Read Memory", and "Go" commands.

Table 6. USART ISP command summary

ISP Command	Usage	Section
Unlock	U <Unlock Code>	4.5.1
Set Baud Rate	B <Baud Rate> <stop bit>	4.5.2
Echo	A <setting>	4.5.3
Write to RAM	W <start address> <number of bytes>	4.5.4
Read Memory	R <address> <number of bytes>	4.5.5
Prepare sectors for write operation	P <start sector number> <end sector number>	4.5.6
Copy RAM to flash	C <Flash address> <RAM address> <number of bytes>	4.5.7
Go	G <address> <Mode>	4.5.8
Erase sector(s)	E <start sector number> <end sector number>	4.5.9
Erase page(s)	X <start page number> <end page number>	4.5.10
Blank check sector(s)	I <start sector number> <end sector number>	4.5.11
Read Part ID	J	4.5.12
Read Boot code version	K	4.5.13
Compare	M <address1> <address2> <number of bytes>	4.5.14
ReadUID	N	4.5.15
Read CRC checksum	S <address> <number of bytes>	4.5.16
Read flash signature	Z	4.5.17

Table 7 lists the supported USART ISP commands for each CRP level.

Table 7. ISP commands allowed for different CRP levels

ISP command	CRP1	CRP2	CRP3 (no entry in ISP mode allowed)
Unlock	yes	yes	n/a
Set Baud Rate	yes	yes	n/a
Echo	yes	yes	n/a
Write to RAM	yes; above 0x1000 0600 only	no	n/a
Read Memory	no	no	n/a
Prepare sectors for write operation	yes	yes	n/a
Copy RAM to flash	yes; not to sector 0	no	n/a
Go	no	no	n/a
Erase sector(s)	yes; sector 0 can only be erased when all sectors are erased.	yes; all sectors only	n/a

Table 7. ISP commands allowed for different CRP levels

ISP command	CRP1	CRP2	CRP3 (no entry in ISP mode allowed)
Erase page(s)	yes; page 0 can only be erased when all pages are erased (not recommended, use Erase Sector).	yes; all pages only	n/a
Blank check sectors	yes; the offset and content fields are 0	no	n/a
Read Part ID	yes	yes	n/a
Read Boot code version	yes	yes	n/a
Compare	no	no	n/a
ReadUID	yes	yes	n/a
Read CRC	no	no	n/a
Read flash signature	yes (full range of the flash only)	no	n/a

4.5.1 Unlock

Table 8. USART ISP Unlock command

Command	U
Input	Unlock code: 23130_{10}
Return Code	CMD_SUCCESS | INVALID_CODE | PARAM_ERROR
Description	This command is used to unlock Flash Write, Erase, and Go commands.
Example	"U 23130<CR><LF>" unlocks the Flash Write/Erase & Go commands.

4.5.2 Set Baud Rate

Table 9. USART ISP Set Baud Rate command

Command	B
Input	Baud Rate: 9600 | 19200 | 38400 | 57600 | 115200 | 230400 | 460800 Stop bit: 1 | 2
Return Code	CMD_SUCCESS | INVALID_BAUD_RATE | INVALID_STOP_BIT | PARAM_ERROR
Description	This command is used to change the baud rate. The new baud rate is effective after the command handler sends the CMD_SUCCESS return code.
Example	"B 57600 1<CR><LF>" sets the serial port to baud rate 57600 bps and 1 stop bit.

4.5.3 Echo

Table 10. USART ISP Echo command

Command	A
Input	Setting: ON = 1 | OFF = 0

Table 10. USART ISP Echo command

Command	**A**
Return Code	CMD_SUCCESS | PARAM_ERROR
Description	The default setting for echo command is ON. When ON the ISP command handler sends the received serial data back to the host.
Example	"A 0<CR><LF>" turns echo off.

4.5.4 Write to RAM

The host should send the plain binary code after receiving the CMD_SUCCESS return code. This ISP command handler responds with “OK<CR><LF>” when the transfer has finished.

Table 11. USART ISP Write to RAM command

Command	**W**
Input	**Start Address:** RAM address where data bytes are to be written. This address should be a word boundary. **Number of Bytes:** Number of bytes to be written. Count should be a multiple of 4
Return Code	CMD_SUCCESS | ADDR_ERROR (Address not on word boundary) | ADDR_NOT_MAPPED | COUNT_ERROR (Byte count is not multiple of 4) | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to download data to RAM. This command is blocked when code read protection levels 2 or 3 are enabled. Writing to addresses below 0x1000 0600 is disabled for CRP1.
Example	"W 268437504 4<CR><LF>" writes 4 bytes of data to address 0x1000 0800.

4.5.5 Read Memory

Reads the plain binary code of the data stream, followed by the CMD_SUCCESS return code.

Table 12. USART ISP Read Memory command

Command	**R**
Input	**Start Address:** Address from where data bytes are to be read. This address should be a word boundary. **Number of Bytes:** Number of bytes to be read. Count should be a multiple of 4.
Return Code	CMD_SUCCESS followed by <actual data (plain binary)> | ADDR_ERROR (Address not on word boundary) | ADDR_NOT_MAPPED | COUNT_ERROR (Byte count is not a multiple of 4) | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to read data from RAM or flash memory. This command is blocked when code read protection is enabled.
Example	"R 268437504 4<CR><LF>" reads 4 bytes of data from address 0x1000 0800.

4.5.6 Prepare sectors for write operation

This command makes flash write/erase operation a two-step process.

Table 13. USART ISP Prepare sectors for write operation command

Command	**P**
Input	**Start Sector Number** **End Sector Number:** Should be greater than or equal to start sector number.
Return Code	CMD_SUCCESS | BUSY | INVALID_SECTOR | PARAM_ERROR
Description	This command must be executed before executing "Copy RAM to flash" or "Erase Sector(s)", or “Erase Pages” command. Successful execution of the "Copy RAM to flash" or "Erase Sector(s)" or “Erase Pages” command causes relevant sectors to be protected again. To prepare a single sector use the same "Start" and "End" sector numbers.
Example	"P 0 0<CR><LF>" prepares the flash sector 0.

4.5.7 Copy RAM to flash

When writing to the flash, the following limitation apply:

• The smallest amount of data that can be written to flash by the copy RAM to flash command is 64 byte (equal to one page).

Table 14. USART ISP Copy command

Command	**C**
Input	**Flash Address(DST):** Destination flash address where data bytes are to be written. The destination address should be a 64 byte boundary. **RAM Address(SRC):** Source RAM address from where data bytes are to be read. **Number of Bytes:** Number of bytes to be written. Should be 64 | 128 | 256 | 512 | 1024
Return Code	CMD_SUCCESS | SRC_ADDR_ERROR (Address not on word boundary) | DST_ADDR_ERROR (Address not on correct boundary) | SRC_ADDR_NOT_MAPPED | DST_ADDR_NOT_MAPPED | COUNT_ERROR (Byte count is not 64 | 128 | 256 | 512 | 1024) | SECTOR_NOT_PREPARED_FOR WRITE_OPERATION | BUSY | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to program the flash memory. The "Prepare Sector(s) for Write Operation" command should precede this command. The affected sectors are automatically protected again once the copy command is successfully executed. This command is blocked when code read protection is enabled. Also see Section 4.3.3 for the number of bytes that can be written.
Example	"C 0 268437504 512<CR><LF>" copies 512 bytes from the RAM address 0x1000 0800 to the flash address 0.

4.5.8 Go

Table 15. USART ISP Go command

Command	G
Input	**Address:** Flash or RAM address from which the code execution is to be started. This address should be on a word boundary. **Mode:** T (Execute program in Thumb Mode) |
Return Code	CMD_SUCCESS | ADDR_ERROR | ADDR_NOT_MAPPED | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to execute a program residing in RAM or flash memory. It may not be possible to return to the ISP command handler once this command is successfully executed. This command is blocked when code read protection is enabled.
Example	"G 0 T<CR><LF>" branches to address 0x0000 0000 in Thumb mode only.

4.5.9 Erase sectors

Table 16. USART ISP Erase sector command

Command	E
Input	**Start Sector Number** **End Sector Number:** Should be greater than or equal to start sector number.
Return Code	CMD_SUCCESS | BUSY | INVALID_SECTOR | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to erase one or more sector(s) of on-chip flash memory. This command only allows erasure of all user sectors when the code read protection is enabled.
Example	"E 2 3<CR><LF>" erases the flash sectors 2 and 3.

4.5.10 Erase pages

Table 17. USART ISP Erase page command

Command	X
Input	**Start Page Number** **End Page Number:** Should be greater than or equal to start page number.
Return Code	CMD_SUCCESS | BUSY | INVALID_PAGE | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | CMD_LOCKED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to erase one or more page(s) of on-chip flash memory.
Example	"X 2 3<CR><LF>" erases the flash pages 2 and 3.

4.5.11 Blank check sectors

Table 18. USART ISP Blank check sector command

Command	**I**
Input	**Start Sector Number:** **End Sector Number:** Should be greater than or equal to start sector number.
Return Code	CMD_SUCCESS | SECTOR_NOT_BLANK (followed by <Offset of the first non blank word location> <Contents of non blank word location>) | INVALID_SECTOR | PARAM_ERROR
Description	This command is used to blank check one or more sectors of on-chip flash memory.
Example	"I 2 3<CR><LF>" blank checks the flash sectors 2 and 3.

4.5.12 Read Part Identification number

Table 19. USART ISP Read Part Identification command

Command	**J**
Input	None.
Return Code	CMD_SUCCESS followed by part identification number (see Table 20).
Description	This command is used to read the part identification number.

Table 20. Device ID register values

Part number	Part ID
LPC865M201JBD64/00	0x00008651
LPC865M201JHI48/00	0x00008652
LPC865M201JHI33/00	0x00008654

4.5.13 Read Boot code version number

Table 21. USART ISP Read Boot Code version number command

Command	**K**
Input	None
Return Code	CMD_SUCCESS followed by 2 bytes of boot code version number in ASCII format. It is to be interpreted as <byte1(Major)>.<byte0(Minor)>.
Description	This command is used to read the boot code version number.

4.5.14 Compare

Table 22. USART ISP Compare command

Command	**M**
Input	**Address1 (DST):** Starting flash or RAM address of data bytes to be compared. This address should be a word boundary. **Address2 (SRC):** Starting flash or RAM address of data bytes to be compared. This address should be a word boundary. **Number of Bytes:** Number of bytes to be compared; should be a multiple of 4.
Return Code	CMD_SUCCESS | (Source and destination data are equal) | COMPARE_ERROR | (Followed by the offset of first mismatch) | COUNT_ERROR (Byte count is not a multiple of 4) | ADDR_ERROR | ADDR_NOT_MAPPED | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to compare the memory contents at two locations.
Example	"M 8192 268437504 4<CR><LF>" compares 4 bytes from the RAM address 0x1000 0800 to the 4 bytes from the flash address 0x2000.

4.5.15 ReadUID

Table 23. USART ReadUID command

Command	**N**
Input	None
Return Code	CMD_SUCCESS followed by four 32-bit words of a unique serial number in ASCII format. The word sent at the lowest address is sent first.
Description	This command is used to read the unique ID.

4.5.16 Read CRC checksum

Get the CRC checksum of a block of RAM or flash. CMD_SUCCESS followed by 8 bytes of CRC checksum in decimal format.

The checksum is calculated as follows:

CRC-32 polynomial: x32 + x26 + x23 + x22 + x16 + x12 + x11 + x10 + x8 + x7 + x5 + x4 + x2 + x + 1

Seed Value: 0xFFFF FFFF

Table 24. USART ISP Read CRC checksum command

Command	S
Input	**Address:** The data are read from this address for CRC checksum calculation. This address must be on a word boundary. **Number of Bytes:** Number of bytes to be calculated for the CRC checksum; must be a multiple of 4.
Return Code	CMD_SUCCESS followed by data in decimal format | ADDR_ERROR (address not on word boundary) | ADDR_NOT_MAPPED | COUNT_ERROR (byte count is not a multiple of 4) | PARAM_ERROR | CODE_READ_PROTECTION_ENABLED
Description	This command is used to read the CRC checksum of a block of RAM or flash memory. This command is blocked when code read protection is enabled.
Example	"S 33587200 4<CR><LF>" reads the CRC checksum for 4 bytes of data from address 0x0200 8000. If checksum value is 0xCBF43926, then the host will receive: "3421780262 <CR><LF>"

4.5.17 Read flash signature

Get the signature for the flash memory, using an internal flash signature generator (see Chapter 5 “Flash signature generator”). When CRP1 is enabled, only the signature of the entire flash can be read and no parameters can be passed.

Table 25. USART ISP Read flash signature command

Command	Z
Input	**Start address:** Start of flash address. Default = 0. Must be 0 when CRP1 is enabled. **End address:** End of flash address. Default = 0xFFFF Must be 0xFFFF when CRP1 is enabled. **Number of wait states:** Number of wait states. Default = 2. **Mode:** Flash controller mode must pass value 0.
Return Code	CMD_SUCCESS followed by 32 bit flash signature in decimal format | When CRP1 is enabled the signature is read for the entire flash | ADDR_NOT_MAPPED | PARAM_ERROR
Description	This command is used to read the flash signature generated by the flash controller.
Example	"Z<CR><LF>" reads the signature of the entire flash generated by the flash controller.

4.5.18 ISP/IAP Error codes

These error codes are located in the error.h file.

Table 26. ISP/IAP Error codes

Return Code	Error code	Description
0x0	CMD_SUCCESS	Command is executed successfully. Sent by ISP handler only when command given by the host has been completely and successfully executed.
0x1	INVALID_COMMAND	Invalid command.
0x2	SRC_ADDR_ERROR	Source address is not on word boundary.
0x3	DST_ADDR_ERROR	Destination address is not on a correct boundary.
0x4	SRC_ADDR_NOT_MAPPED	Source address is not mapped in the memory map. Count value is taken into consideration where applicable.
0x5	DST_ADDR_NOT_MAPPED	Destination address is not mapped in the memory map. Count value is taken into consideration where applicable.
0x6	COUNT_ERROR	Byte count is not multiple of 4 or is not a permitted value.
0x7	INVALID_SECTOR/INVALID_PAGE	Sector/page number is invalid or end sector number is greater than start sector number.
0x8	SECTOR_NOT_BLANK	Sector is not blank.
0x9	SECTOR_NOT_PREPARED_ FOR_WRITE_OPERATION	Command to prepare sector for write operation was not executed.
0xA	COMPARE_ERROR	Source and destination data not equal.
0xB	BUSY	Flash programming hardware interface is busy.
0xC	PARAM_ERROR	Insufficient number of parameters or invalid parameter.
0xD	ADDR_ERROR	Address is not on word boundary.
0xE	ADDR_NOT_MAPPED	Address is not mapped in the memory map. Count value is taken into consideration where applicable.
0xF	CMD_LOCKED	Command is locked.
0x10	INVALID_CODE	Unlock code is invalid.
0x11	INVALID_BAUD_RATE	Invalid baud rate setting.
0x12	INVALID_STOP_BIT	Invalid stop bit setting.
0x13	CODE_READ_ PROTECTION_ENABLED	Code read protection enabled.
0x14	-	Reserved.
0x15	USER_CODE_CHECKSUM	User code checksum is invalid.
0x16	-	Reserved.
0x17	EFRO_NO_POWER	FRO not turned on in the PDRUNCFG register.
0x18	FLASH_NO_POWER	Flash not turned on in the PDRUNCFG register.
0x19	-	Reserved.
0x1A	-	Reserved.
0x1B	FLASH_NO_CLOCK	Flash clock disabled in the AHBCLKCTRL register.
0x1C	REINVOKE_ISP_CONFIG	Reinvoke ISP not successful.
0x1D	NO_VALID_IMAGE	Invalid image

4.6 IAP commands

For in application programming the IAP routine should be called with a word pointer in register r0 pointing to memory (RAM) containing command code and parameters. The result of the IAP command is returned in the result table pointed to by register r1. The user can reuse the command table for result by passing the same pointer in registers r0 and r1. The parameter table should be big enough to hold all the results in case the number of results are more than number of parameters. Parameter passing is illustrated in the Figure 5.

The number of parameters and results vary according to the IAP command. The maximum number of parameters is 5, passed to the "Copy RAM to FLASH" command. The maximum number of results is 5, returned by the "ReadUID" command. The command handler sends the status code INVALID_COMMAND when an undefined command is received. The IAP routine resides at location 0x0F001D97 and it is thumb code, therefore called as 0x0F001D98 by the Cortex-M0+ to insure Thumb operation.

The IAP function could be called in the following way using C:

Define the IAP location entry point. Since the least significant bit of the IAP location is set there will be a change to Thumb instruction set if called by the Cortex-M0+.

Define data structure or pointers to pass IAP command table and result table to the IAP function:

```
unsigned int command_param[5];
unsigned int status_result[5];
```

or

```
unsigned int * command_param;
unsigned int * status_result;
command_param = (unsigned int *) 0x...
status_result =(unsigned int *) 0x...
```

Define pointer to function type, which takes two parameters and returns void. Note the IAP returns the result with the base address of the table residing in R1.

```
typedef void (*IAP)(unsigned int [],unsigned int[]);
IAP iap_entry;
```

Setting the function pointer:

```
#define IAP_LOCATION *(volatile unsigned int *)(0x0F001D98)
iap_entry=(IAP) IAP_LOCATION;
```

To call the IAP use the following statement.

```
iap_entry (command_param,status_result);
```

Up to 4 parameters can be passed in the r0, r1, r2 and r3 registers respectively (see the *ARM Thumb Procedure Call Standard SWS ESPC 0002 A-05)*. Additional parameters are passed on the stack. Up to 4 parameters can be returned in the r0, r1, r2 and r3 registers respectively. Additional parameters are returned indirectly via memory.

Before prepare/erase/program flash IAP API is called, FLASHCACHECFG register needs to be set to 0 first. The flash memory is not accessible during a write or erase operation.

Table 27. IAP Command Summary

IAP Command	Command code	Section
Prepare sector(s) for write operation	50 (decimal)	4.6.1
Copy RAM to flash	51 (decimal)	4.6.2
Erase sector(s)	52 (decimal)	4.6.3
Blank check sector(s)	53 (decimal)	4.6.4
Read Part ID	54 (decimal)	4.6.5
Read Boot code version	55 (decimal)	4.6.6
Compare	56 (decimal)	4.6.7
Reinvoke ISP	57 (decimal)	4.6.8
Read UID	58 (decimal)	4.6.9
Erase page(s)	59 (decimal)	4.6.10
Read Signature	73 (decimal)	4.6.11

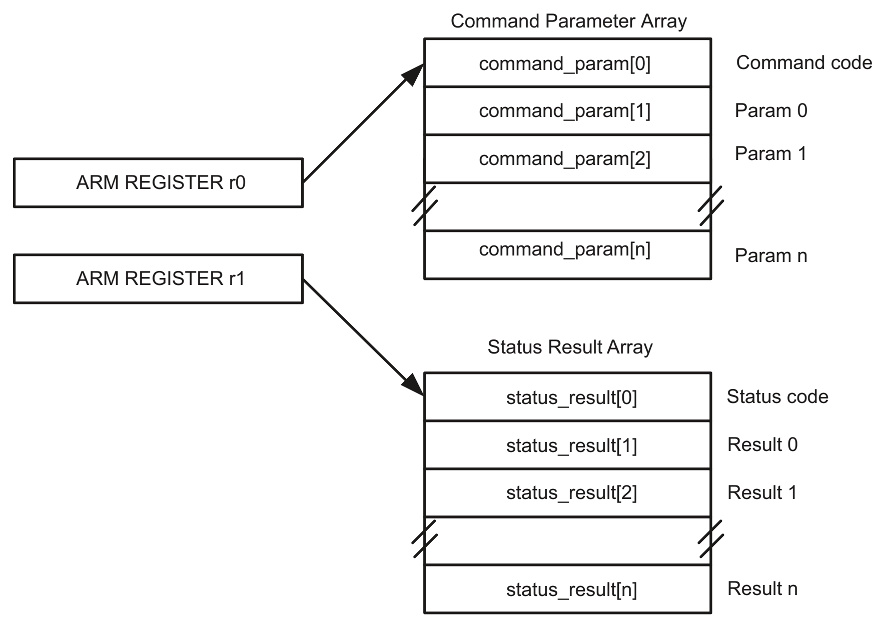


Fig 5. IAP parameter passing

4.6.1 Prepare sector(s) for write operation

This command makes flash write/erase operation a two step process.

Table 28. IAP Prepare sector(s) for write operation command

Command	Prepare sector(s) for write operation
Input	**Command code: 50 (decimal)** **Param0:** Start Sector Number **Param1:** End Sector Number (should be greater than or equal to start sector number).
Status code	CMD_SUCCESS | BUSY | INVALID_SECTOR
Result	None
Description	This command must be executed before executing "Copy RAM to flash" or "Erase Sector(s)" or “Erase page(s)” command. Successful execution of the "Copy RAM to flash" or "Erase Sector(s)" or “Erase page(s)” command causes relevant sectors to be protected again. To prepare a single sector use the same "Start" and "End" sector numbers.

4.6.2 Copy RAM to flash

See Section 4.5.7 for limitations on the write-to-flash process.

Table 29. IAP Copy RAM to flash command

Command	Copy RAM to flash
Input	**Command code: 51 (decimal)** **Param0(DST):** Destination flash address where data bytes are to be written. This address should be a 64 byte boundary. **Param1(SRC):** Source RAM address from which data bytes are to be read. This address should be a word boundary. **Param2:** Number of bytes to be written. Should be 64 | 128 | 256 | 512 | 1024. **Param3:** System Clock Frequency (CCLK) in kHz.
Status code	CMD_SUCCESS | SRC_ADDR_ERROR (Address not a word boundary) | DST_ADDR_ERROR (Address not on correct boundary) | SRC_ADDR_NOT_MAPPED | DST_ADDR_NOT_MAPPED | COUNT_ERROR (Byte count is not 64 | 128 | 256 | 512 | 1024) | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | BUSY
Result	None
Description	This command is used to program the flash memory. The affected sectors should be prepared first by calling "Prepare Sector for Write Operation" command. The affected sectors are automatically protected again once the copy command is successfully executed. Also see Section 4.3.3 for the number of bytes that can be written. **Remark:** All user code must be written in such a way that no master accesses the flash while this command is executed and the flash is programmed.

4.6.3 Erase Sector(s)

Table 30. IAP Erase Sector(s) command

Command	Erase Sector(s)
Input	**Command code: 52 (decimal)** **Param0:** Start Sector Number **Param1:** End Sector Number (should be greater than or equal to start sector number). **Param2:** System Clock Frequency (CCLK) in kHz.
Status code	CMD_SUCCESS | BUSY | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | INVALID_SECTOR
Result	None
Description	This command is used to erase a sector or multiple sectors of on-chip flash memory. To erase a single sector use the same "Start" and "End" sector numbers. **Remark:** All user code must be written in such a way that no master accesses the flash while this command is executed and the flash is erased.

4.6.4 Blank check sector(s)

Table 31. IAP Blank check sector(s) command

Command	Blank check sector(s)
Input	**Command code: 53 (decimal)** **Param0:** Start Sector Number **Param1:** End Sector Number (should be greater than or equal to start sector number).
Status code	CMD_SUCCESS | BUSY | SECTOR_NOT_BLANK | INVALID_SECTOR
Result	**Result0:** Offset of the first non blank word location if the status code is SECTOR_NOT_BLANK. **Result1:** Contents of non blank word location.
Description	This command is used to blank check a sector or multiple sectors of on-chip flash memory. To blank check a single sector use the same "Start" and "End" sector numbers.

4.6.5 Read Part Identification number

Table 32. IAP Read Part Identification command

Command	Read part identification number
Input	**Command code: 54 (decimal)** **Parameters:** None
Status code	CMD_SUCCESS
Result	**Result0:** Part Identification Number.
Description	This command is used to read the part identification number.

4.6.6 Read Boot code version number

Table 33. IAP Read Boot Code version number command

Command	**Read boot code version number**
Input	**Command code: 55 (decimal)** **Parameters:** None
Status code	CMD_SUCCESS
Result	**Result0:** 2 bytes of boot code version number. Read as <byte1(Major)>.<byte0(Minor)>
Description	This command is used to read the boot code version number.

4.6.7 Compare <address1> <address2> <no of bytes>

Table 34. IAP Compare command

Command	**Compare**
Input	**Command code: 56 (decimal)** **Param0(DST):** Starting flash or RAM address of data bytes to be compared; should be a word boundary. **Param1(SRC):** Starting flash or RAM address of data bytes to be compared; should be a word boundary. **Param2:** Number of bytes to be compared; should be a multiple of 4.
Status code	CMD_SUCCESS | COMPARE_ERROR | COUNT_ERROR (Byte count is not a multiple of 4) | ADDR_ERROR | ADDR_NOT_MAPPED
Result	**Result0:** Offset of the first mismatch if the status code is COMPARE_ERROR.
Description	This command is used to compare the memory contents at two locations.

4.6.8 Reinvoke ISP

Table 35. Reinvoke ISP

Command	**Compare**
Input	**Command code: 57 (decimal)**
Status code	ERR_ISP_REINVOKE_ISP_CONFIG
Result	None.
Description	This command is used to invoke the ISP. If the ISP is invoked, then the CPU clock is switched to FRO 24 MHz. This command is used to invoke the boot loader in ISP mode. It maps boot vectors and configures the peripherals for ISP. This command may be used when a valid user program is present in the internal flash memory and the ISP entry pin are not accessible to force the ISP mode. ROM code will enable the clocks to default once "Reinvoke ISP" command is received.

4.6.9 ReadUID

Table 36. IAP ReadUID command

Command	Compare
Input	**Command code: 58 (decimal)**
Status code	CMD_SUCCESS
Result	**Result0:** The first 32-bit word (at the lowest address). **Result1:** The second 32-bit word. **Result2:** The third 32-bit word. **Result3:** The fourth 32-bit word.
Description	This command is used to read the unique ID.

4.6.10 Erase page

Table 37. IAP Erase page command

Command	Erase page
Input	**Command code: 59 (decimal)** **Param0:** Start page number. **Param1:** End page number (should be greater than or equal to start page) **Param2:** System Clock Frequency (CCLK) in kHz**.**
Status code	CMD_SUCCESS | BUSY | SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION | INVALID_PAGE
Result	None
Description	This command is used to erase a page or multiple pages of on-chip flash memory. To erase a single page use the same "start" and "end" page numbers. **Remark:** All user code must be written in such a way that no master accesses the flash while this command is executed and the flash is erased.

4.6.11 Read Signature

Table 38. IAP Read Signature command

Command	Read Signature
Input	**Command code: 73 (decimal)** **Param0:** Start flash address. **Param1:** End flash address. **Param2:** Number of wait states. **Param3:** Must be 0.
Status code	CMD_SUCCESS
Result	**Result0:** The 32-bit generated signature.
Description	This command is used to obtain a 32-bit signature value of the flash region. See Section 4.5.17 “Read flash signature” and Chapter 5 for more information. **Remark:** See Section 5.5.1.2 “Signature generation” to ensure that the flash signature is generated correctly. When CRP1 is enabled, only the signature of the entire flash can be read and no parameters can be passed.

4.6.12 IAP Error Codes

See Table 26 “ISP/IAP Error codes”.

UM11607

Chapter 5: Flash signature generator

Rev. 3 — April 2023 User manual

5.1 How to read this chapter

The flash signature generator is identical on all LPC86x parts.

5.2 Features

- Controls flash access time.
- Provides registers for flash signature generation.

5.3 General description

The flash signature generator is accessible for programming flash wait states and for generating the flash signature.

5.4 Register description

Table 39. Register overview: FMC (base address 0x4004 0000)

Name	Access	Address offset	Description	Reset value	Reference
FLASHCFG	R/W	0x010	Flash configuration register	-	Section 5.4.1
FMSSTART	R/W	0x020	Signature start address register	0	Section 5.4.2
FMSSTOP	R/W	0x024	Signature stop-address register	0	Section 5.4.3
FMSW0	R	0x02C	Signature word	-	Section 5.4.4
FMSTAT	R	0xFE0	Signature generation status register	0	Section 5.4.5
FMSTATCLR	W	0xFE8	Signature generation status clear register.	-	Section 5.4.6

5.4.1 Flash configuration register

Access time to the flash memory can be configured independently of the system frequency by writing to the FLASHCFG register.

Table 40. Flash configuration register (FLASHCFG, address 0x4004 0010) bit description

Bit	Symbol	Value	Description	Reset value
1:0	FLASHTIM		Flash memory access time. FLASHTIM +1 is equal to the number of system clocks used for flash access.	0x2
		0x0	1 system clock flash access time. (for system clock rates up to 24 MHz).	
		0x1	2 system clocks flash access time. (for system clock rates up to 48 MHz).	
		0x2	3 system clocks flash access time. (for system clock rates up to 60 MHz).	
		0x3	Reserved.	
31:2	-	-	Reserved. **User software must not change the value of these bits. Bits 31:2 must be written back exactly as read**.	-

5.4.2 Flash signature start address register

Table 41. Flash Module Signature Start register (FMSSTART, 0x4004 0020) bit description

Bit	Symbol	Description	Reset value
16:0	START	Signature generation start address (corresponds to AHB byte address bits[18:2]).	0
31:17	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

5.4.3 Flash signature stop address register

Table 42. Flash Module Signature Stop register (FMSSTOP, 0x4004 0024) bit description

Bit	Symbol	Value	Description	Reset value
16:0	STOPA		Stop address for signature generation (the word specified by STOPA is included in the address range). The address is in units of memory words, not bytes.	0
30:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	0
31	STRTBIST		When this bit is written to 1, signature generation starts. At the end of signature generation, this bit is automatically cleared.	0

5.4.4 Flash signature generation result register

The signature generation result register returns the flash signature produced by the embedded signature generator.

The generated flash signature can be used to verify the flash memory contents. The generated signature can be compared with an expected signature and thus makes saves time and code space. The method for generating the signature is described in Section 5.5.1.

Table 43. FMSW0 register bit description (FMSW0, address: 0x4004 002C)

Bit	Symbol	Description	Reset value
31:0	SIG	32-bit signature.	-

5.4.5 Flash module signature status register

The read-only FMSTAT register provides a means of determining when signature generation has completed. Completion of signature generation can be checked by polling the SIG_DONE bit in FMSTAT. SIG_DONE should be cleared via the FMSTATCLR register before starting a signature generation operation, otherwise the status might indicate completion of a previous operation.

Table 44. Flash module signature status register (FMSTAT, offset 0x0FE0) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved. Read value is undefined, only zero should be written.	NA
1	SIG_DONE	When 1, a previously started signature generation has completed. See FMSTATCLR register description for clearing this flag.	0
31:2	-	Reserved. Read value is undefined, only zero should be written.	NA

5.4.6 Flash module signature status clear register

The FMSTATCLR register is used to clear the signature generation completion flag.

Table 45. Flash module signature status clear register (FMSTATCLR, offset 0x0FE8) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved. Read value is undefined, only zero should be written.	NA
1	SIG_DONE_CLR	Writing a 1 to this bits clears the signature generation completion flag (SIG_DONE) in the FMSTAT register.	0
31:2	-	Reserved. Read value is undefined, only zero should be written.	NA

5.5 Functional description

5.5.1 Flash signature generation

The flash module contains a built-in signature generator. This generator can produce a 32-bit signature from a range of flash memory. A typical usage is to verify the flashed contents against a calculated signature (e.g. during programming).

The address range for generating a signature must be aligned on flash-word boundaries, that is, 32-bit boundaries. Once started, signature generation completes independently. While signature generation is in progress, the flash memory cannot be accessed for other purposes, and an attempted read will cause a wait state to be asserted until signature generation is complete. Code outside of the flash (e.g. internal RAM) can be executed during signature generation. This can include interrupt services, if the interrupt vector table is re-mapped to memory other than the flash memory. The code that initiates signature generation should also be placed outside of the flash memory.

5.5.1.1 Signature generation address and control registers

These registers control automatic signature generation. A signature can be generated for any part of the flash memory contents. The address range to be used for generation is defined by writing the start address to the signature start address register (FMSSTART) and the stop address to the signature stop address register (FMSSTOP. The start and stop addresses must be aligned to 32-bit boundaries.

Signature generation is started by setting the STRTBIST bit in the FMSSTOP register. Setting the STRTBIST bit is typically combined with the signature stop address in a single write.

Table 41 and Table 42 show the bit assignments in the FMSSTART and FMSSTOP registers respectively.

5.5.1.2 Signature generation

A signature can be generated for any part of the flash contents. The address range to be used for signature generation is defined by writing the start address to the FMSSTART register, and the stop address to the FMSSTOP register.

The signature generation is started by writing a 1 to the SIG_START bit in the FMSSTOP register. Starting the signature generation is typically combined with defining the stop address, which is done in the STOP bits of the same register.

The time that the signature generation takes is proportional to the address range for which the signature is generated. Reading of the flash memory for signature generation uses a self-timed read mechanism and does not depend on any configurable timing settings for the flash. A safe estimation for the duration of the signature generation is:

Duration = int((60 / tcy) + 3) x (FMSSTOP - FMSSTART + 1)

When signature generation is triggered via software, the duration is in AHB clock cycles, and tcy is the time in ns for one AHB clock. The SIG_DONE bit in FMSTAT can be polled by software to determine when signature generation is complete.

After signature generation, a 32-bit signature can be read from the FMSW0 register. The 32-bit signature reflects the corrected data read from the flash and the flash parity bits and check bit values.

5.5.1.3 Content verification

The signature as it is read from the FMSW0 register must be equal to the reference signature. The following pseudo-code shows the algorithm to derive the reference signature:

```
sign = 0
FOR address = FMSSTART.START to FMSSTOP.STOPA
{
      FOR i = 0 TO 30
      {
            nextSign[i] = f_Q[address][i] XOR sign[i + 1]
      }
      nextSign[31] = f_Q[address][31] XOR sign[0] XOR sign[10] XOR sign[30] XOR
      sign[31]
      sign = nextSign
}
signature32 = sign
```

Chapter 6: Nested Vectored Interrupt Controller (NVIC)

6.1 How to read this chapter

The NVIC is identical on all LPC86x parts.

6.2 Features

- Nested Vectored Interrupt Controller that is an integral part of the ARM Cortex-M0+.
- Tightly coupled interrupt controller provides low interrupt latency.
- Controls system exceptions and peripheral interrupts.
- The NVIC supports 32 vectored interrupts.
- Four programmable interrupt priority levels with hardware priority level masking.
- Software interrupt generation using the ARM exceptions SVCall and PendSV (see Ref. 3).
- Support for NMI.
- ARM Cortex-M0+ Vector table offset register VTOR implemented.

6.3 General description

The Nested Vectored Interrupt Controller (NVIC) is an integral part of the Cortex-M0+. The tight coupling to the CPU allows for low interrupt latency and efficient processing of late arriving interrupts.

6.3.1 Interrupt sources

Table 46 lists the interrupt sources for each peripheral function. Each peripheral device may have one or more interrupt lines to the Vectored Interrupt Controller. Each line may represent more than one interrupt source. Interrupts with the same priority level are serviced in the order of their interrupt number.

See Ref. 3 for a detailed description of the NVIC and the NVIC register description.

Table 46. Connection of interrupt sources to the NVIC

Interrupt number	Vector address	Name	Description	Flags
0	040	SPI0_IRQ	SPI0 interrupt	See Table 282 “SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI0), 0x4005 C00C (SPI1)) bit description”.
1	044	SPI1_IRQ	SPI1 interrupt	Same as SPI0_IRQ
2	048	-	Reserved	-
3	04C	UART0_IRQ	USART0 interrupt	See Table 268 “USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1), 0x4006C00C (USART2)) bit description”

Table 46. Connection of interrupt sources to the NVIC

Interrupt number	Vector address	Name	Description	Flags
4	050	UART1_IRQ	USART1 interrupt	Same as UART0_IRQ
5	054	UART2_IRQ	USART2 interrupt	Same as UART0_IRQ
6	058	FTM0_IRQ	FlexTimer0 interrupt	-
7	05C	FTM1_IRQ	FlexTimer1 interrupt	-
8	060	I2C0_IRQ	I2C0 interrupt	See Table 298 "Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0)) bit description".
9	064	-	Reserved	-
10	068	MRT_IRQ	Multi-rate timer interrupt	Global MRT interrupt. GFLAG0 GFLAG1 GFLAG2 GFLAG3
11	06C	ACMP_IRQ	Analog comparator interrupt	COMPEDGE - rising, falling, or both edges can set the bit.
12	070	WDT_IRQ	Windowed watchdog timer interrupt	WARNINT - watchdog warning interrupt
13	074	BOD_IRQ	BOD interrupt	BODINTVAL - BOD interrupt level
14	078	FLASH_IRQ	Flash interrupt	-
15	07C	WKT_IRQ	Self-wake-up timer interrupt	ALARMFLAG
16	080	ADC_SEQA_IRQ	ADC sequence A completion interrupt	-
17	084	ADC_SEQB_IRQ	ADC sequence B completion interrupt	-
18	088	ADC_THCMP_IRQ	ADC threshold compare interrupt	-
19	08C	ADC_OVR_IRQ	ADC overrun interrupt	-
20	090	DMA_IRQ	DMA0 controller interrupt	-
21	094	I3C0_IRQ	I3C interface 0 interrupt	-
22	098	GPIO_HS_IRQ0	GPIO group A interrupt	-.
23	09C	GPIO_HS_IRQ1	GPIO group B interrupt	-
24	0A0	PININT0_IRQ	Pin interrupt 0 or pattern match engine slice 0 interrupt	PSTAT - pin interrupt status
25	0A4	PININT1_IRQ	Pin interrupt 1 or pattern match engine slice 1 interrupt	PSTAT - pin interrupt status
26	0A8	PININT2_IRQ	Pin interrupt 2 or pattern match engine slice 2 interrupt	PSTAT - pin interrupt status
27	0AC	PININT3_IRQ	Pin interrupt 3 or pattern match engine slice 3 interrupt	PSTAT - pin interrupt status

Table 46. Connection of interrupt sources to the NVIC

Interrupt number	Vector address	Name	Description	Flags
28	0B0	PININT4_IRQ	Pin interrupt 4 or pattern match engine slice 4 interrupt	PSTAT - pin interrupt status
29	0B4	PININT5_IRQ	Pin interrupt 5 or pattern match engine slice 5 interrupt	PSTAT - pin interrupt status
30	0B8	PININT6_IRQ	Pin interrupt 6 or pattern match engine slice 6 interrupt	PSTAT - pin interrupt status
31	0BC	PININT7_IRQ	Pin interrupt 7 or pattern match engine slice 7 interrupt	PSTAT - pin interrupt status

6.3.2 Non-Maskable Interrupt (NMI)

The part supports the NMI, which can be triggered by an peripheral interrupt or triggered by software. The NMI has the highest priority exception other than the reset.

You can set up any peripheral interrupt listed in Table 46 as NMI using the NMISRC register in the SYSCON block (Table 112). To avoid using the same peripheral interrupt as NMI exception and normal interrupt, disable the interrupt in the NVIC when you configure it as NMI.

6.3.3 Vector table offset

The vector table contains the reset value of the stack pointer and the start addresses, also called exception vectors, for all exception handlers. On system reset, the vector table is located at address 0x0000 0000. Software can write to the VTOR register in the NVIC to relocate the vector table start address to a different memory location. For a description of the VTOR register, see the ARM Cortex-M0+ documentation (Ref. 3).

6.4 Register description

The NVIC registers are located on the ARM private peripheral bus.

Table 47. Register overview: NVIC (base address 0xE000 E000)

Name	Access	Address offset	Description	Reset value	Reference
ISER0	RW	0x100	Interrupt Set Enable Register 0. This register allows enabling interrupts and reading back the interrupt enables for specific peripheral functions.	0	Table 48
-	-	0x104	Reserved.	-	-
ICER0	RW	0x180	Interrupt Clear Enable Register 0. This register allows disabling interrupts and reading back the interrupt enables for specific peripheral functions.	0	Table 49
-	-	0x184	Reserved.	0	-
ISPR0	RW	0x200	Interrupt Set Pending Register 0. This register allows changing the interrupt state to pending and reading back the interrupt pending state for specific peripheral functions.	0	Table 50
-	-	0x204	Reserved.	0	-
ICPR0	RW	0x280	Interrupt Clear Pending Register 0. This register allows changing the interrupt state to not pending and reading back the interrupt pending state for specific peripheral functions.	0	Table 51
-	-	0x284	Reserved.	0	-
-	-	0x304	Reserved.	0	-
IPR0	RW	0x400	Interrupt Priority Registers 0. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 0 to 3.	0	Table 52
IPR1	RW	0x404	Interrupt Priority Registers 1 This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 4 to 7.	0	Table 53
IPR2	RW	0x408	Interrupt Priority Registers 2. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 8 to 11.	0	Table 54
IPR3	RW	0x40C	Interrupt Priority Registers 3. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 12 to 15.	0	Table 55
IPR4	RW	0x410	Interrupt Priority Registers 4. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 16 to 19.	0	Table 56
IPR5	RW	0x414	Interrupt Priority Registers 5. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 20 to 23.	0	Table 57
IPR6	RW	0x418	Interrupt Priority Registers 6. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 24 to 27.	0	Table 58
IPR7	RW	0x41C	Interrupt Priority Registers 7. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 28 to 31.	0	Table 59

6.4.1 Interrupt Set Enable Register 0 register

The ISER0 register allows to enable peripheral interrupts or to read the enabled state of those interrupts. Disable interrupts through the ICER0 (Section 6.4.2).

The bit description is as follows for all bits in this register:

Write — Writing 0 has no effect, writing 1 enables the interrupt.

Read — 0 indicates that the interrupt is disabled, 1 indicates that the interrupt is enabled.

Table 48. Interrupt Set Enable Register 0 register (ISER0, address 0xE000 E100) bit description

Bit	Symbol	Description	Reset value
0	ISE_SPI0	Interrupt enable.	0
1	ISE_SPI1	Interrupt enable.	0
2	-	Reserved.	0
3	ISE_UART0	Interrupt enable.	0
4	ISE_UART1	Interrupt enable.	0
5	ISE_UART2	Interrupt enable.	0
6	ISE_FTM0	Interrupt enable.	0
7	ISE_FTM1	Interrupt enable.	0
8	ISE_I2C0	Interrupt enable.	0
9	-	Reserved.	0
10	ISE_MRT	Interrupt enable.	0
11	ISE_ACMP	Interrupt enable.	0
12	ISE_WDT	Interrupt enable.	0
13	ISE_BOD	Interrupt enable.	0
14	ISE_FLASH	Interrupt enable.	0
15	ISE_WKT	Interrupt enable.	0
16	ISE_ADC_SEQA	Interrupt enable.	0
17	ISE_ADC_SEQB	Interrupt enable.	0
18	ISE_ADC_THCMP	Interrupt enable.	0
19	ISE_ADC_OVR	Interrupt enable.	0
20	ISE_DMA0	Interrupt enable.	0
21	ISE_I3C0	Interrupt enable.	0
22	ISE_GPIO_HS_IRQ0	Interrupt enable.	0
23	ISE_GPIO_HS_IRQ1	Interrupt enable.	0
24	ISE_PININT0	Interrupt enable.	0
25	ISE_PININT1	Interrupt enable.	0
26	ISE_PININT2	Interrupt enable.	0
27	ISE_PININT3	Interrupt enable.	0
28	ISE_PININT4	Interrupt enable.	0
29	ISE_PININT5	Interrupt enable.	0
30	ISE_PININT6	Interrupt enable.	0
31	ISE_PININT7	Interrupt enable.	0

6.4.2 Interrupt clear enable register 0

The ICER0 register allows disabling the peripheral interrupts, or for reading the enabled state of those interrupts. Enable interrupts through the ISER0 registers (Section 6.4.1).

The bit description is as follows for all bits in this register:

Write — Writing 0 has no effect, writing 1 disables the interrupt.

Read — 0 indicates that the interrupt is disabled, 1 indicates that the interrupt is enabled.

Table 49. Interrupt clear enable register 0 (ICER0, address 0xE000 E180)

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt disable.	0
1	ICE_SPI1	Interrupt disable.	0
2	-	Reserved.	0
3	ICE_UART0	Interrupt disable.	0
4	ICE_UART1	Interrupt disable.	0
5	ICE_UART2	Interrupt disable.	0
6	ICE_FTM0	Interrupt disable.	0
7	ICE_FTM1	Interrupt disable.	0
8	ICE_I2C0	Interrupt disable.	0
9	-	Reserved.	0
10	ICE_MRT	Interrupt disable.	0
11	ICE_ACMP	Interrupt disable.	ISE_ACMP
12	ICE_WDT	Interrupt disable.	0
13	ICE_BOD	Interrupt disable.	0
14	ICE_FLASH	Interrupt disable.	0
15	ICE_WKT	Interrupt disable.	0
16	ICE_ADC_SEQA	Interrupt disable.	0
17	ICE_ADC_SEQB	Interrupt disable.	0
18	ICE_ADC_THCMP	Interrupt disable.	0
19	ICE_ADC_OVR	Interrupt disable.	0
20	ICE_DMA0	Interrupt disable.	0
21	ICE_I3C0	Interrupt disable.	0
22	ICE_GPIO_HS_IRQ0	Interrupt disable.	0
23	ICE_GPIO_HS_IRQ1	Interrupt disable.	0
24	ICE_PININT0	Interrupt disable.	0
25	ICE_PININT1	Interrupt disable.	0
26	ICE_PININT2	Interrupt disable.	0
27	ICE_PININT3	Interrupt disable.	0
28	ICE_PININT4	Interrupt disable.	0
29	ICE_PININT5	Interrupt disable.	0
30	ICE_PININT6	Interrupt disable.	0
31	ICE_PININT7	Interrupt disable.	0

6.4.3 Interrupt Set Pending Register 0 register

The ISPR0 register allows setting the pending state of the peripheral interrupts, or for reading the pending state of those interrupts. Clear the pending state of interrupts through the ICPR0 registers (Section 6.4.4).

The bit description is as follows for all bits in this register:

Write — Writing 0 has no effect, writing 1 changes the interrupt state to pending.

Read — 0 indicates that the interrupt is not pending, 1 indicates that the interrupt is pending.

Table 50. Interrupt set pending register 0 register (ISPR0, address 0xE000 E200) bit description

Bit	Symbol	Description	Reset value
0	ISP_SPI0	Interrupt pending set.	0
1	ISP_SPI1	Interrupt pending set.	0
2	-	Reserved	0
3	ISP_UART0	Interrupt pending set.	0
4	ISP_UART1	Interrupt pending set.	0
5	ISP_UART2	Interrupt pending set.	0
6	ISP_FTM0	Interrupt pending set.	0
7	ISP_FTM1	Interrupt pending set.	0
8	ISP_I2C0	Interrupt pending set.	0
9	-	Reserved	0
10	ISP_MRT	Interrupt pending set.	0
11	ISP_ACMP	Interrupt pending set.	0
12	ISP_WDT	Interrupt pending set.	0
13	ISP_BOD	Interrupt pending set.	0
14	ISP_FLASH	Interrupt pending set.	0
15	ISP_WKT	Interrupt pending set.	0
16	ISP_ADC_SEQA	Interrupt pending set.	0
17	ISP_ADC_SEQB	Interrupt pending set.	0
18	ISP_ADC_THCMP	Interrupt pending set.	0
19	ISP_ADC_OVR	Interrupt pending set.	0
20	ISP_SDMA	Interrupt pending set.	0
21	ISP_I3C0	Interrupt pending set.	0
22	ISP_GPIO_HS_IRQ0	Interrupt pending set.	0
23	ISP_GPIO_HS_IRQ1	Interrupt pending set.	0
24	ISP_PININT0	Interrupt pending set.	0
25	ISP_PININT1	Interrupt pending set.	0
26	ISP_PININT2	Interrupt pending set.	0
27	ISP_PININT3	Interrupt pending set.	0
28	ISP_PININT4	Interrupt pending set.	0
29	ISP_PININT5	Interrupt pending set for pinint5.	0
30	ISP_PININT6	Interrupt pending set for pinint6.	0
31	ISP_PININT7	Interrupt pending set for pinint7.	0

6.4.4 Interrupt Clear Pending Register 0 register

The ICPR0 register allows clearing the pending state of the peripheral interrupts, or for reading the pending state of those interrupts. Set the pending state of interrupts through the ISPR0 register (Section 6.4.3).

The bit description is as follows for all bits in this register:

Write — Writing 0 has no effect, writing 1 changes the interrupt state to not pending.

Read — 0 indicates that the interrupt is not pending, 1 indicates that the interrupt is pending.

Table 51. Interrupt clear pending register 0 register (ICPR0, address 0xE000 E280) bit description

Bit	Symbol	Function	Reset value
0	ICP_SPI0	Interrupt pending clear.	0
1	ICP_SPI1	Interrupt pending clear.	0
2	-	Reserved	0
3	ICP_UART0	Interrupt pending clear.	0
4	ICP_UART1	Interrupt pending clear.	0
5	ICP_UART2	Interrupt pending clear.	0
6	ICP_FTM0	Reserved	0
7	ICP_FTM1	Interrupt pending clear.	0
8	ICP_I2C0	Interrupt pending clear.	0
9	-	Reserved	0
10	ICP_MRT	Interrupt pending clear.	0
11	ICP_ACMP	Interrupt pending clear.	0
12	ICP_WDT	Interrupt pending clear.	0
13	ICP_BOD	Interrupt pending clear.	0
14	ICP_FLASH	Interrupt pending clear.	0
15	ICP_WKT	Interrupt pending clear.	0
16	ICP_ADC_SEQA	Interrupt pending clear.	0
17	ICP_ADC_SEQB	Interrupt pending clear.	0
18	ICP_ADC_THCMP	Interrupt pending clear.	0
19	ICP_ADC_OVR	Interrupt pending clear.	0
20	ICP_SDMA	Interrupt pending clear.	0
21	ICP_I3C0	Interrupt pending clear.	0
22	ICP_GPIO_HS_IRQ0	Interrupt pending clear.	0
23	ICP_GPIO_HS_IRQ1	Interrupt pending clear.	0
24	ICP_PININT0	Interrupt pending clear.	0
25	ICP_PININT1	Interrupt pending clear.	0
26	ICP_PININT2	Interrupt pending clear.	0
27	ICP_PININT3	Interrupt pending clear.	0
28	ICP_PININT4	Interrupt pending clear.	0
29	ICP_PININT5	Interrupt pending clear.	0
30	ICP_PININT6	Interrupt pending clear.	0
31	ICP_PININT7	Interrupt pending clear.	0

6.4.5 Interrupt Priority Register 0

The IPR0 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 52. Interrupt Priority Register 0 (IPR0, address 0xE000 E400) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_SPI0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	IP_SPI1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	-	Reserved
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_UART0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.6 Interrupt Priority Register 1

The IPR1 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 53. Interrupt Priority Register 1 (IPR1, address 0xE000 E404) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_UART1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	IP_UART2	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_FTM0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_FTM1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.7 Interrupt Priority Register 2

The IPR2 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 54. Interrupt Priority Register 2 (IPR2, address 0xE000 E408) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_I2C0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	-	Reserved.
21:16	-	These bits ignore writes, and read as 0.

Table 54. Interrupt Priority Register 2 (IPR2, address 0xE000 E408) *…continued***bit description**

Bit	Symbol	Description
23:22	IP_MRT	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_ACMP	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.8 Interrupt Priority Register 3

The IPR3 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 55. Interrupt Priority Register 3 (IPR3, address 0xE000 E40C) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_WDT	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	IP_BOD	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_FLASH	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_WKT	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.9 Interrupt Priority Register 4

The IPR3 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 56. Interrupt Priority Register 4 (IPR4, address 0xE000 E410) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_ADC_SEQA	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	IP_ADC_SEQB	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_ADC_THCMP	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_ADC_OVR	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.10 Interrupt Priority Register 5

The IPR3 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 57. Interrupt Priority Register 5 (IPR5, address 0xE000 E414) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_DMA	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.

Table 57. Interrupt Priority Register 5 (IPR5, address 0xE000 E414) *…continued* **bit description**

Bit	Symbol	Description
15:14	IP_I3C0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_GPIO_HS_IRQ0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	Reserved.
31:30	IP_GPIO_HS_IRQ1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.11 Interrupt Priority Register 6

The IPR6 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 58. Interrupt Priority Register 6 (IPR6, address 0xE000 E418) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_PININT0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	IP_PININT1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_PININT2	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_PININT3	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

6.4.12 Interrupt Priority Register 7

The IPR7 register controls the priority of four peripheral interrupts. Each interrupt can have one of 4 priorities, where 0 is the highest priority.

Table 59. Interrupt Priority Register 7 (IPR7, address 0xE000 E41C) bit description

Bit	Symbol	Description
5:0	-	These bits ignore writes, and read as 0.
7:6	IP_PININT4	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
13:8	-	These bits ignore writes, and read as 0.
15:14	IP_PININT5	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_PININT6	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_PININT7	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

Chapter 7: Peripheral Input Multiplexing Connections

7.1 How to read this chapter

This chapter lists the peripheral input multiplexing connections on this device.

7.2 FlexTimer peripheral input multiplexing connections

Note: FlexTimer trigger interrupt does not support automatic clear function, the related interrupt service routine code is used to explicitly clear it.

The following figure describes the FTM, ADC, and DMA interconnection.

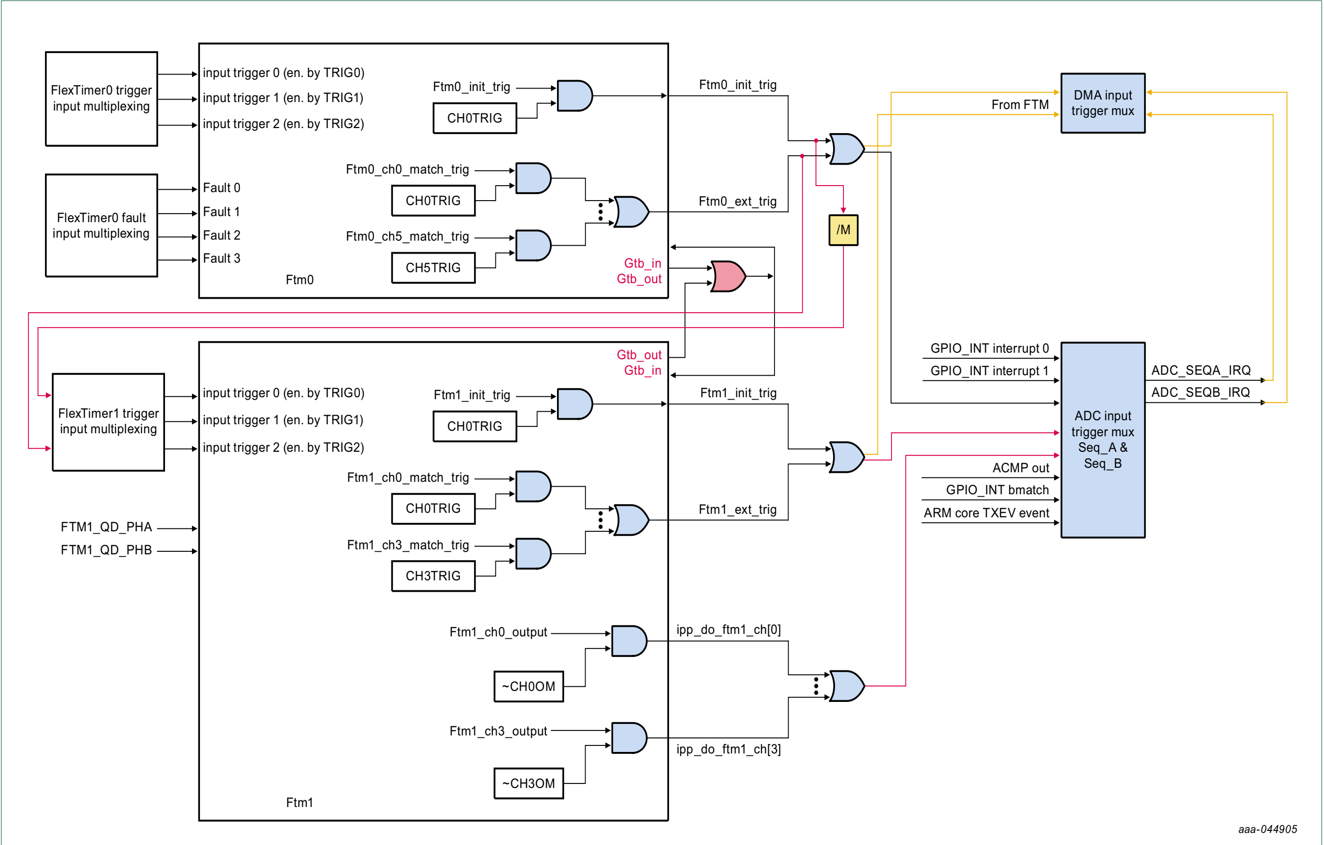


Fig 6. FTM, ADC, and DMA interconnections

Note: See Section 8.6.27 “FTM0INTTRIGDIV” for the meaning of divider /M

7.2.1 FlexTimer0 trigger inputs

FlexTimer0 trigger inputs are selected by FTM0_INMUX0/1/2 registers described in Chapter 15 “Input multiplexing and DMA trigger multiplexing (INPUT MUX, DMA TRIGMUX)”

Table 60. FlexTimer0 trigger input source selection

Selection	Input source
0	FTM1_INIT_TRIG
1	FTM1_EXT_TRIG
2	ADC0_THCMP_IRQ
3	ACMP0_OUT
4	GPIOINT_BMATCH
5	ARM_TXEV
6	MRT_IRQ (global MRT interrupt)

7.2.2 FlexTimer1 trigger inputs

FlexTimer1 trigger inputs are selected by FTM0_INMUX0/1/2 registers described in Chapter 15 “Input multiplexing and DMA trigger multiplexing (INPUT MUX, DMA TRIGMUX)”

Table 61. FlexTimer1 trigger input source selection

Selection	Input source
0	FTM0_INIT_TRIG divider (divided by M)
1	FTM0_EXT_TRIG
2	ADC0_THCMP_IRQ
3	ACMP0_OUT
4	GPIOINT_BMATCH
5	ARM_TXEV
6	MRT_IRQ (global MRT interrupt)

7.2.3 FlexTimer0 fault inputs

The FlexTimer fault input source selection is controlled by FTM fault configuration register and FTM0_FLT_INMUX0/1/2 register. FTM fault configuration register is described in Section 8.6.28 “FTM fault configuration register”.

The FTM0_FLT_INMUX0/1/2 register is described in Chapter 15 “Input multiplexing and DMA trigger multiplexing (INPUT MUX, DMA TRIGMUX)”.

Table 62. FlexTimer0 fault input source selection

Selection	Input source
0	FTM1_INIT_TRIG
1	FTM1_EXT_TRIG
2	ADC0_THCMP_IRQ
3	ACMP0_OUT
4	GPIOINT_BMATCH
5	ARM_TXEV
6	DEBUG_HALTED
7	Software fault control bit in SysCON
8	FTM0_FAULT n pin function (specific fault n pin for corresponding FAULT n input)

7.3 GPIO_INT peripheral input multiplexing connections

Table 63. GPIO_INT peripheral input multiplexing connections

Selection	Input source
0	PIO0_0
1	PIO0_1
2	PIO0_2
3	PIO0_3
4	PIO0_4
5	PIO0_5
6	PIO0_6
7	PIO0_7
8	PIO0_8
9	PIO0_9
10	PIO0_10
11	PIO0_11
12	PIO0_12
13	PIO0_13
14	PIO0_14
15	PIO0_15
16	PIO0_16
17	PIO0_17
18	PIO0_18
19	PIO0_19
20	PIO0_20
21	PIO0_21
22	PIO0_22
23	PIO0_23
24	PIO0_24
25	PIO0_25
26	PIO0_26
27	PIO0_27
28	PIO0_28
29	PIO0_29
30	PIO0_30
31	PIO0_31
32	PIO1_0
33	PIO1_1
34	PIO1_2
35	PIO1_3
36	PIO1_4
37	PIO1_5
38	PIO1_6

Table 63. GPIO_INT peripheral input multiplexing connections

Selection	Input source
39	PIO1_7
40	PIO1_8
41	PIO1_9
42	PIO1_10
43	PIO1_11
44	PIO1_12
45	PIO1_13
46	PIO1_14
47	PIO1_15
48	PIO1_16
49	PIO1_17
50	PIO1_18
51	PIO1_19
52	PIO1_20
53	PIO1_21

7.4 ADC trigger inputs

Table 64. ADC trigger inputs

Selection	Input source
0	No hardware trigger
1	GPIO_INT0
2	GPIO_INT1
3	FTM0_INIT_TRIG ORed with FTM0_EXT_TRIG
4	FTM1_INIT_TRIG ORed with FTM1_EXT_TRIG
5	ORed all FTM1_CHn_OUT
6	ACMP0_OUT
7	GPIOINT_BMATCH
8	ARM_TXEV

7.5 Analog comparator inputs

Table 65. Analog comparator inputs

Selection	Input source
0	VLADDER
1	ACMP_I1
2	ACMP_I2

Table 65. Analog comparator inputs

Selection	Input source
3	ACMP_I3
4	ACMP_I4
5	ACMP_I5
6	V_BANDGAP

7.6 DMA assignments

7.6.1 Input connections for SDMA request inputs

Table 66. SDMA request inputs

Input	Function
0	UART0_RX_DMA
1	UART0_TX_DMA
2	UART1_RX_DMA
3	UART1_TX_DMA
4	UART2_RX_DMA
5	UART2_TX_DMA
6	SPI0_RX_DMA
7	SPI0_TX_DMA
8	SPI1_RX_DMA
9	SPI1_TX_DMA
10	I2C0_RX_DMA
11	I2C0_TX_DMA
12	I3C0_RX_DMA
13	I3C0_TX_DMA
14	Reserved
15	Reserved

7.6.2 Peripheral input multiplexing connections for SDMA trigger inputs

Table 67. SDMA triggers

Selection	Input source
0	GPIO_INT4
1	GPIO_INT5
2	GPIO_INT6
3	GPIO_INT7
4	ADC0_SEQA_IRQ
5	ADC0_SEQB_IRQ
6	COMP0_OUT
7	FTM0_INIT_TRIG ORed with FTM0_EXT_TRIG

Table 67. SDMA triggers

Selection	Input source
8	FTM1_INIT_TRIG ORed with FTM1_EXT_TRIG
9	Ored (FTM0_CH0, FTM0_CH1,.., FTM0_CH5)
10	Ored (FTM1_CH0, FTM1_CH1,.., FTM1_CH3)
11	SDMA_TRIGOUT_A
12	SDMA_TRIGOUT_B

7.6.3 Peripheral input multiplexing connections for SDMA Trigger_OUTs feeding back into the SDMA Trigger Input Multiplexer

Table 68. Connection of interrupt sources to the NVIC

Selection	Input source
0	SDMA_CH0_TRIGOUT
1	SDMA_CH1_TRIGOUT
2	SDMA_CH2_TRIGOUT
3	SDMA_CH3_TRIGOUT
4	SDMA_CH4_TRIGOUT
5	SDMA_CH5_TRIGOUT
6	SDMA_CH6_TRIGOUT
7	SDMA_CH7_TRIGOUT
8	SDMA_CH8_TRIGOUT
9	SDMA_CH9_TRIGOUT
10	SDMA_CH10_TRIGOUT
11	SDMA_CH11_TRIGOUT
12	SDMA_CH12_TRIGOUT
13	SDMA_CH13_TRIGOUT
14	SDMA_CH14_TRIGOUT
15	SDMA_CH15_TRIGOUT

UM11607

Chapter 8: System configuration (SYSCON)

Rev. 3 — April 2023 **User manual**

8.1 How to read this chapter

The system configuration block is identical for all LPC86x parts.

8.2 Features

- Clock control
  - Configure system oscillator, low power oscillator, and FRO oscillator.
  - Enable clocks to individual peripherals and memories.
  - Configure clock output.
  - Configure clock dividers, digital filter clock, and USART baud rate clock.
  - Configure FlexTimers and ADC clock.
- Monitor and release reset to individual peripherals.
- Select pins for external pin interrupts and pattern match engine.
- Configuration of reduced power modes.
- Wake-up control.
- BOD configuration.
- Interrupt latency control.
- Select a source for the NMI.
- Calibrate system tick timer.

8.3 Basic configuration

Configure the SYSCON block as follows:

- The SYSCON uses the CLKIN, CLKOUT, $\overline{\text{RESET}}$, and XTALIN/OUT pins. Configure the pin functions through the switch matrix. See Section 8.4.
- No clock configuration is needed. The clock to the SYSCON block is always enabled. By default, the SYSCON block is clocked by the FRO.

8.3.1 Set up the FRO

The FRO provides a selectable fro_oscout of 36 MHz, 48 MHz, and 60 MHz outputs that can be used as a system clock. Also, the fro_oscout can be divided down to provide frequencies of 30 MHz, 24 MHz, or 18 MHz for system clock.

By default, the fro_oscout is 48 MHz and is divided by 2 to provide a default system (CPU) clock frequency of 24 MHz.

1. By default, the FRO is enabled. If required, the FRO can be enabled in the PDRUNCFG register:

   Section 8.6.47 “Power configuration register”

2. Select the fro_oscout (60 MHz/48 MHz/36 MHz) using the set_fro_frequency API call: Chapter 10 "FRO API ROM routine" and Figure 7 "Clock generation".
3. The FROOSCCTRL register can be used to select direct fro_oscout or divided fro_oscout for fro clock.
4. Use the FRODIRECTCLKUEN register to update the fro clock: Section 8.6.7 "FRO direct clock source update register".

8.3.2 Set up the PLL

The PLL creates a stable output clock at a higher frequency than the input clock. If you need a main clock with a frequency higher than the input clock, use the PLL to boost the input frequency.

1. Select the PLL input in the SYSPLLCLKSEL register. You have the following input options:
   – FRO: 24 MHz internal oscillator (default).
   – External clock input: It can be external crystal oscillator using the XTALIN/XTALOUT pins or CLKIN from external pin.

   Remark: The min frequency for PLL is 10 MHz.

   Section 8.6.9 "System PLL clock source select register"
2. Update the PLL clock source in the SYSPLLCLKUEN register.

   Section 8.6.10 "System PLL clock source update register"
3. Power-down the system PLL in the PDRUNCFG register.
4. Configure the PLL M and N dividers.

   Section 8.6.2 "System PLL control register"
5. Power up the system PLL in the PDRUNCFG register.

   Section 8.6.47 "Power configuration register"
6. Wait for the PLL to lock by monitoring the PLL lock status.

   Section 8.6.3 "System PLL status register"

8.3.3 Configure the main clock and system clock

The clock source for the registers and memories is derived from main clock. The main clock can be sourced from the main clock pre PLL or from the PLL.

The divided main clock is called the system clock and clocks the core, the memories, and the peripherals (register interfaces and peripheral clocks).

1. Select the main clock pre PLL. You have the following options:
   – FRO: 48 MHz internal oscillator (default).
   – External clock input: It can be external crystal oscillator using the XTALIN/XTALOUT pins or CLKIN from external pin.
   – Low power oscillator.
   – FRO DIV: 24 MHz (default).

   Section 8.6.12 "Main clock PLL source update enable register"

2. Update the main clock source.

   Section 8.6.14 “Main clock source update enable register”
3. Select the main clock. You have the following options:
   – Main clock pre PLL.
   – PLL output: You must configure the PLL to use the PLL output.

     Section 8.6.9 “System PLL clock source select register”
4. Update the main clock PLL source.

   Section 8.6.14 “Main clock source update enable register”
5. Select the divider value for the system clock. A divider value of 0 disables the system clock.

   Section 8.6.15 “System clock divider register”
6. Select the memories and peripherals that are operating in your application and therefore must have an active clock. The core is always clocked.

   Section 8.6.23 “System clock control 0 register”

8.3.4 Set up the system oscillator using XTALIN and XTALOUT

To use the system oscillator with the LPC86x, assign the XTALIN and XTALOUT pins, which connect to the external crystal, through the fixed-pin function in the switch matrix. XTALIN and XTALOUT can only be assigned to pins PIO0_8 and PIO0_9.

1. In the IOCON block, remove the pull-up and pull-down resistors in the IOCON registers for pins PIO0_8 and PIO0_9.
2. In the switch matrix block, enable the 1-bit functions for XTALIN and XTALOUT.
3. In the SYSOSCCTRL register, disable the BYPASS bit and select the oscillator frequency range according to the desired oscillator output clock.
4. Set SYSOSC_PD bit to 0 in PDRUNCFG register to turn on the system oscillator.
5. Wait 500 μs for the system oscillator to stabilize.

Related registers:

Table 156 “PIO0_8 register (PIO0_8, address 0x4004 4038) bit description”

Table 155 “PIO0_9 register (PIO0_9, address 0x4004 4034) bit description”

Table 139 “Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description”

Table 75 “System oscillator control register (SYSOSCCTRL, address 0x4004 8020) bit description”

8.4 Pin description

The SYSCON inputs and outputs are assigned to external pins through the switch matrix.

See Section 11.3.1 “Connect an internal signal to a package pin” to assign the CLKOUT function to a pin.

See Section 11.3.2 to enable the clock input, the oscillator pins, and the external reset input.

Table 69. SYSCON pin description

Function	Direction	Pin	Description	SWM register	Reference
CLKOUT	O	any	CLKOUT clock output.	PINASSIGN8	Table 136
CLKIN	I	PIO0_1/ACMP_I1/CLKIN	External clock input to the system PLL. Disable the ACMP_I1 function in the PINENABLE register.	PINENABLE0	Table 139
XTALIN	I	PIO0_8/XTALIN	Input to the system oscillator.	PINENABLE0	Table 139
XTALOUT	O	PIO0_9/XTALOUT	Output from the system oscillator.	PINENABLE0	Table 139
$\overline{\text{RESET}}$	I	$\overline{\text{RESET}}$/PIO0_5	External reset input	PINENABLE0	Table 139

8.5 General description

8.5.1 Clock generation

The system control block generates all clocks for the chip. Only the low-power oscillator used for wake-up timing is controlled by the PMU. Except for the USART clock, SPI clock, I^{2}C clock, I3C clock, FlexTimer clock, ADC clock, and the clock to configure the glitch filters of the digital I/O pins, the clocks to the core and peripherals run at the same frequency. The maximum system clock frequency is 60 MHz. See Figure 7.

Remark: The main clock frequency is limited to 60 MHz.

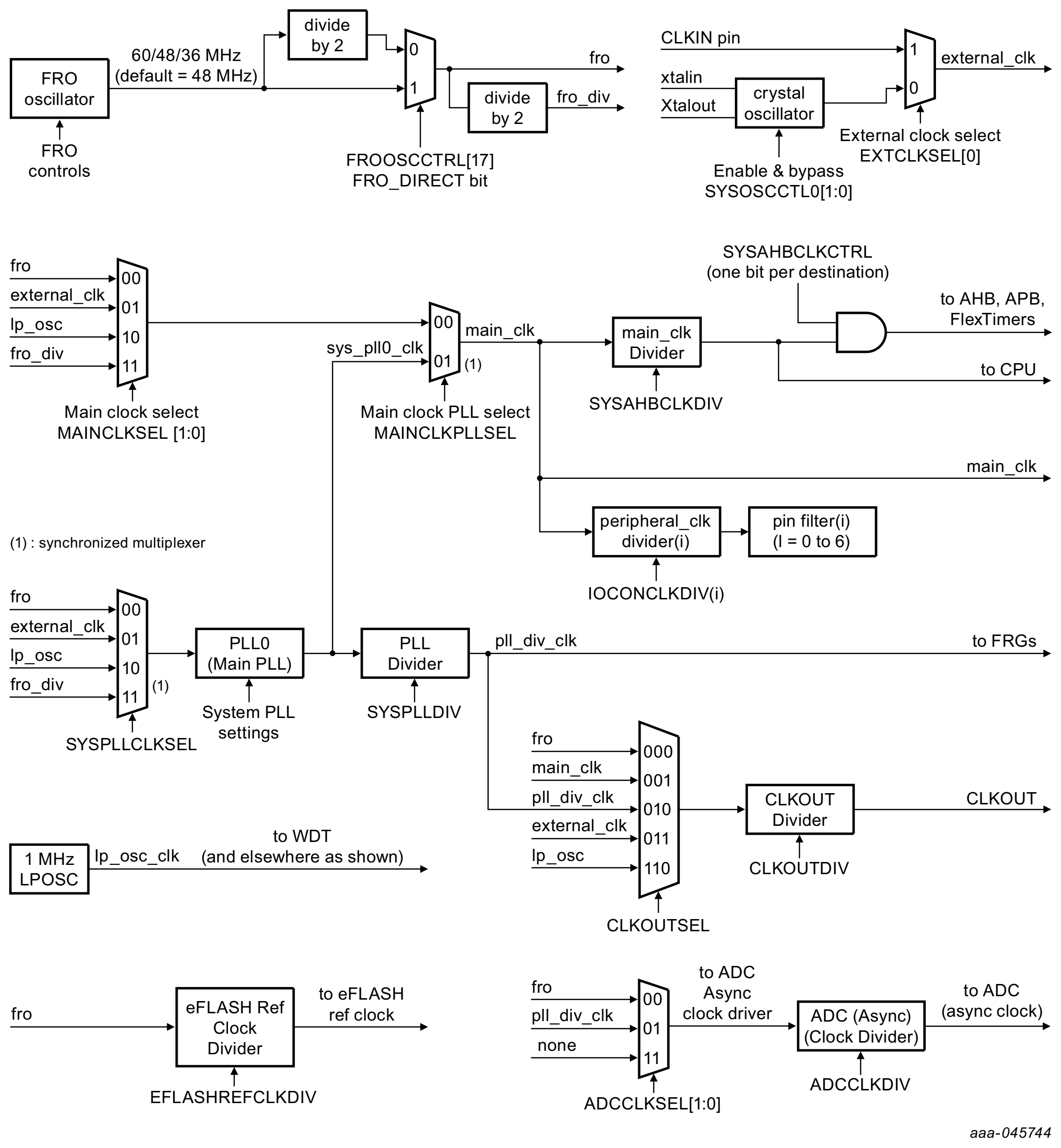


Fig 7. Clock generation

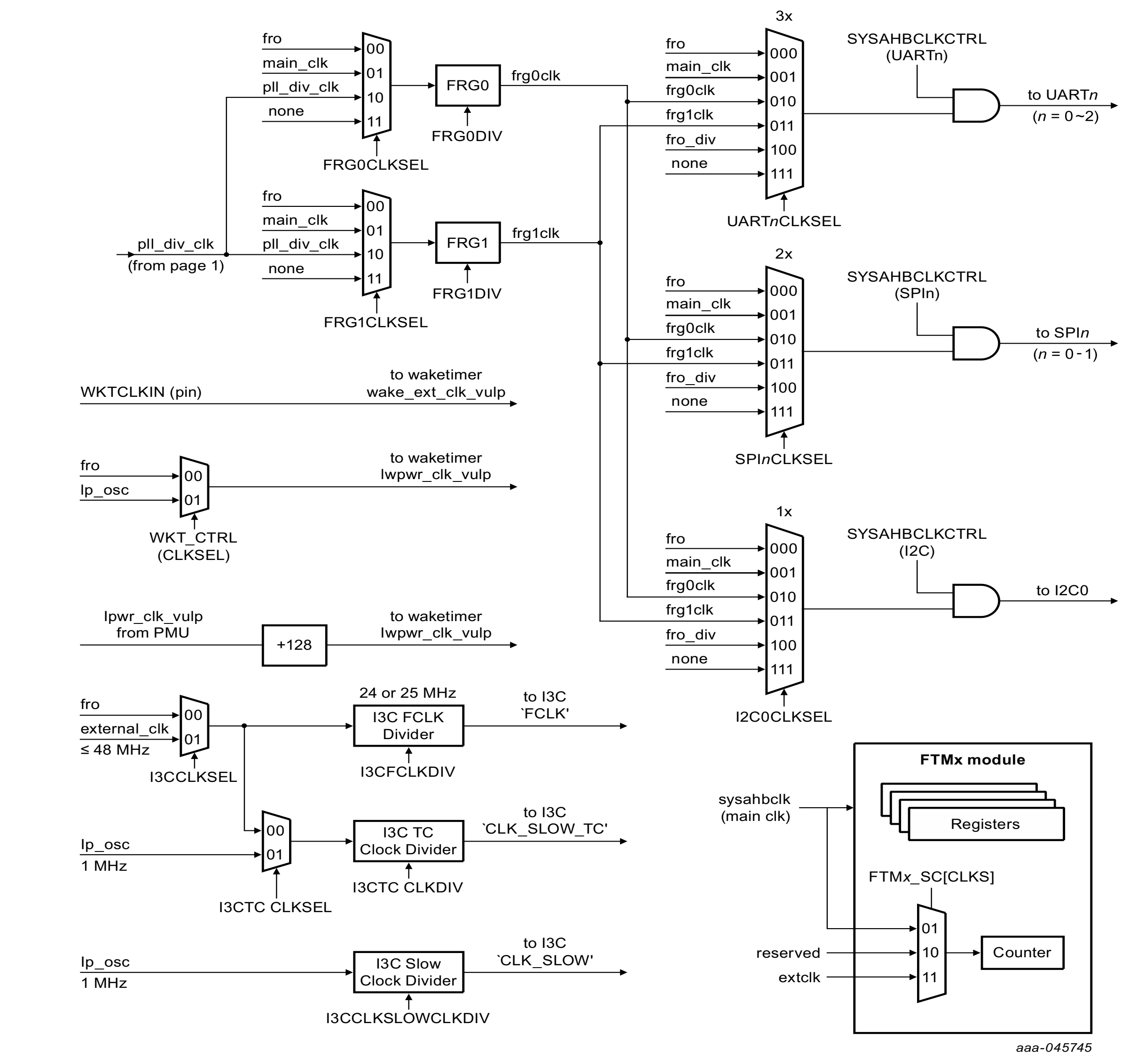


Fig 8. Clock generation (continued)

Table 70. Clocking diagram signal name descriptions

Name	Description
sys_osc_clk	This is the internal clock that comes from external crystal oscillator through dedicated pins.
frg_clk	The output of the Fractional Rate Generator. The FRG and its source selection are shown in Figure 8 “Clock generation (continued)”.
fro	The output of the currently selected on-chip FRO oscillator. See Figure 7 “Clock generation”.
fro_div	The FRO output. This may be either 30 MH, 24 MHz, or 18 MHz. See Figure 7 “Clock generation”.
main_clk	The main clock used by the CPU and AHB bus, and potentially many others. The main clock and its source selection are shown in Figure 7 “Clock generation”.
“none”	A tied-off source that should be selected to save power when the output of the related multiplexer is not used.

Table 70. Clocking diagram signal name descriptions

Name	Description
sys_pll0_clk	The output of the System PLL. The System PLL and its source selection are shown in Figure 7 “Clock generation”.
lp_osc_clk	The output of the low power oscillator, which has a selectable target frequency. It must also be enabled in the PDRINCFG0 register.
xtalin	Input of the main oscillator. If used, this is connected to an external crystal and load capacitor.
xtalout	Output of the main oscillator. If used, this is connected to an external crystal and load capacitor.
clk_in	This is the internal clock that comes from the main CLK_IN pin function. Connect that function to the pin by selecting it in the IOCON block.
external_clk	This is the internal clock that comes from the external crystal oscillator or the CLK_IN pin.
extclk	FlexTimer external clock coming from FTM0_EXTCLK or FTM1_EXTCLK.

8.5.2 Power control of analog components

The system control block controls the power to the analog components such as the oscillators and PLL, the BOD, and the analog comparator. For details, see the following registers:

Section 8.6.45 “Deep-sleep mode configuration register”

Section 8.6.2 “System PLL control register”

Section 8.6.4 “System oscillator control register”

8.5.3 Configuration of reduced power-modes

The system control block configures analog blocks that can remain running in the reduced power modes (the BOD and the low power oscillator for safe operation) and enables various interrupts to wake up the chip when the internal clocks are shut down in Deep-sleep and Power-down modes. For details, see the following registers:

Section 8.6.47 “Power configuration register”

Section 8.6.44 “Start logic 1 interrupt wake-up enable register”

8.5.4 Reset and interrupt control

The peripheral reset control register in the system control register allows to assert and release individual peripheral resets.

Up to eight external pin interrupts can be assigned to any digital pin in the system control block (see Section 8.6.42 “Pin interrupt select registers”).

8.6 Register description

All system control block registers reside on word address boundaries. Details of the registers appear in the description of each function.

Reset values describe the content of the registers after the bootloader has executed.

All address offsets shown in Table 71 as reserved should not be written to.

Table 71. Register overview: System configuration (base address 0x4004 8000)

Name	Access	Offset	Description	Reset value	Section
SYSMEMREMAP	R/W	0x000	System memory remap	0x0	8.6.1
-	-	0x004	Reserved	-	-
SYSPLLCTRL	R/W	0x008	System PLL control	0	8.6.2
SYSPLLSTAT	R	0x00C	System PLL status	0	8.6.3
-	-	0x010	Reserved	-	-
-	-	0x014	Reserved	-	-
-	-	0x018	Reserved	-	-
-	-	0x01C	Reserved	-	-
SYSOSCCTRL	R/W	0x020	System oscillator control	0x000	8.6.4
LPOSCCTRL	R/W	0x024	Low power oscillator control	0x000	8.6.5
FROOSCCTRL	R/W	0x028	FRO oscillator control	0x8801	8.6.7
-	-	0x02C	Reserved	-	-
FRODIRECTCLKUEN	R/W	0x030	FRO direct clock source update enable	0	8.6.7
-	-	0x034	Reserved	-	-
SYSRSTSTAT	R/W	0x038	System reset status register	0x0B	8.6.8
		0x03C			
SYSPLLCLKSEL	R/W	0x040	System PLL clock source select	0	8.6.9
SYSPLLCLKUEN	R/W	0x044	System PLL clock source update enable	0	8.6.10
MAINCLKPLLSEL	R/W	0x048	Main clock PLL source select	0	8.6.11
MAINCLKPLLUEN	R/W	0x04C	Main clock PLL source update enable	0	8.6.12
MAINCLKSEL	R/W	0x050	Main clock source select	0	8.6.13
MAINCLKUEN	R/W	0x054	Main clock source update enable	0	8.6.14
SYSAHBCLKDIV	R/W	0x058	System clock divider	1	8.6.15
SYSPLLDIV	R/W	0x05C	System PLL clock divider	0	8.6.16
-	-	0x060	Reserved	-	-
ADCCLKSEL	R/W	0x064	ADC clock source select	0	8.6.17
ADCCLKDIV	R/W	0x068	ADC clock divider	0	8.6.18
WKTCLKSEL	R/W	0x06C	WKT clock source select	0	8.6.19
-	-	0x070	Reserved	-	-
EXTCLKSEL	R/W	0x074	External clock source select	0	8.6.20
I3CCLKDIV	R/W	0x078	I3C clock divider	0	8.6.21
LPOSCEN	R/W	0x07C	LPOSC enable	1	8.6.22
SYSAHBCLKCTRL0	R/W	0x080	System clock control 0	0x17	8.6.23
-	-	0x084	Reserved	-	-

Table 71. Register overview: System configuration (base address 0x4004 8000) *…continued*

Name	Access	Offset	Description	Reset value	Section
PRESETCTRL0	R/W	0x088	Peripheral reset control 0	0xFFFFFFFF	8.6.24
PRESETCTRL1	R/W	0x08C	Peripheral reset control 1	0x1F	8.6.25
UART0CLKSEL	R/W	0x090	Function clock source select for UART0	0x7	8.6.26
UART1CLKSEL	R/W	0x094	Function clock source select for UART1	0x7	8.6.26
UART2CLKSEL	R/W	0x098	Function clock source select for UART2	0x7	8.6.26
-	R/W	0x09C	Reserved	-	-
-	R/W	0x0A0	Reserved	-	-
I2C0CLKSEL	R/W	0x0A4	Function clock source select for I^{2}C0	0x7	8.6.26
I3CCLKSEL	R/W	0x0A8	Function clock source select for I3C0	0x7	8.6.26
I3CSLOWTCCLKSEL	R/W	0x0AC		0x7	8.6.26
I3CSLOWCLKSEL	R/W	0x0B0		0x7	8.6.26
SPI0CLKSEL	R/W	0x0B4	Function clock source select for SPI0	0x7	8.6.26
SPI1CLKSEL	R/W	0x0B8	Function clock source select for SPI1	0x7	8.6.26
FTM0INTTRIGDIV	R/W	0x0BC	FTM trigger input divider	0x0	8.6.27
EFLASHREFCLKDIV	R/W	0x0C0	Reserved	-	-
-	-	0x0C4	Reserved	-	-
FTMFLTCFG	-	0x0C8	FTM fault configuration	0x0	8.6.28
-	-	0x0CC	Reserved	-	-
FRG0DIV	R/W	0x0D0	Fractional generator divider value	0x0	8.6.29
FRG0MULT	R/W	0x0D4	Fractional generator multiplier value	0x0	8.6.30
FRG0CLKSEL	R/W	0x0D8	FRG0 clock source select	0	8.6.31
-	-	0x0DC	Reserved	-	-
FRG1DIV	R/W	0x0E0	Fractional generator divider value	0x0	8.6.32
FRG1MULT	R/W	0x0E4	Fractional generator multiplier value	0x0	8.6.33
FRG1CLKSEL	R/W	0x0E8	FRG1 clock source select	0	8.6.34
-	-	0x0EC	Reserved	-	
CLKOUTSEL	R/W	0x0F0	CLKOUT clock source select	0	8.6.35
CLKOUTDIV	R/W	0x0F4	CLKOUT clock divider	0	8.6.36
-	-	0x0F8	Reserved	-	-
-	-	0x100 - 0x130	Reserved	-	-
IOCONCLKDIV6	R/W	0x134	Peripheral clock 6 to the IOCON block for programmable glitch filter	0	8.6.37
IOCONCLKDIV5	R/W	0x138	Peripheral clock 5 to the IOCON block for programmable glitch filter	0	8.6.37
IOCONCLKDIV4	R/W	0x13C	Peripheral clock 4 to the IOCON block for programmable glitch filter	0	8.6.37
IOCONCLKDIV3	R/W	0x140	Peripheral clock 3 to the IOCON block for programmable glitch filter	0	8.6.37
IOCONCLKDIV2	R/W	0x144	Peripheral clock 2 to the IOCON block for programmable glitch filter	0	8.6.37
IOCONCLKDIV1	R/W	0x148	Peripheral clock 1 to the IOCON block for programmable glitch filter	0	8.6.37

Table 71. Register overview: System configuration (base address 0x4004 8000) *…continued*

Name	Access	Offset	Description	Reset value	Section
IOCONCLKDIV0	R/W	0x14C	Peripheral clock 0 to the IOCON block for programmable glitch filter	0	8.6.37
BODCTRL	R/W	0x150	Brown-Out Detect	0x0000 0010	8.6.38
SYSTCKCAL	R/W	0x154	System tick counter calibration	0	8.6.39
-	R/W	0x158 - 0x16C	Reserved	-	-
IRQLATENCY	R/W	0x170	IRQ delay. Allows trade-off between interrupt latency and determinism.	0x0000 0010	8.6.40
NMISRC	R/W	0x174	NMI Source Control	0	8.6.41
PINTSEL0	R/W	0x178	GPIO Pin Interrupt Select register 0	0	8.6.42
PINTSEL1	R/W	0x17C	GPIO Pin Interrupt Select register 1	0	8.6.42
PINTSEL2	R/W	0x180	GPIO Pin Interrupt Select register 2	0	8.6.42
PINTSEL3	R/W	0x184	GPIO Pin Interrupt Select register 3	0	8.6.42
PINTSEL4	R/W	0x188	GPIO Pin Interrupt Select register 4	0	8.6.42
PINTSEL5	R/W	0x18C	GPIO Pin Interrupt Select register 5	0	8.6.42
PINTSEL6	R/W	0x190	GPIO Pin Interrupt Select register 6	0	8.6.42
PINTSEL7	R/W	0x194	GPIO Pin Interrupt Select register 7	0	8.6.42
-	-	0x198 - 0x200	Reserved	-	-
STARTERP0	R/W	0x204	Start logic 0 pin wake-up enable register	0	8.6.43
-	-	0x208 - 0x210	Reserved	-	-
STARTERP1	R/W	0x214	Start logic 1 interrupt wake-up enable register	0	8.6.44
-	-	0x218 - 0x22C	Reserved	-	-
PDSLEEPCFG	R/W	0x230	Power-down states in deep-sleep mode	0xFFFF	8.6.45
PDAWAKECFG	R/W	0x234	Power-down states for wake-up from deep-sleep	0xEDF0	8.6.46
PDRUNCFG	R/W	0x238	Power configuration register	0xEDF0	8.6.47
		0x23C	Reserved		
FLASHCACHECFG	R/W	0x240	Flash cache configuration register	0	8.6.48
		0x300 - 0x3F4	Reserved		
DEVICE_ID	R	0x3F8	Device ID	part dependent	8.6.49

8.6.1 System memory remap register

The system memory remap register selects whether the exception vectors are read from boot ROM, flash, or SRAM. By default, the flash memory is mapped to address 0x0000 0000. When the MAP bits in the SYSMEMREMAP register are set to 0x0 or 0x1, the boot ROM or RAM respectively are mapped to the bottom 512 bytes of the memory map (addresses 0x0000 0000 to 0x0000 0200).

Table 72. System memory remap register (SYSMEMREMAP, address 0x4004 8000) bit description

Bit	Symbol	Value	Description	Reset value
1:0	MAP		System memory remap. Value 0x3 is reserved.	0x0
		0x0	Bootloader Mode. Interrupt vectors are re-mapped to Boot ROM.	
		0x1	User RAM Mode. Interrupt vectors are re-mapped to Static RAM.	
		0x2	User Flash Mode. Interrupt vectors are not re-mapped and reside in Flash.	
31:2	-	-	Reserved	-

8.6.2 System PLL control register

This register connects and enables the system PLL and configures the PLL multiplier and divider values. The PLL accepts an input frequency from 10 MHz to 25 MHz from various clock sources. The input frequency is multiplied to a higher frequency and then divided down to provide the actual clock used by the CPU, peripherals, and memories. The PLL can produce a clock up to the maximum allowed for the CPU.

Remark: The divider values for P and M must be selected so that the PLL output clock frequency FCLKOUT is lower than 100 MHz.

Table 73. System PLL control register (SYSPLLCTRL, address 0x4004 8008) bit description

Bit	Symbol	Value	Description	Reset value
4:0	MSEL		Feedback divider value. The division value M is the programmed MSEL value + 1. 00000: Division ratio M = 1 to 11111: Division ratio M = 32	0
6:5	PSEL		Post divider ratio P. The division ratio is 2 × P.	0
		0x0	P = 1	
		0x1	P = 2	
		0x2	P = 4	
		0x3	P = 8	
31:7	-	-	Reserved. Do not write ones to reserved bits.	-

8.6.3 System PLL status register

This register is a Read-only register and supplies the PLL lock status (see Section 8.7.3.1).

Table 74. System PLL status register (SYSPLLSTAT, address 0x4004 800C) bit description

Bit	Symbol	Value	Description	Reset value
0	LOCK		PLL lock status	0
		0	PLL not locked	
		1	PLL locked	
31:1	-	-	Reserved	-

8.6.4 System oscillator control register

This register configures the frequency range for the system oscillator. The system oscillator itself is powered on or off in the PDRUNCFG register. See Table 118.

Table 75. System oscillator control register (SYSOSCCTRL, address 0x4004 8020) bit description

Bit	Symbol	Value	Description	Reset value
0	BYPASS		Bypass system oscillator	0x0
		0	Disabled. Oscillator is not bypassed.	
		1	Enabled. PLL input (sys_osc_clk) is fed directly from the XTALIN pin bypassing the oscillator. Use this mode when using an external clock source instead of the crystal oscillator.	
1	FREQRANGE		Determines oscillator frequency range.	0x0
		0	1 - 20 MHz frequency range.	
		1	15 - 25 MHz frequency range	
31:2	-	-	Reserved	0x00

8.6.5 Low power oscillator control register

This register configures the low power oscillator. The oscillator consists of an analog and a digital part. The analog part contains the oscillator function and generates an analog clock (Fclkana). With the digital part, the analog output clock can be divided to the required output clock frequency wdt_osc_clk. The analog output frequency (Fclkana) can be adjusted with the FREQSEL bits between 600 kHz and 4.6 MHz. With the digital part Fclkana will be divided (divider ratios = 2, 4,...,64) to wdt_osc_clk using the DIVSEL bits. The output clock frequency of the low power oscillator can be calculated as wdt_osc_clk = Fclkana/(2 x (1 + DIVSEL)) = 9.3 kHz to 2.3 MHz (nominal values).

Remark: Any setting of the FREQSEL bits will yield a Fclkana value within ?40% of the listed frequency value. The low power oscillator is the clock source with the lowest power consumption. If accurate timing is required, use the FRO or system oscillator.

Remark: The frequency of the low power oscillator is undefined after reset. The low power oscillator frequency must be programmed by writing to the LPOSCCTRL register before using the low power oscillator.

Table 76. Low-power oscillator control register (LPOSCCTRL, address 0x4004 8024) bit description

Bit	Symbol	Value	Description	Reset value
7:0	LPOSC_TRIM		Select divider for Fclkana. lp_osc_clk = Fclkana/ (2 x (1 + DIVSEL)) 00000: 2 x (1 + DIVSEL) = 2 00001: 2 x (1 + DIVSEL) = 4 to 11111: 2 x (1 + DIVSEL) = 64	0
11:8	LPOSC_TRIM_T		Select low power oscillator analog output frequency (Fclkana).	0
14:12	LPOSC_TRIM_C	1		0
30:15	-	-	Reserved	0
31	LPOSC_TST_IREF_EN			0

8.6.6 FRO oscillator control register

The FROOSCCTRL register can be used to select direct fro_oscout (60 MHz, 48 MHz, 36 MHz) or select divided fro_oscout (30 MHz, 24 MHz, or 18 MHz).

The set_fro_frequency API call (Chapter 10 “FRO API ROM routine”) must be used to select desired output frequency from FRO. See Figure 9 “LPC86x clock generation (continued)”.

Table 77. FRO oscillator control register (FROOSCCTRL, address 0x4004 8028) bit description

Bit	Symbol	Value	Description	Reset value
16:0	-		Reserved	0
17	FRO_DIRECT		FRO direct clock select	0
		0	fro_oscout is divided by 2 (normal boot).	
		1	FRO clock is direct from FRO oscillator	
31:18	-	-	Reserved	-

8.6.7 FRO direct clock source update register

The FRODIRECTCLKUEN register updates the clock source of the FRO clock with the new input clock after the FROOSCCTRL register bit 17 has been written to. In order for the update to take effect, first write a zero to the FRODIRECTCLKUEN register and then write a one to FRODIRECTCLKUEN.

Table 78. FRO direct clock source update enable register (FRODIRECTCLKUEN, address 0x4004 8030) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable FRO clock source update.	0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	-

8.6.8 System reset status register

The SYSRSTSTAT register shows the source of the latest reset event. The bits are cleared by writing a one to any of the bits. The POR event clears all other bits in this register. If another reset signal - for example the external $\overline{\text{RESET}}$ pin - remains asserted after the POR signal is negated, then its bit is set to detected. Write a one to clear the reset.

The reset value given in Table 79 applies to the POR reset.

Table 79. System reset status register (SYSRSTSTAT, address 0x4004 8038) bit description

Bit	Symbol	Value	Description	Reset value
0	POR		POR reset status	0x0B
		0	No POR detected	
		1	POR detected. Writing a one clears this reset.	
1	EXTRST		Status of the external $\overline{\text{RESET}}$ pin. External reset status.	Depends on $\overline{\text{RESET}}$ pin
		0	No reset event detected.	
		1	Reset detected. Writing a one clears this reset.	
2	WDT		Status of the Watchdog reset.	0
		0	No WDT reset detected.	
		1	WDT reset detected. Writing a one clears this reset.	
3	BOD		Status of the Brown-out detect reset	0
		0	No BOD reset detected	
		1	BOD reset detected. Writing a one clears this reset.	
4	SYSRST		Status of the software system reset	0
		0	No System reset detected	
		1	System reset detected. Writing a one clears this reset.	
31:5	-	-	Reserved	-

8.6.9 System PLL clock source select register

This register selects the clock source for the system PLL. The SYSPLLCLKUEN register (see Section 8.6.10) must be toggled from LOW to HIGH for the update to take effect.

Table 80. System PLL clock source select register (SYSPLLCLKSEL, address 0x4004 8040) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		System PLL clock source	0
		0x0	FRO	
		0x1	External clock	
		0x2	Low power oscillator	
		0x3	FRO DIV	
31:2	-	-	Reserved	-

8.6.10 System PLL clock source update register

This register updates the clock source of the system PLL with the new input clock after the SYSPLLCLKSEL register has been written to. In order for the update to take effect, first write a zero to the SYSPLLUEN register and then write a one to SYSPLLUEN.

Table 81. System PLL clock source update enable register (SYSPLLCLKUEN, address 0x4004 8044) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable system PLL clock source update	0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	-

8.6.11 Main clock PLL source select register

The MAINCLKPLLSEL register selects the main system clock, which can be the system PLL (sys_pllclkout), or the main clock pre pll. The main system clock clocks the core, the peripherals, and the memories.

Bit 0 of the MAINCLKUEN register (see Section 8.6.14) must be toggled from 0 to 1 for the update to take effect.

Table 82. Main clock source select register (MAINCLKPLLSEL, address 0x4004 8048) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		Clock source for main clock	0
		0x0	main_clk_pre_pll	
		0x1	SYS PLL	
		0x2	None	
		0x3	None	
31:2	-	-	Reserved	-

8.6.12 Main clock PLL source update enable register

.The MAINCLKPLLUEN register updates the clock source of the main clock with the new input clock after the MAINCLKPLLSEL register has been written to. In order for the update to take effect, first write a zero to bit 0 of this register, then write a one.

Table 83. Main clock source update enable register (MAINCLKPLLUEN, address 0x4004 804C) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable main clock source update.	0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	-

8.6.13 Main clock source select register

The **MAINCLKSEL** register selects the main_clock_pre_pll, which can be the FRO, external clock, low power oscillator, or FRO_DIV.

Bit 0 of the MAINCLKUEN register (Section 8.6.14) must be toggled from 0 to 1 for the update to take effect.

Table 84. Main clock source select register (MAINCLKSEL, address 0x4004 8050) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		Clock source for main clock pre pll	0
		0x0	FRO	
		0x1	External clock	
		0x2	Low power oscillator	
		0x3	FRO_DIV = FRO / 2	
31:2	-	-	Reserved	-

8.6.14 Main clock source update enable register

The **MAINCLKUEN** register updates the clock source of the main clock with the new input clock after the MAINCLKSEL register has been written to. In order for the update to take effect, first write a zero to bit 0 of this register, then write a one.

Table 85. Main clock source update enable register (MAINCLKUEN, address 0x4004 8054) bit description

Bit	Symbol	Value	Description	Reset value
0	ENA		Enable main clock source update	0
		0	No change	
		1	Update clock source	
31:1	-	-	Reserved	-

8.6.15 System clock divider register

This register controls how the main clock is divided to provide the system clock to the core, memories, and the perispherals. The system clock can be shut down completely by setting the DIV field to zero.

Table 86. System clock divider register (SYSAHBCLKDIV, address 0x4004 8058) bit description

Bit	Symbol	Description	Reset value
7:0	DIV	System AHB clock divider values 0: System clock disabled. 1: Divide by 1. … 255: Divide by 255.	0x01
31:8	-	Reserved	-

8.6.16 System PLL clock divider register

Table 87. System PLL clock divider register (SYSPLLDIV, address 0x4004 805C) bit description

Bit	Symbol	Description	Reset value
7:0	DIV	System PLL clock divider values 0: System clock disabled. 1: Divide by 1. … 255: Divide by 255.	0x01
31:8	-	Reserved	-

8.6.17 ADC clock source select register

The ADCCLKSEL register selects the ADC clock, which can be the FRO or sys_pll.

Table 88. ADC clock source select register (ADCCLKSEL, address 0x4004 8064) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		Clock source for ADC clock.	0x0
		0x0	FRO	
		0x1	SYS PLL	
		0x2	None	
		0x3	None	
31:3	-		Reserved	-

8.6.18 ADC clock divider register

The ADCCLKDIV register controls how the ADC clock is divided to provide the ADC clock to the ADC controller. The ADC clock can be shut down completely by setting the DIV field to zero.

Table 89. ADC clock divider register (ADCCLKDIV, address 0x4004 8068) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		ADC clock divider values. 0: ADC clock disabled. 1: Divide by 1. … 255: Divide by 255.	0x0
31:8	-		Reserved	-

8.6.19 WKT clock source select register

The WKTCLKSEL register selects the WKT clock.

Table 90. WKT clock source select register (WKTCLKSEL, address 0x4004 806C) bit description

Bit	Symbol	Value	Description	Reset value
0	WKTCLK SEL		WKT clock source selection 0: FRO_INT 1: LPOSC	0x0
31:1	-		Reserved	-

8.6.20 External clock source select register

The EXTCLKSEL register selects the external clock, which can be the system oscillator or clk_in (direct from external IO).

Table 91. External clock source select register (EXTCLKSEL, address 0x4004 8074) bit description

Bit	Symbol	Value	Description	Reset value
0	SEL		Clock source for external clock	0x0
		0x0	System oscillator	
		0x1	CLK_IN	
31:1	-		Reserved	-

8.6.21 I3C clock divider register

Table 92. I3C clock divider register (I3CCLKDIV, address 0x4004 8078) bit description

Bit	Symbol	Value	Description	Reset value
7:0	I3C_FC LK_DIV		i3c_fclk fast clock divider	0x0
15:8	I3C_SL OW_TC _CLK_D IV		i3c_slow_tc_clk clock divider	0x0
23:16	I3C_SL OW_CL K_DIV		i3c_slow_clk divider	0x0
31:24	-		Reserved	-

8.6.22 LPOSC enable register

Table 93. External clock source select register (LPOSCEN, address 0x4004 807C) bit description

Bit	Symbol	Value	Description	Reset value
0	WDT_C LK_EN		WDT count clock enable	0x1
		0	Disable	
		1	Enable	
1	WKT_C LK_EN		WKT count clock enable	0x0

Table 93. External clock source select register (LPOSCEN, address 0x4004 807C) bit description

Bit	Symbol	Value	Description	Reset value
		0	Disable	
		1	Enable	
31:2	-		Reserved	-

8.6.23 System clock control 0 register

The SYSAHBCLKCTRL0 register enables the clocks to individual system and peripheral blocks. The system clock (bit 0) provides the clock for the AHB, the APB bridge, the ARM Cortex-M0+, the SYSCON block, and the PMU. This clock cannot be disabled.

Table 94. System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description

Bit	Symbol	Value	Description	Reset value
0	SYS		Enables the clock for the AHB, the APB bridge, the Cortex-M0+ core clocks, SYSCON, and the PMU. This bit is read only and always reads as 1.	1
1	ROM		Enables clock for ROM.	1
		0	Disable	
		1	Enable	
2	RAM		Enables clock for SRAM	1
		0	Disable	
		1	Enable	
3			Reserved	0
4	FLASH		Enables clock for flash.	1
		0	Disable	
		1	Enable	
5	I2C0		Enables clock for I2C0.	0
		0	Disable	
		1	Enable	
6	GPIO0		Enables clock for GPIO0 port registers.	0
		0	Disable	
		1	Enable	
7	SWM		Enables clock for switch matrix.	0
		0	Disable	
		1	Enable	
8			Reserved	0
9	WKT		Enables clock for self-wake-up timer.	0
		0	Disable	
		1	Enable	
10	MRT		Enables clock for multi-rate timer.	0
		0	Disable	
		1	Enable	

Table 94. System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
11	SPI0		Enables clock for SPI0.	0
		0	Disable	
		1	Enable	
12	SPI1		Enables clock for SPI1.	
		0	Disable	
		1	Enable	
13	CRC		Enables clock for CRC.	0
		0	Disable	
		1	Enable	
14	UART0		Enables clock for USART0.	0
		0	Disable	
		1	Enable	
15	UART1		Enables clock for USART1.	0
		0	Disable	
		1	Enable	
16	UART2		Enables clock for USART2.	0
		0	Disable	
		1	Enable	
17	WWDT		Enables clock for WWDT.	0
		0	Disable	
		1	Enable	
18	IOCON		Enables clock for IOCON block.	0
		0	Disable	
		1	Enable	
19	ACMP		Enables clock to analog comparator.	0
		0	Disable	
		1	Enable	
20	GPIO1		Enables clock for GPIO1 port registers.	0
		0	Disable	
		1	Enable	
21	FTM0		Enables clock to FTM0.	0
		0	Disable	
		1	Enable	
22	FTM1		Enables clock to FTM1.	0
		0	Disable	
		1	Enable	
23	I3C		Enables clock to I3C.	0
		0	Disable	
		1	Enable	

Table 94. System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
24	ADC		Enables clock to ADC.	0
		0	Disable	
		1	Enable	
27:25			Reserved	0
28	GPIO_INT		Enable clock for GPIO pin interrupt registers.	0
		0	Disable	
		1	Enable	
29	DMA		Enables clock to DMA.	0
		0	Disable	
		1	Enable	
31:30			Reserved	0

8.6.24 Peripheral reset control 0 register

The PRESET0CTRL register allows software to reset specific peripherals. A zero in any assigned bit in this register resets the specified peripheral. A 1 clears the reset and allows the peripheral to operate.

Table 95. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
3:0	-		Reserved	1
4	FLASH_RST_N		Flash controller reset control	1
		0	Assert the flash controller reset.	
		1	Clear the flash controller reset.	
5	I2C0_RST_N		I^{2}C0 reset control	1
		0	Assert the I^{2}C0 reset.	
		1	Clear the I^{2}C0 reset.	
6	GPIO0_RST_N		GPIO0 reset control	1
		0	Assert the GPIO0 reset	
		1	Clear the GPIO0 reset.	
7	SWM_RST_N		SWM reset control	1
		0	Assert the SWM reset.	
		1	Clear the SWM reset.	
8			Reserved	1
9	WKT_RST_N		Self-wake-up timer (WKT) reset control	1
		0	Assert the WKT reset.	
		1	Clear the WKT reset.	
10	MRT_RST_N		Multi-rate timer (MRT) reset control	1
		0	Assert the MRT reset.	
		1	Clear the MRT reset.	

Table 95. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
11	SPI0_RST_N		11SPI0_RST_N	1
		0	Assert the SPI0 reset.	
		1	Clear the SPI0 reset.	
12	SPI1_RST_N		SPI1 reset control	1
		0	Assert the SPI1 reset.	
		1	Clear the SPI1 reset.	
13	CRC_RST_N		CRC engine reset control	1
		0	Assert the CRC reset.	
		1	Clear the CRC reset.	
14	UART0_RST_N		UART0 reset control	1
		0	Assert the UART0 reset.	
		1	Clear the flash UART0 reset.	
15	UART1_RST_N		UART1 reset control	1
		0	Assert the UART1 reset.	
		1	Clear the UART1 reset.	
16	UART2_RST_N		UART2 reset control	1
		0	Assert the UART2 reset.	
		1	Clear the UART2 reset.	
17	-	-	Reserved	
18	IOCON_RST_N		IOCON reset control	1
		0	Assert the IOCON reset.	
		1	Clear the IOCON reset.	
19	ACMP_RST_N		Analog comparator reset control	1
		0	Assert the analog comparator reset.	
		1	Clear the analog comparator reset.	
20	GPIO1_RST_N		GPIO1 reset control	1
		0	Assert the GPIO1 reset.	
		1	Clear the GPIO1 reset.	
21	FTM0_RST_N		FTM0 reset control	1
		0	Assert the FTM0 reset.	
		1	Clear the FTM0 reset.	
22	FTM1_RST_N		FTM1 reset control	1
		0	Assert the FTM1 reset.	
		1	Clear the FTM1 reset.	
23	I3C_RST_N		I3C reset control	1
		0	Assert the I3C reset.	
		1	Clear the I3C reset.	
24	ADC_RST_N		ADC reset control	1
		0	Assert the ADC reset.	
		1	Clear the ADC reset.	

Table 95. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
27:25	-	-	Reserved	1
28	GPIOINT_RST_N		GPIOINT reset control	1
		0	Assert the GPIOINT reset.	
		1	Clear the GPIOINT reset.	
29	DMA_RST_N		DMA reset control	1
		0	Assert the DMA reset.	
		1	Clear the DMA reset.	
31:30	-	-	Reserved	1

8.6.25 Peripheral reset control 1 register

The PRESETCTRL1 register allows software to reset specific peripherals. A zero in any assigned bit in this register resets the specified peripheral. A 1 clears the reset and allows the peripheral to operate.

Table 96. Peripheral reset control 1 register (PRESETCTRL1, address 0x4004 808C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved	1
3	FRG0_RST_N		Fractional baud rate generator 0 reset control	1
		0	Assert the FRG0 reset.	
		1	Clear the FRG0 reset.	
4	FRG1_RST_N		Fractional baud rate generator 1 reset control	1
		0	Assert the FRG1 reset.	
		1	Clear the FRG1 reset.	
31:5	-	-	Reserved	1

8.6.26 Peripheral clock source select registers

The peripheral clock source select registers select function clock sources for the serial peripherals shown in the following list. The potential clock sources are the same for each peripheral. See Table 97.

- UART0 clock source select register (UART0CLKSEL, address 0x4004 8090).
- UART1 clock source select register (UART1CLKSEL, address 0x4004 8094).
- UART2 clock source select register (UART2CLKSEL, address 0x4004 8098).
- I^{2}C0 clock source select register (I2C0CLKSEL, address 0x4004 80A4).
- I3C clock source select register (I3CFCLKSEL, address 0x4004 80A8).
- I3C clock source select register (I3CSLOWTCCLKSEL, address 0x4004 80AC).
- I3C clock source select register (I3CSLOWCLKSEL, address 0x4004 80B0).
- SPI0 clock source select register (SPI0CLKSEL, address 0x4004 80B4).
- SPI1 clock source select register (SPI1CLKSEL, address 0x4004 80B8).

Table 97. Peripheral clock source select registers

Bit	Symbol	Value	Description	Reset value
2:0	SEL		Peripheral clock source	0x7
		0x0	FRO	
		0x1	Main clock	
		0x2	FRG0 clock	
		0x3	FRG1 clock	
		0x4	FRO_DIV = FRO / 2	
		0x5	Reserved	
		0x6	Reserved	
		0x7	None	
31:3	-	-	Reserved	-

8.6.27 FTM0INTTRIGDIV

Table 98. FTM0INTTRIGDIV register (FTM0INTTRIGDIV, address 0x4004 80BC) bit description

Bit	Symbol	Value	Description	Reset value
6:0	FTM0_INT_TRIG_ DIV		ftm0 ext_trigger_ftm0 or init_trigger_ftm0 divided by 1~127	0
			CLKOUT clock divider values	
		0	Disable FTM internal trigger divider	
		1	CLKOUT divided by 1	
		2	CLKOUT divided by 2	
		127	CLKOUT divided by 127	
31:7	-	-	Reserved	0

8.6.28 FTM fault configuration register

Table 99. FTMFLTCFG register (FTMFLTCFG, address 0x4004 80C8) bit description

Bit	Symbol	Value	Description	Reset value
0	FTM0_FAULT0_MUX_SEL		Mux selection FTM0 fault from external I/O or internal.	0
		0	Select external I/O	
		1	Select internal	
1	FTM0_FAULT1_MUX_SEL		Mux selection FTM0 fault from external IO or internal.	0
		0	Select external I/O	
		1	Select internal	
2	FTM0_FAULT2_MUX_SEL		Mux selection FTM0 fault from external IO or internal.	0
		0	Select external I/O	
		1	Select internal	
3	FTM0_FAULT3_MUX_SEL		Mux selection FTM0 fault from external IO or internal.	0
		0	Select external I/O	
		1	Select internal	
4	FTM0_SW_FAULT		Software fault	0
31:5	-	-	Reserved	0

8.6.29 Fractional generator 0 divider value register

The UART, I^2C, SPI clock come from the FCLK multiplexer. The FRGCLK0 is one clock source of the FCLK multiplexer and its output from the fractional generator 0 can be adjusted by a fractional divider:

frg0clk = frg0_src_clk/(1 + MULT/DIV).

FRG0_SRC_CLK is input clock of fractional generator 0, which can be the FRO, main clock, or sys pll clock.

The fractional portion (1 + MULT/DIV) is determined by the two fractional divider registers in the SYSCON block:

- The DIV value programmed in this register is the denominator of the divider used by the fractional rate generator to create the fractional component of FRG0CLK.
- The MULT value of the fractional divider is programmed in the FRG0MULT register. See Table 101.

Remark: To use of the fractional baud rate generator, you must write 0xFF to this register to yield a denominator value of 256. All other values are not supported.

See also:

Section 18.3.1 “Configure the USART clock and baud rate”.

Section 18.7.1 “Clocking and baud rates”.

Table 100. Fractional generator 0 divider value register (FRG0DIV, address 0x4004 80D0) bit description

Bit	Symbol	Description	Reset value
7:0	DIV	Denominator of the fractional divider. DIV is equal to the programmed value +1. Always set to 0xFF to use with the fractional baud rate generator.	0
31:8	-	Reserved	-

8.6.30 Fractional generator 0 multiplier value register

The UART, I^{2}C, and SPI clocks come from the FCLK multiplexer. The FRG0CLK is one clock source of the FCLK multiplexer and its output from the fractional generator 0 can be adjusted by a fractional divider:

frg0clk = frg0_src_clk/(1 + MULT/DIV).

FRG0_SRC_CLK is input clock of fractional generator 0, which can be the FRO, main clock, or sys pll clock.

The fractional portion (1 + MULT/DIV) is determined by the two fractional divider registers in the SYSCON block:

- The DIV denominator of the fractional divider value is programmed in the FRG0DIV register. See Table 100.
- The MULT value programmed in this register is the numerator of the fractional divider value used by the fractional rate generator to create the fractional component to the baud rate.

Remark: To use of the fractional baud rate generator, you must write 0xFF to this register to yield a denominator value of 256. All other values are not supported.

See also:

Section 18.3.1 “Configure the USART clock and baud rate”.

Section 18.7.1 “Clocking and baud rates”.

Table 101. Fractional generator 0 multiplier value register (FRG0MULT, address 0x4004 80D4) bit description

Bit	Symbol	Description	Reset value
7:0	MULT	Numerator of the fractional divider. MULT is equal to the programmed value.	0
31:8	-	Reserved	-

8.6.31 FRG0 clock source select register

The FRG0CLKSEL register selects the frg0_src clock, which can be the FRO, main clock, or sys_pll.

Table 102. FRG0 clock source select register (FRG0CLKSEL, address 0x4004 80D8) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		Clock source for FRG0_SRC clock	0x0
		0x0	FRO	
		0x1	Main clock	
		0x2	SYS PLL	
		0x3	None	
31:2	-	-	Reserved	-

8.6.32 Fractional generator 1 divider value register

The UART, I2C, SPI clock come from the FCLK multiplexer. The FRG1CLK is one clock source of the FCLK multiplexer and its output from the fractional generator 1 can be adjusted by a fractional divider:

frg1clk = frg1_src_clk/(1 + MULT / DIV)

FRG1_SRC_CLK is input clock of fractional generator 1, which can be the FRO, main clock, or sys pll clock.

The fractional portion (1 + MULT / DIV) is determined by the two fractional divider registers in the SYSCON block:

- The DIV value programmed in this register is the denominator of the divider used by the fractional rate generator to create the fractional component of FRG1CLK.
- The MULT value of the fractional divider is programmed in the FRG1MULT register. See Table 104.

Remark: To use the fractional baud rate generator, you must write 0xFF to this register to yield a denominator value of 256. All other values are not supported.

See also:

Section 18.3.1 “Configure the USART clock and baud rate”.

Section 18.7.1 “Clocking and baud rates”.

Table 103. Fractional generator 1 divider value register (FRG1DIV, address 0x4004 80E0) bit description

Bit	Symbol	Value	Description	Reset value
7:0	DIV		Denominator of the fractional divider. DIV is equal to the programmed value +1. Always set to 0xFF to use with the fractional baud rate generator.	0
31:8	-		Reserved	-

8.6.33 Fractional generator 1 multiplier value register

The UART, I^{2}C, and SPI clocks come from the FCLK multiplexer. The FRG1CLK is one clock source of the FCLK multiplexer and its output from the fractional generator 1 can be adjusted by a fractional divider:

frg1clk = frg1_src_clk/(1 + MULT/DIV).

FRG1_SRC_CLK is input clock of fractional generator 1, which can be the FRO, main clock, or sys pll clock.

The fractional portion (1 + MULT/DIV) is determined by the two fractional divider registers in the SYSCON block:

- The DIV denominator of the fractional divider value is programmed in the FRG1DIV register. See Table 103.
- The MULT value programmed in this register is the numerator of the fractional divider value used by the fractional rate generator to create the fractional component to the baud rate.

See also:

Section 18.3.1 “Configure the USART clock and baud rate”.

Section 18.7.1 “Clocking and baud rates”.

Table 104. Fractional generator 1 multiplier value register (FRG1MULT, address 0x4004 80E4) bit description

Bit	Symbol	Value	Description	Reset value
7:0	MULT		Numerator of the fractional divider. MULT is equal to the programmed value.	0
31:8	-		Reserved	-

8.6.34 FRG1 clock source select register

The FRG1CLKSEL register selects the frg1_src clock, which can be the FRO, main clock, or sys_pll.

Table 105. FRG1 clock source select register (FRG1CLKSEL, address 0x4004 80E8) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SEL		Clock source for FRG1_SRC clock	0x0
		0x0	FRO	
		0x1	Main clock	
		0x2	SYS PLL	
		0x3	None	
31:2	-	-	Reserved	-

8.6.35 CLKOUT clock source select register

This register selects the signal visible on the CLKOUT pin. Any oscillator or the main clock can be selected.

Table 106. CLKOUT clock source select register (CLKOUTSEL, address 0x4004 80F0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	SEL		CLKOUT clock source	0
		0x0	FRO	
		0x1	Main clock	
		0x2	SYS PLL	
		0x3	External clock	
		0x4	Low power oscillator	
		0x5	None	
		0x6	None	
		0x7	None	
31:3	-	-	Reserved	0

8.6.36 CLKOUT clock divider register

The **CLKOUTDIV** register determines the divider value for the signal on the CLKOUT pin.

Table 107. CLKOUT clock divider registers (CLKOUTDIV, address 0x4004 80F4) bit description

Bit	Symbol	Description	Reset value
7:0	DIV	CLKOUT clock divider values 0: Disable CLKOUT clock divider. 1: Divide by 1. … 255: Divide by 255.	0
31:8	-	Reserved	-

8.6.37 IOCON glitch filter clock divider registers 6 to 0

These registers individually configure the seven peripheral input clocks (IOCONFILTR_PCLK) to the IOCON programmable glitch filter. The clocks can be shut down by setting the DIV bits to 0x0.

Table 108. IOCON glitch filter clock divider registers 6 to 0 (IOCONCLKDIV[6:0], address 0x4004 8134 (IOCONCLKDIV6) to 0x004 814C (IOCONFILTCLKDIV0)) bit description

Bit	Symbol	Description	Reset value
7:0	DIV	IOCON glitch filter clock divider values 0: Disable IOCONFILTR_PCLK. 1: Divide by 1. … 255: Divide by 255.	0
31:8	-	Reserved	0x00

8.6.38 BOD control register

The BOD control register selects four separate threshold values for sending a BOD interrupt to the NVIC and for forced reset. Reset and interrupt threshold values listed in Table 109 are typical values.

Both the BOD interrupt and the BOD reset, depending on the value of bit BODRSTENA in this register, can wake-up the chip from sleep, deep-sleep, and power-down modes.

See the LPC86x data sheet for the BOD reset and interrupt levels.

Note: Startup time of 30us maximum is needed after power up BOD and before enabling BOD reset/interrupt.

Table 109. BOD control register (BODCTRL, address 0x4004 8150) bit description

Bit	Symbol	Value	Description	Reset value
1:0	BODRSTLEV		BOD reset level	0
		0x0	Reserved.	
		0x1	Level 1.	
		0x2	Level 2.	
		0x3	Level 3.	
3:2	BODINTVAL		BOD interrupt level	0
		0x0	Reserved	
		0x1	Level 1.	
		0x2	Level 2.	
		0x3	Level 3.	
4	BODRSTENA		BOD reset enable	1
		0	Disable reset function.	
		1	Enable reset function.	
31:5	-	-	Reserved	0x00

8.6.39 System tick counter calibration register

This register determines the value of the SYST_CALIB register.

Table 110. System tick timer calibration register (SYSTCKCAL, address 0x4004 8154) bit description

Bit	Symbol	Description	Reset value
25:0	CAL	System tick timer calibration value	0
31:26	-	Reserved	-

8.6.40 IRQ latency register

The IRQLATENCY register is an eight-bit register which specifies the minimum number of cycles (0-255) permitted for the system to respond to an interrupt request. The intent of this register is to allow the user to select a trade-off between interrupt response time and determinism.

Setting this parameter to a very low value (e.g. zero) will guarantee the best possible interrupt performance but will also introduce a significant degree of uncertainty and jitter. Requiring the system to always take a larger number of cycles (whether it needs it or not) will reduce the amount of uncertainty but may not necessarily eliminate it.

Theoretically, the ARM Cortex-M0+ core should always be able to service an interrupt request within 15 cycles. However, system factors external to the CPU, such as bus latencies or peripheral response times, can increase the time required to complete a previous instruction before an interrupt can be serviced. Therefore, accurately specifying a minimum number of cycles that will ensure determinism will depend on the application.

The default setting for this register is 0x010.

Table 111. IRQ latency register (IRQLATENCY, address 0x4004 8170) bit description

Bit	Symbol	Description	Reset value
7:0	LATENCY	8-bit latency value	0x010
31:8	-	Reserved	-

8.6.41 NMI source selection register

The NMI source selection register selects a peripheral interrupt as source for the NMI interrupt of the ARM Cortex-M0+ core. For a list of all peripheral interrupts and their IRQ numbers see Table 46. For a description of the NMI functionality, see Section 6.3.2.

Remark: When you want to change the interrupt source for the NMI, you must first disable the NMI source by setting bit 31 in this register to 0. Then change the source by updating the IRQN bits and re-enable the NMI source by setting bit 31 to 1.

Table 112. NMI source selection register (NMISRC, address 0x4004 8174) bit description

Bit	Symbol	Description	Reset value
4:0	IRQN	The IRQ number of the interrupt that acts as the Non-Maskable Interrupt (NMI) if bit 31 is 1. See Table 46 for the list of interrupt sources and their IRQ numbers.	0
30:5	-	Reserved	-
31	NMIEN	Write a 1 to this bit to enable the Non-Maskable Interrupt (NMI) source selected by bits 4:0.	0

Remark: If the NMISRC register is used to select an interrupt as the source of Non-Maskable interrupts, and the selected interrupt is enabled, one interrupt request can result in both a Non-Maskable and a normal interrupt. This can be avoided by disabling the normal interrupt in the NVIC.

8.6.42 Pin interrupt select registers

Each of these 8 registers selects one pin from all digital pins as the source of a pin interrupt or as the input to the pattern match engine. To select a pin for any of the eight pin interrupts or pattern match engine inputs, write the GPIO port pin number as 0 to 31 for pins PIO0_0 to PIO0_31 to the INTPIN bits and GPIO port pin number as 32 to 63 for pins PIO1_0 to PIO1_31. For example, setting INTPIN to 0x5 in PINTSEL0 selects pin PIO0_5 for pin interrupt 0.

Remark: The GPIO port pin number serves to identify the pin to the PINTSEL register. Any digital input function, including GPIO, can be assigned to this pin through the switch matrix.

Each of the 8 pin interrupts must be enabled in the NVIC using interrupt slots # 24 to 31 (see Table 46).

To use the selected pins for pin interrupts or the pattern match engine, see Section 14.5.2 "Pattern match engine".

Table 113. Pin interrupt select registers (PINTSEL[0:7], address 0x4004 8178 (PINTSEL0) to 0x4004 8194 (PINTSEL7)) bit description

Bit	Symbol	Description	Reset value
5:0	INTPIN	Pin number select for pin interrupt or pattern match engine input. (PIO0_0 to PIO0_31correspond to numbers 0 to 31 and PIO1_0 to PIO1_31 correspond to numbers 32 to 63).	0
31:6	-	Reserved	-

8.6.43 Start logic 0 pin wake-up enable register

The STARTERP0 register enables the selected pin interrupts for wake-up from deep-sleep mode and power-down modes.

Remark: Also enable the corresponding interrupts in the NVIC. See Table 46 "Connection of interrupt sources to the NVIC".

Table 114. Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description

Bit	Symbol	Value	Description	Reset value
0	PINT0		GPIO pin interrupt 0 wake-up	0
		0	Disabled	
		1	Enabled	
1	PINT1		GPIO pin interrupt 1 wake-up	0
		0	Disabled	
		1	Enabled	
2	PINT2		GPIO pin interrupt 2 wake-up	0
		0	Disabled	
		1	Enabled	
3	PINT3		GPIO pin interrupt 3 wake-up	0
		0	Disabled	
		1	Enabled	
4	PINT4		GPIO pin interrupt 4 wake-up	0
		0	Disabled	
		1	Enabled	
5	PINT5		GPIO pin interrupt 5 wake-up	0
		0	Disabled	
		1	Enabled	

Table 114. Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
6	PINT6		GPIO pin interrupt 6 wake-up	0
		0	Disabled	
		1	Enabled	
7	PINT7		GPIO pin interrupt 7 wake-up	0
		0	Disabled	
		1	Enabled	
31:8	-		Reserved	-

8.6.44 Start logic 1 interrupt wake-up enable register

This register selects which interrupts wake up the part from deep-sleep and power-down modes.

Remark: Also enable the corresponding interrupts in the NVIC. See Table 46 "Connection of interrupt sources to the NVIC".

Table 115. Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description

Bit	Symbol	Value	Description	Reset value
0	SPI0		SPI0 interrupt wake-up	0
		0	Disabled	
		1	Enabled	
1	SPI1		SPI1 interrupt wake-up	0
		0	Disabled	
		1	Enabled	
2	-		Reserved	-
3	USART0		USART0 interrupt wake-up. Configure USART in synchronous slave mode.	0
		0	Disabled	
		1	Enabled	
4	USART1		USART1 interrupt wake-up. Configure USART in synchronous slave mode.	0
		0	Disabled	
		1	Enabled	
5	USART2		USART2 interrupt wake-up. Configure USART in synchronous slave mode.	0
		0	Disabled	
		1	Enabled	
7:6	-		Reserved	-
8	I2C0		I2C0 interrupt wake-up.	0
		0	Disabled	
		1	Enabled	

Table 115. Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
11:9	-		Reserved	0
12	WWDT		WWDT interrupt wake-up	0
		0	Disabled	
		1	Enabled	
13	BOD		BOD interrupt wake-up	0
		0	Disabled	
		1	Enabled	
14	-		Reserved	-
15	WKT		Self-wake-up timer interrupt wake-up	0
		0	Disabled	
		1	Enabled	
31:16	-		Reserved.	-

8.6.45 Deep-sleep mode configuration register

The bits in this register (BOD_PD and WDTOSC_OD) can be programmed to control aspects of Deep-sleep and Power-down modes. The bits are loaded into corresponding bits of the PDRUNCFG register when Deep-sleep mode or Power-down mode is entered.

Remark: Hardware forces the analog blocks to be powered down in Deep-sleep and Power-down modes. An exception are the BOD and low power oscillator, which can be configured to remain running through this register. The WDTOSC_PD value written to the PDSLEEPCFG register is overwritten if the LOCK bit in the WWDT MOD register (see Table 377) is set. See Section 22.6.1 for details.

Table 116. Deep-sleep configuration register (PDSLEEPCFG, address 0x4004 8230) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0b111
3	BOD_PD		BOD power-down control for Deep-sleep and Power-down mode	1
		0	Powered	
		1	Powered down	
5:4	-		Reserved.	11
6	LPOSC_PD		Low power oscillator power-down control for Deep-sleep and Power-down mode. Changing this bit to powered-down has no effect when the LOCK bit in the WWDT MOD register is set. In this case, the low power oscillator is always running.	1
		0	Powered	
		1	Powered down	
15:7	-		Reserved	0b111111111
31:16	-	-	Reserved	0

8.6.46 Wake-up configuration register

This register controls the power configuration of the device when waking up from Deep-sleep or Power-down mode.

Table 117. Wake-up configuration register (PDAWAKECFG, address 0x4004 8234) bit description

Bit	Symbol	Value	Description	Reset value
0	FROOUT_PD		FRO oscillator output wake-up configuration	0
		0	Powered	
		1	Powered down	
1	FRO_PD		FRO oscillator power-down wake-up configuration	0
		0	Powered	
		1	Powered down	
2	FLASH_PD		Flash wake-up configuration	0
		0	Powered	
		1	Powered down	
3	BOD_PD		BOD wake-up configuration	0
		0	Powered	
		1	Powered down	
4	ADC_PD		ADC wake-up configuration	1
		0	Powered	
		1	Powered down	
5	SYSOSC_PD		Crystal oscillator wake-up configuration	1
		0	Powered	
		1	Powered down	
6	LPOSC_PD		Low power oscillator wake-up configuration. Changing this bit to powered-down has no effect when the LOCK bit in the WWDT MOD register is set. In this case, the low power oscillator is always running.	1
		0	Powered	
		1	Powered down	
7	SYSPLL_PD		System PLL wake-up configuration	1
		0	Powered	
		1	Powered down	
9:8	-		Reserved. Always write these bits as 0b01	0b01
10	-		Reserved	1
12:11	-		Reserved. Always write these bits as 0b01	0b01
14:13	-		Reserved	
15	ACMP		Analog comparator wake-up configuration	1
		0	Powered	
		1	Powered down	
31:16	-	-	Reserved	0

8.6.47 Power configuration register

The PDRUNCFG register controls the power to the various analog blocks. This register can be written to at any time while the chip is running, and a write will take effect immediately with the exception of the power-down signal to the FRO.

To avoid glitches when powering down the FRO, the FRO clock is automatically switched off at a clean point. Therefore, for the FRO a delay is possible before the power-down state takes effect.

The system oscillator requires typically 500 µs to start up after the SYSOSC_PD bit has been changed from 1 to 0. There is no hardware flag to monitor the state of the system oscillator. Therefore, add a software delay of about 500 µs before using the system oscillator after power-up.

Table 118. Power configuration register (PDRUNCFG, address 0x4004 8238) bit description

Bit	Symbol	Value	Description	Reset value
0	FROOUT_PD		FRO oscillator output power	0
		0	Powered	
		1	Powered down	
1	FRO_PD		FRO oscillator power down	0
		0	Powered	
		1	Powered down	
2	FLASH_PD		Flash power down	0
		0	Powered	
		1	Powered down	
3	BOD_PD		BOD power down	0
		0	Powered	
		1	Powered down	
4	ADC_PD		ADC wake-up configuration	1
		0	Powered	
		1	Powered down	
5	SYSOSC_PD		Crystal oscillator power down. After power-up, add a software delay of approximately 500 µs before using.	1
		0	Powered	
		1	Powered down	
6	LPOSC_PD		Low power oscillator power down. Changing this bit to powered-down has no effect when the LOCK bit in the WWDT MOD register is set. In this case, the low power oscillator is always running.	1
		0	Powered	
		1	Powered down	
7	SYSPLL_PD		System PLL power down	1
		0	Powered	
		1	Powered down	
9:8	-		Reserved. Always write these bits as 0b01	0b01

Table 118. Power configuration register (PDRUNCFG, address 0x4004 8238) bit description

Bit	Symbol	Value	Description	Reset value
14:10	-		Reserved. Always write these bits as 0b00011	0b11011
15	ACMP		Analog comparator power down	1
		0	Powered	
		1	Powered down	
31:16	-	-	Reserved	0

8.6.48 Flash cache configuration register

Table 119. Flash cache configuration register (FLASHCACHECFG, address 0x4004 8240) bit description

Bit	Symbol	Value	Description	Reset value
0	FLASH_CACHE_ ENABLE		Flash cache enable	1
		0	Disable	
		1	Enable	
1	FLASH_BUFFER _ENABLE		Flash buffer enable	1
		0	Disable	
		1	Enable	
2	FLASH_CACHE_ INVALID		Flash cache invalid	0
		0	Not invalid	
		1	Invalid	
31:3	-	-	Reserved	0

8.6.49 Device ID register

The device ID register is a read-only register and contains the part ID for each part. This register is also read by the ISP/IAP commands (see Table 120).

Table 120. Device ID register values

Part number	Part ID
LPC865M201JBD64	0x00008651
LPC865M201JHI48	0x00008652
LPC865M201JHI33	0x00008654

8.7 Functional description

8.7.1 Reset

Reset has the following sources: the $\overline{\text{RESET}}$ pin, Watchdog Reset, Power-On Reset (POR), and Brown Out Detect (BOD). In addition, there is an ARM software reset.

The $\overline{\text{RESET}}$ pin is a Schmitt trigger input pin. Assertion of chip Reset by any source, once the operating voltage attains a usable level, starts the FRO causing reset to remain asserted until the external Reset is de-asserted, the oscillator is running, and the flash controller has completed its initialization.

When the internal Reset is removed, the processor begins executing at address 0, which is initially the Reset vector mapped from the boot block. At that point, all of the processor and peripheral registers have been initialized to predetermined values.

8.7.2 Brown-out detection

The brown-out detection circuit includes up to three levels for monitoring the voltage on the V_{DD} pin. If this voltage falls below one of the selected levels, the BOD asserts an interrupt signal to the NVIC or issues a reset, depending on the value of the BODRSTENA bit in the BOD control register (Table 109).

The interrupt signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC (see Table 47) in order to cause a CPU interrupt; if not, software can monitor the signal by reading a dedicated status register.

If the BOD interrupt is enabled in the STARTERP1 register (see Table 115) and in the NVIC, the BOD interrupt can wake up the chip from Deep-sleep and power-down mode.

If the BOD reset is enabled, the forced BOD reset can wake up the chip from Deep-sleep or Power-down mode.

8.7.3 System PLL functional description

The LPC86x uses the system PLL to create the clocks for the core and peripherals.

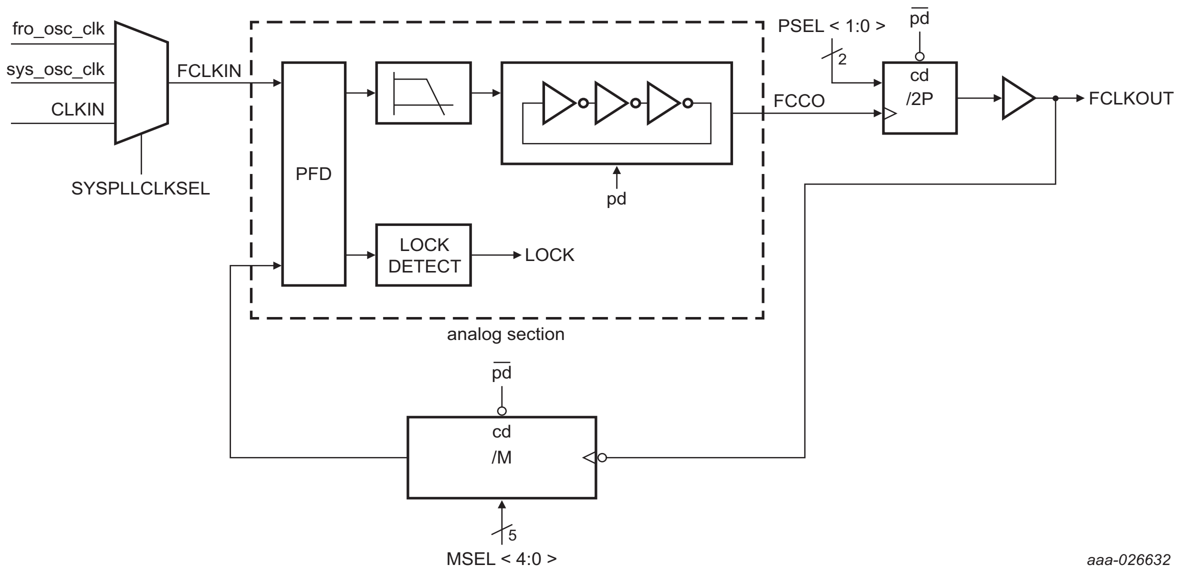


Fig 9. System PLL block diagram

The block diagram of this PLL is shown in Figure 9. The input frequency range is 10 MHz to 25 MHz. The input clock is fed directly to the Phase-Frequency Detector (PFD). This block compares the phase and frequency of its inputs, and generates a control signal when phase and/ or frequency do not match. The loop filter filters these control signals and drives the current controlled oscillator (CCO), which generates the CCO clock. The CCO frequency range is 156 MHz to 320 MHz. This clock is divided by 2xP by the programmable post divider to create the output clock. The output clock is then divided by M by the programmable feedback divider to generate the feedback clock. The output signal of the phase-frequency detector is also monitored by the lock detector, to signal when the PLL has locked on to the input clock.

Remark: The divider values for P and M must be selected so that the PLL output clock frequency FCLKOUT is lower than 30 MHz because the main clock is limited to a maximum frequency of 60 MHz.

8.7.3.1 Lock detector

The lock detector measures the phase difference between the rising edges of the input and feedback clocks. Only when this difference is smaller than the so called “lock criterion” for more than eight consecutive input clock periods, the lock output switches from low to high. A single too large phase difference immediately resets the counter and causes the lock signal to drop (if it was high). Requiring eight phase measurements in a row to be below a certain figure ensures that the lock detector will not indicate lock until both the phase and frequency of the input and feedback clocks are very well aligned. This effectively prevents false lock indications, and thus ensures a glitch free lock signal.

8.7.3.2 Power-down control

To reduce the power consumption when the PLL clock is not needed, a PLL Power-down mode has been incorporated. This mode is enabled by setting the SYSPLL_PD bit to one in the Power-down configuration register (Table 118). In this mode, the internal current reference will be turned off, the oscillator and the phase-frequency detector will be stopped and the dividers will enter a reset state. While in PLL Power-down mode, the lock output will be low to indicate that the PLL is not in lock. When the PLL Power-down mode is terminated by setting the SYSPLL_PD bit to zero, the PLL will resume its normal operation and will make the lock signal high once it has regained lock on the input clock.

8.7.3.3 Divider ratio programming

8.7.3.3.1 Post divider

The division ratio of the post divider is controlled by the PSEL bits. The division ratio is two times the value of P selected by PSEL bits as shown in Table 73. This guarantees an output clock with a 50% duty cycle.

8.7.3.3.2 Feedback divider

The feedback divider's division ratio is controlled by the MSEL bits. The division ratio between the PLL's output clock and the input clock is the decimal value on MSEL bits plus one, as specified in Table 73.

8.7.3.3.3 Changing the divider values

Changing the divider ratio while the PLL is running is not recommended. As there is no way to synchronize the change of the MSEL and PSEL values with the dividers, the risk exists that the counter will read in an undefined value, which could lead to unwanted spikes or drops in the frequency of the output clock. The recommended way of changing between divider settings is to power down the PLL, adjust the divider settings and then let the PLL start up again.

8.7.3.4 Frequency selection

The PLL frequency equations use the following parameters (also see Figure 9):

Table 121. PLL frequency parameters

Parameter	System PLL
FCLKIN	Frequency of sys_pllclkin (input clock to the system PLL) from the SYSPLLCLKSEL multiplexer (see Section 8.6.9).
FCCO	Frequency of the Current Controlled Oscillator (CCO); 156 to 320 MHz.
FCLKOUT	Frequency of sys_pllclkout. This is the PLL output frequency and must be $\leq$ 30 MHz.
P	System PLL post divider ratio; PSEL bits in SYSPLLCTRL (see Section 8.6.2).
M	System PLL feedback divider register; MSEL bits in SYSPLLCTRL (see Section 8.6.2).

8.7.3.4.1 Normal mode

In this mode the PLL is enabled, giving a 50% duty cycle clock with the following frequency relations:

$$Fclkout = M \times Fclkin = (FCCO)/(2 \times P) \quad (1)$$

To select the appropriate values for M and P, it is recommended to follow these steps:

1. Specify the input clock frequency Fclkin.
2. Calculate M to obtain the desired output frequency Fclkout with M = F_{clkout} / F_{clkin}.
3. Find a value so that FCCO = 2 × P × F_{clkout}.
4. Verify that all frequencies and divider values conform to the limits specified in Table 73.

Remark: The divider values for P and M must be selected so that the PLL output clock frequency FCLKOUT is ≤ 60 MHz.

Table 122 shows how to configure the PLL for a 12 MHz crystal oscillator using the SYSPLLCTRL register (Table 73).

Table 122. PLL configuration examples

FCLKIN frequency	FCLKOUT frequency	MSEL bits Table 73	M divider value	PSEL bits Table 73	P divider value	FCCO frequency	System clock
12 MHz	24 MHz	00001 (binary)	2	10 (binary)	4	192 MHz	24 MHz

8.7.3.4.2 PLL Power-down mode

In this mode, the internal current reference will be turned off, the oscillator and the phase-frequency detector will be stopped and the dividers will enter a reset state. While in PLL Power-down mode, the lock output will be low, to indicate that the PLL is not in lock. When the PLL Power-down mode is terminated by SYSPLL_PD bit to zero in the Power-down configuration register (Table 118), the PLL will resume its normal operation and will make the lock signal high once it has regained lock on the input clock.

Chapter 9: Peripheral Bridge (AIPS_Lite)

9.1 How to read this chapter

The AIPS_Lite module is available on all LPC86x devices.

9.2 Introduction

AIPS_Lite converts the ARM AHB interface to an interface that can access most of the slave peripherals on this chip.

This peripheral bridge occupies 64 MB of the address space, which is divided into peripheral slots of 4 KB each. All the peripherals may not be used. See the memory mapping chapter for details on slot assignments. The bridge includes separate clock enable inputs for each of the slots to accommodate slower peripherals.

This device includes one instance of the AIPS module, AIPS0.

9.2.1 Features

The peripheral bridge supports peripheral slots with 8-, 16-, and 32-bit datapath width.

9.2.2 General operation

The slave devices connected to the peripheral bridge are modules that contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge.

The register maps of the peripherals are located on 4-KB boundaries. Each peripheral is allocated one or more 4-KB block(s) of the memory map.

9.3 Memory map and register definition

The AIPS module(s) on this chip do(es) not contain any user-programmable registers.

9.4 Functional description

The peripheral bridge functions as a bus protocol translator between the ARM AHB interface and the slave peripheral bus. The AIPS-Lite bridge on this device provides access to the FlexTimers, FTM0 and FTM1.

The peripheral bridge manages all transactions for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

9.4.1 Access support

Aligned and misaligned 32-bit, 16-bit, and byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the slave peripheral bus. Peripheral registers must not be misaligned, although no explicit checking is performed by the peripheral bridge. All accesses are performed with a single transfer.

All accesses to the peripheral slots must be sized less than or equal to the designated peripheral slot size. If an access is attempted that is larger than the targeted port, an error response is generated.

UM11607

Chapter 10: FRO API ROM routine

Rev. 3 — April 2023 **User manual**

10.1 How to read this chapter

The ROM-based set fro_oscout API call is available on all parts.

10.2 Features

- Select desired on-chip fro_oscout (60 MHz/48 MHz/36 MHz).

10.3 Basic configuration

Control of FRO output frequency can be configured through a simple call to the ROM.

The set_fro_frequency API call must be used to the select desired fro_oscout (60 MHz/ 48 MHz/36 MHz). This is performed by executing a function, which is pointed by a pointer within the ROM Driver Table. Figure 10 shows the pointer structure used to call the set FRO frequency API.

Remark: Disable all interrupts before making calls to the FRO API. The interrupts can be re-enabled after the FRO API call is completed.

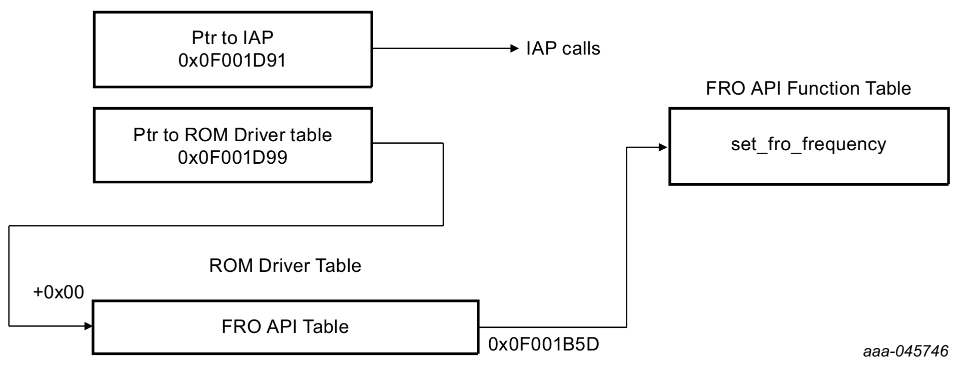


Fig 10. ROM pointer structure

10.4 API description

The FRO API provides a function to configure the fro_oscout. The FRO API can be called in the application code through a simple API call. See an API example below:

```
// Pointer to set ROM API entry functions
#define ROM_ENTRY_LOCATION 0x0F001D98   /* Thumb code needs to +1 */

#define FREQ_36000KHZ 36000

typedef struct _PWRD {
  void (*set_fro_frequency)(unsigned frequency);
}  PWRD;

typedef struct _ROM {
   const PWRD * pPWRD;
}  ROM;

You can call this API by:

    ROM **rom;

    rom = (ROM **)ROM_ENTRY_LOCATION;
    (*rom)->pPWRD->set_fro_frequency(FREQ_36000KHZ);
```

The following function prototypes are used:

Table 123. FRO API call

Function prototype	API description	Reference
`void set_fro_frequency(uint32_t iFreq);`	Setup the fro_oscout to either 60 MHz, 48 MHz or 36 MHz.	Section 10.4.1

10.4.1 set_fro_frequency

This routine sets up the required fro_oscout (60 MHz/48 MHz/36 MHz). The requested frequency is set up and the appropriate factory trim value is used.

See Figure 8 “Clock generation (continued)” for more details to select fro_oscout (60/48/36 MHz).

Table 124 shows the set_fro_frequency routine.

Table 124. set_fro_frequency routine

Routine	sidiv
Prototype	void set_fro_frequency(uint32_t iFreq);
Input parameter	Param0 Required frequency (in kHz). Example: parameter 1: 36000 => 36 MHz, 48000=>48 Mhz, 60000=>60 Mhz.
Return	None.
Description	Setup the fro_oscout to either 60 MHz, 48 MHz, or 36MHz.

10.4.1.1 Param0: frequency

The frequency is the required fro_oscout, 60 MHz, 48 MHz, or 36 MHz.

UM11607

Chapter 11: Switch matrix (SWM)

Rev. 3 — April 2023 **User manual**

11.1 How to read this chapter

The switch matrix is identical for all LPC86x parts.

11.2 Features

- Flexible assignment of digital peripheral functions to pins
  - Most digital functions can be assigned to any GPIO pin.
  - Selected digital functions can be assigned to up to 3 GPIO pins.
- Enable/disable of analog functions

11.3 Basic configuration

Once configured, no clocks are needed for the switch matrix to function. The system clock is needed only to write to or read from the pin assignment registers. After the switch matrix is configured, disable the clock to the switch matrix block in the SYSAHBCLKCTRL register.

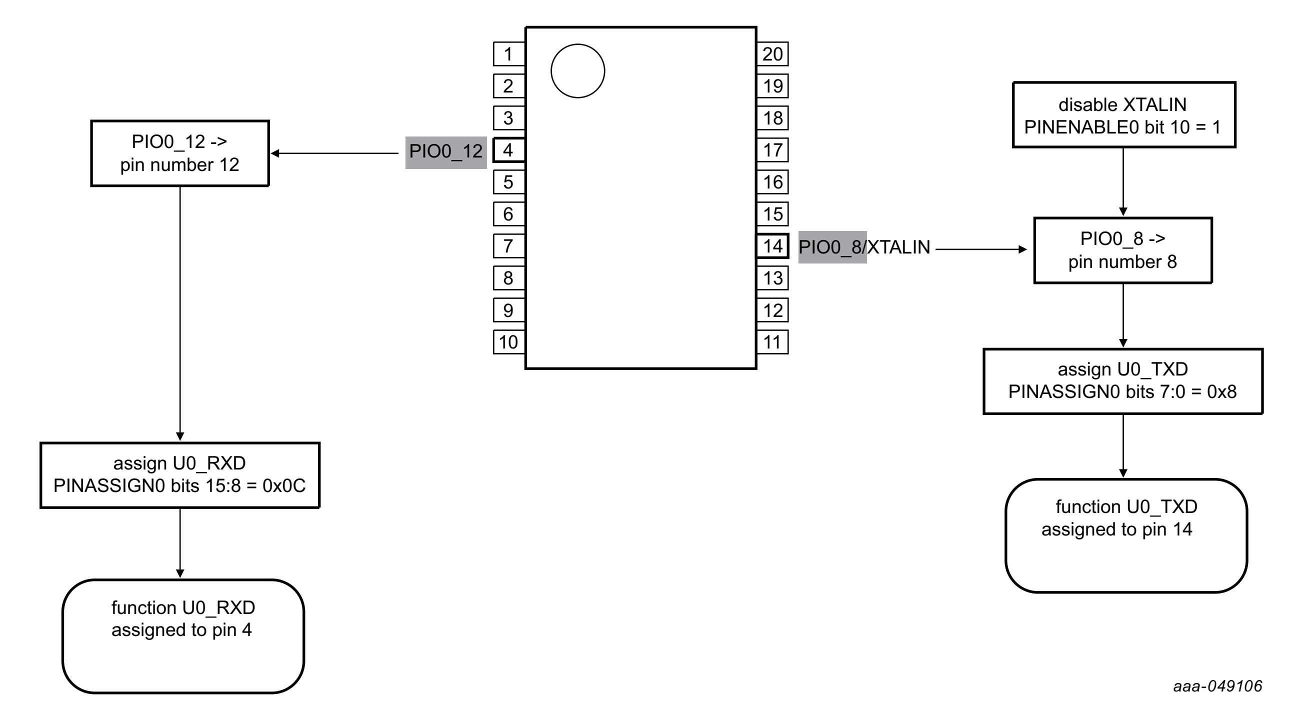
Before activating a peripheral or enabling its interrupt, use the switch matrix to connect the peripheral to external pins.

The serial wire debug pins SWDIO and SWCLK are enabled by default on pins PIO0_2 and PIO0_3.

Remark: For the purpose of programming the pin functions through the switch matrix, every pin except the power and ground pins is identified in a package-independent way by its GPIO port pin number.

Remark: The switch matrix is reset by a system reset from the $\overline{\text{RESET}}$ pin as well as all other resets.

11.3.1 Connect an internal signal to a package pin


1
2
3
4
5
6
7
8
9
10
20
19
18
17
16
15
14
13
12
11
PIO0_12
PIO0_12 -> pin number 12
assign U0_RXD PINASSIGN0 bits 15:8 = 0x0C
function U0_RXD assigned to pin 4
PIO0_8/XTALIN
disable XTALIN PINENABLE0 bit 10 = 1
PIO0_8 -> pin number 8
assign U0_TXD PINASSIGN0 bits 7:0 = 0x8
function U0_TXD assigned to pin 14
aaa-049106

A pin is identified for the purpose of programming the switch matrix by its default GPIO port pin number.

Fig 11. Example: Connect function U0_RXD and U0_TXD to pins 4 and 14

The switch matrix connects all internal signals listed in the table of movable functions through the pin assignment registers to external pins on the package. External pins are identified by their default GPIO pin number PIO0_n or PIO1_(n-32). Follow these steps to connect an internal signal FUNC to an external pin. An example of a movable function is the UART transmit signal TXD:

1. Find the pin function in the list of movable functions in Table 125 or in the data sheet.
2. Use the LPC86x data sheet to decide which pin x on the LPC86x package to connect the pin function to.
3. Use the pin description table to find the default GPIO function PIO0_n or PIO1_(n-32) assigned to package pin x. m is the pin number.
4. Locate the pin assignment register for the function FUNC in the switch matrix register description.
5. Disable any special functions on pin PIO0_n in the PINENABLE0 register or PIO1_(n-32) in the PINENABLE1 register.
6. Program the pin number n into the bits assigned to the pin function.

The pin function is now connected to pin x on the package.

11.3.2 Enable an analog input or other special function

The switch matrix enables functions that can only be assigned to one pin. Examples are analog inputs, all GPIO pins, and the debug SWD pins.

- If you want to assign a GPIO pin to a pin on any LPC86x package, disable any special function available on this pin in the PINENABLE0 or PINENABLE 1 register and do not assign any movable function to it.
- For all other functions that are not in the table of movable functions, do the following:
  a. Locate the function in the pin description table in the data sheet. This shows the package pin for this function.
  b. Enable the function in the PINENABLE0 PINENABLE1 register. All other possible functions on this pins are now disabled.

11.3.3 Changing the pin function assignment

Pin function assignments can be changed “on-the-fly” from one peripheral to another while the part is running. To disconnect a peripheral from the pins and change the pin function assignment, follow these steps:

1. Enable the clock to the switch matrix.
2. Find the pin assign register for the current pin function. For example, register PINASSIGN0 for pin function U0_RXD.
3. Set the corresponding bits in the PINASSIGN register to their default value 0xFF.
4. Clear all pending interrupts for the disconnected peripheral and ensure that the peripheral is in a defined state.
5. In the pin assign register for the new pin function, program the pin number.
6. Disable the clock to the switch matrix.

11.4 General description

The switch matrix connects internal signals (functions) to external pins. Functions are signals coming from or going to a single pin on the package and coming from or going to an on-chip peripheral block. Examples of functions are the GPIOs, the UART transmit output (TXD), or the clock output CLKOUT. Many peripherals have several functions that must be connected to external pins.

The switch matrix also enables the output driver for digital functions that are outputs. The electrical pin characteristics for both inputs and outputs (internal pull-up/down resistors, inverter, digital filter, open-drain mode) are configured by the IOCON block for each pin.

Most functions can be assigned through the switch matrix to any external pin that is not a power or ground pin. These functions are called movable functions.

Some digital functions are not fully movable. For these, up to three specific port pins can be selected for each function, or the function can be deselected allowing the port to be used for other movable functions, GPIO, or analog functions, as allowed for each pin.

A few functions like the crystal oscillator pins (XTALIN/XTALOUT) or the analog comparator inputs can only be assigned to one particular external pin with the appropriate electrical characteristics. These functions are called fixed-pin functions. If a fixed-pin function is not used, it can be replaced by any other movable function.

For fixed-pin analog functions, the switch matrix enables the analog input or output and disables the digital pad.

GPIOs are special fixed-pin functions. Each GPIO is assigned to one and only one external pin by default. External pins are therefore identified by their fixed-pin GPIO function. The level on a digital input is always reflected in the GPIO port register and in the pin interrupt/pattern match state, if selected, regardless of which (digital) function is assigned to the pin through the switch matrix.

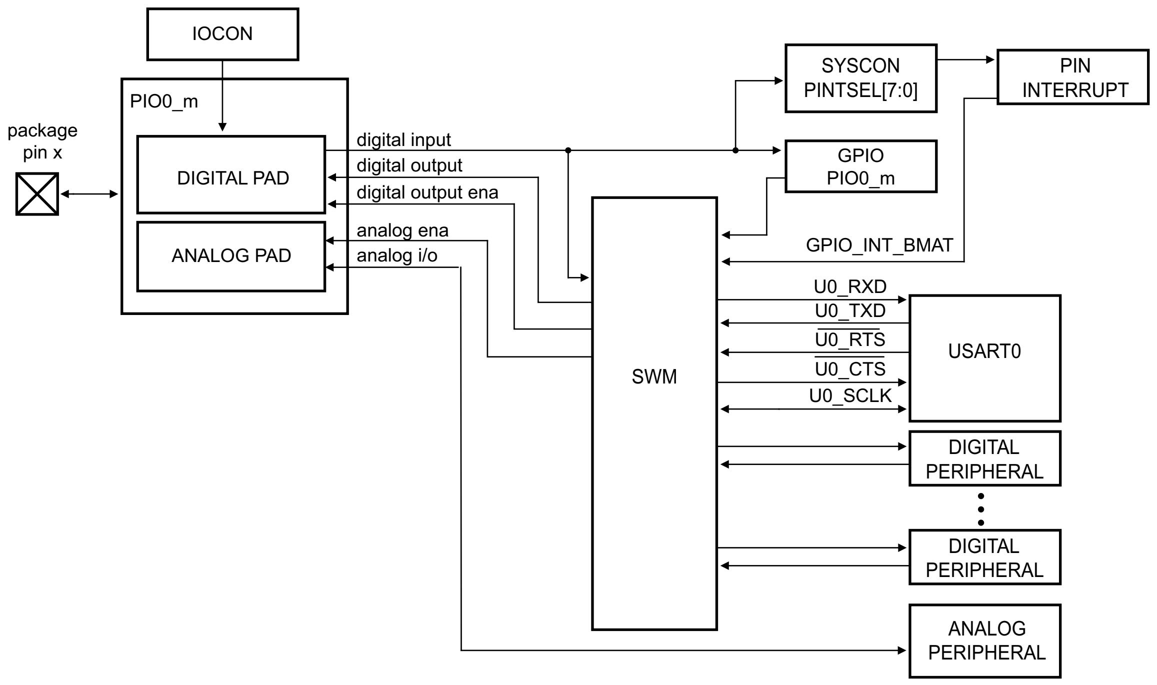


Fig 12. Functional diagram of the switch matrix

Remark: From all movable and fixed-pin functions, you can assign multiple functions to the same pin but no more than one output or bidirectional function (see Figure 12). Use the following guidelines when assigning pins:

- You can connect one digital input signal on a pin to multiple internal inputs by programming the same pin number in one or more PINASSIGN register.

  Example:

  You can enable the CLKIN input in the PINENABLE0 register on pin PIO0_1.
- It is allowed to let one digital output function control one or more digital inputs by programming the same pin number in the PINASSIGN register bit fields for the output and inputs.

  Example:

  You can loop back the USART transmit output to the receive input by assigning the same pin number to Un_RXD and Un_TXD.
- You cannot connect more than one output or bidirectional function to a pin.
- When you assign any function to a pin through the switch matrix, the GPIO output becomes disabled.
- Enabling any analog fixed-pin function disables all digital functions on the same pin.
- Enabling any digital fixed-pin function disables all analog pin function on the same pin.
- Digital and analog functions cannot share the same pin.

11.4.1 Movable functions

Table 125. Movable functions (assign to pins PIO0_0 to PIO0_31 and PIO1_0 to PIO1_21 through switch matrix)

Function name	Type	Description	SWM Pin assign register	Reference
U0_TXD	O	Transmitter output for USART0.	PINASSIGN0	Table 128
U0_RXD	I	Receiver input for USART0.	PINASSIGN0	Table 128
$\overline{\text{U0\_RTS}}$	O	Request To Send output for USART0.	PINASSIGN0	Table 128
$\overline{\text{U0\_CTS}}$	I	Clear To Send input for USART0.	PINASSIGN0	Table 128
U0_SCLK	I/O	Serial clock input/output for USART0 in synchronous mode.	PINASSIGN1	Table 129
U1_TXD	O	Transmitter output for USART1.	PINASSIGN1	Table 129
U1_RXD	I	Receiver input for USART1.	PINASSIGN1	Table 129
$\overline{\text{U1\_RTS}}$	O	Request To Send output for USART1.	PINASSIGN1	Table 129
$\overline{\text{U1\_CTS}}$	I	Clear To Send input for USART1.	PINASSIGN2	Table 130
U1_SCLK	I/O	Serial clock input/output for USART1 in synchronous mode.	PINASSIGN2	Table 130
U2_TXD	O	Transmitter output for USART2.	PINASSIGN2	Table 130
U2_RXD	I	Receiver input for USART2.	PINASSIGN2	Table 130
$\overline{\text{U2\_RTS}}$	O	Request To Send output for USART2.	PINASSIGN3	Table 131
$\overline{\text{U2\_CTS}}$	I	Clear To Send input for USART2.	PINASSIGN3	Table 131
U2_SCLK	I/O	Serial clock input/output for USART2 in synchronous mode.	PINASSIGN3	Table 131
SPI0_SCK	I/O	Serial clock for SPI0.	PINASSIGN3	Table 131
SPI0_MOSI	I/O	Master Out Slave In for SPI0.	PINASSIGN4	Table 132
SPI0_MISO	I/O	Master In Slave Out for SPI0.	PINASSIGN4	Table 132
SPI0_SSEL0	I/O	Slave select 0 for SPI0.	PINASSIGN4	Table 132
SPI0_SSEL1	I/O	Slave select 1 for SPI0.	PINASSIGN4	Table 132
SPI0_SSEL2	I/O	Slave select 2 for SPI0.	PINASSIGN5	Table 133
SPI0_SSEL3	I/O	Slave select 3 for SPI0.	PINASSIGN5	Table 133
SPI1_SCK	I/O	Serial clock for SPI1.	PINASSIGN5	Table 133
SPI1_MOSI	I/O	Master Out Slave In for SPI1.	PINASSIGN5	Table 133
SPI1_MISO	I/O	Master In Slave Out for SPI1.	PINASSIGN6	Table 134
SPI1_SSEL0	I/O	Slave select 0 for SPI1.	PINASSIGN6	Table 134
SPI1_SSEL1	I/O	Slave select 1 for SPI1.	PINASSIGN6	Table 134
I2C0_SDA	I/O	I^{2}C-bus data input/output.	PINASSIGN6	Table 134
I2C0_SCL	I/O	I^{2}C-bus clock input/output.	PINASSIGN7	Table 135
I3C0_SDA	I/O	I^{3}C-bus data input/output.	PINASSIGN7	Table 135
I3C0_SCL	I/O	I^{3}C-bus clock input/output.	PINASSIGN7	Table 135
I3C0_PUR	I/O		PINASSIGN7	Table 135
ACMP_O	O	Analog comparator output	PINASSIGN8	Table 136
CLKOUT	O	Clock output	PINASSIGN8	Table 136
GPIO_INT_BMAT	O	Output of the pattern match engine	PINASSIGN9	Table 136

Table 126. Flextimer pin assignments

Function name	Type	Selection 0	Selection 1	Selection 2	Selection 3
FTM0_EXTCLK	I	P0_24	P0_30	-	Not connected
FTM0_CH0	I/O	P0_17	P1_1	-	Not connected
FTM0_CH1	I/O	P0_18	P1_2	P0_16	Not connected
FTM0_CH2	I/O	P0_19	P1_3	P1_2	Not connected
FTM0_CH3	I/O	P0_20	P1_4	P0_27	Not connected
FTM0_CH4	I/O	P0_21	P1_5	P0_25	Not connected
FTM0_CH5	I/O	P0_22	P1_6	P0_24	Not connected
FTM0_FAULT0	I	P0_10	P1_7	P0_28	Not connected
FTM0_FAULT1	I	P0_11	P1_12	P1_3	Not connected
FTM0_FAULT2	I	P0_13	P1_13	-	Not connected
FTM0_FAULT3	I	P0_23	P1_14	-	Not connected
FTM1_EXTCLK	I	P0_25	P0_29	-	Not connected
FTM1_CH0	I/O	P0_15	P1_8	-	Not connected
FTM1_CH1	I/O	P0_16	P1_9	-	Not connected
FTM1_CH2	I/O	P0_26	P0_31	-	Not connected
FTM1_CH3	I/O	P0_27	P1_0	-	Not connected
FTM1_QD_PHA	I	P0_24	P0_29	-	Not connected
FTM1_QD_PHB	I	P0_25	P0_30	-	Not connected

11.4.2 Switch matrix register interface

The switch matrix consists of two blocks of pin-assignment registers PINASSIGN and PINENABLE. Every function has an assigned field (1-bit, 2-bit, or 8-bit wide) within this bank of registers where you can program the external pin - identified by its GPIO function - you want the function to connect to.

GPIO0 range from PIO0_0 to PIO0_31 and GPIO1 range from PIO1_0 to PIO1_21, for assignment through the pin-assignment registers, are numbered 0 to 53.

The following functions must be assigned to port pins in different ways:

1. **Movable functions** (PINASSIGN0 to 14):

   All movable functions are digital functions. Assign movable functions to pin numbers through the 8 bits of the PINASSIGN register associated with this function. Once the function is assigned a pin PIO0_n or PIO1_(n-32), it is connected through this pin to a physical pin on the package.

   Remark: You can assign only one digital output function to an external pin at any given time.

   Remark: You can assign more than one digital input function to one external pin.

2. **Fixed-pin functions** (PINENABLE0 to 1):

   Some functions require pins with special characteristics and cannot be moved to other physical pins. Hence these functions are mapped to a fixed port pin. Examples of fixed-pin functions are the oscillator pins or comparator inputs.

   Each fixed-pin function is associated with one bit in the PINENABLE0 or PINENABLE1 register which selects or deselects the function.

– If a fixed-pin function is deselected, any movable function can be assigned to its port and pin.
– If a fixed-pin function is deselected and no movable function is assigned to this pin, the pin is assigned its GPIO function.
– On reset, all fixed-pin functions are deselected.
– If a fixed-pin analog function is selected, its assigned pin cannot be used for any other function.

11.5 Register description

Table 127. Register overview: Switch matrix (base address 0x4000 C000)

Name	Access	Offset	Description	Reset value	Reference
PINASSIGN0	R/W	0x000	Pin assign register 0. Assign movable functions U0_TXD, U0_RXD, U0_RTS, U0_CTS.	0xFFFF FFFF	Table 128
PINASSIGN1	R/W	0x004	Pin assign register 1. Assign movable functions U0_SCLK, U1_TXD, U1_RXD, U1_RTS.	0xFFFF FFFF	Table 129
PINASSIGN2	R/W	0x008	Pin assign register 2. Assign movable functions U1_CTS, U1_SCLK, U2_TXD, U2_RXD.	0xFFFF FFFF	Table 130
PINASSIGN3	R/W	0x00C	Pin assign register 3. Assign movable functions U2_RTS, U2_CTS, U2_SCLK, SPI0_SCK.	0xFFFF FFFF	Table 131
PINASSIGN4	R/W	0x010	Pin assign register 4. Assign movable functions SPI0_MOSI, SPI0_MISO, SPI0_SSEL0, SPI0_SSEL1.	0xFFFF FFFF	Table 132
PINASSIGN5	R/W	0x014	Pin assign register 5. Assign movable functions SPI0_SSEL2, SPI0_SSEL3, SPI1_SCK, SPI1_MOSI	0xFFFF FFFF	Table 133
PINASSIGN6	R/W	0x018	Pin assign register 6. Assign movable functions SPI1_MISO, SPI1_SSEL0, SPI1_SSEL1.	0xFFFF FFFF	Table 134
PINASSIGN7	R/W	0x01C	Pin assign register 7. Assign movable functions I2C0_SCL, I3C0_SDA, I3C0_SCL, I3C0_PUR.	0xFFFF FFFF	Table 135
PINASSIGN8	R/W	0x020	Pin assign register 8. Assign movable functions, ACMP0_OUT,CLKOUT, GPIOINT_BMATCH.	0x00FF FFFF	Table 136

Table 127. Register overview: Switch matrix (base address 0x4000 C000) *…continued*

Name	Access	Offset	Description	Reset value	Reference
FTM_PINASSIGN0	R/W	0x180	FlexTimer Pin assign register 0 Assign movable functions FTM0_EXTCLK...FTM1_CH3	0xFFFFFFFF	Table 137
FTM_PINASSIGN1	R/W	0x184	FlexTimer Pin assign register 1 Assign movable functions FTM1_QD_PHA and FTM1_QD_PHB	0x000000F	Table 138
PINENABLE0	R/W	0x1C0	Pin enable register 0. Enables fixed-pin functions ACMP_In, SWCLK, SWDIO, RESET, CLKIN, VDDCMP, I2C0_SDA, I2C0_SCL, and ADC_n.	0x00FFFF1F	Table 139

11.5.1 Pin assign register 0

Table 128. Pin assign register 0 (PINASSIGN0, address 0x4000 C000) bit description

Bit	Symbol	Description	Reset value
7:0	U0_TXD_O	U0_TXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	U0_RXD_I	U0_RXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	U0_RTS_O	U0_RTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	U0_CTS_I	U0_CTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.2 Pin assign register 1

Table 129. Pin assign register 1 (PINASSIGN1, address 0x4000 C004) bit description

Bit	Symbol	Description	Reset value
7:0	U0_SCLK_IO	U0_SCLK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	U1_TXD_O	U1_TXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	U1_RXD_I	U1_RXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	U1_RTS_O	U1_RTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.3 Pin assign register 2

Table 130. Pin assign register 2 (PINASSIGN2, address 0x4000 C008) bit description

Bit	Symbol	Description	Reset value
7:0	U1_CTS_I	U1_CTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	U1_SCLK_IO	U1_SCLK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	U2_TXD_O	U2_TXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	U2_RXD_I	U2_RXD function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.4 Pin assign register 3

Table 131. Pin assign register 3 (PINASSIGN3, address 0x4000 C00C) bit description

Bit	Symbol	Description	Reset value
7:0	U2_RTS_O	U2_RTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	U2_CTS_I	U2_CTS function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	U2_SCLK_IO	U2_SCLK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	SPI0_SCK_IO	SPI0_SCK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.5 Pin assign register 4

Table 132. Pin assign register 4 (PINASSIGN4, address 0x4000 C010) bit description

Bit	Symbol	Description	Reset value
7:0	SPI0_MOSI_IO	SPI0_MOSI function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	SPI0_MISO_IO	SPI0_MISIO function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	SPI0_SSEL0_IO	SPI0_SSEL0 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	SPI0_SSEL1_IO	SPI0_SSEL1 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.6 Pin assign register 5

Table 133. Pin assign register 5 (PINASSIGN5, address 0x4000 C014) bit description

Bit	Symbol	Description	Reset value
7:0	SPI0_SSEL2_IO	SPI0_SSEL2 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	SPI0_SSEL3_IO	SPI0_SSEL3 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	SPI1_SCK_IO	SPI1_SCK function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	SPI1_MOSI_IO	SPI1_MOSI function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.7 Pin assign register 6

Table 134. Pin assign register 6 (PINASSIGN6, address 0x4000 C018) bit description

Bit	Symbol	Description	Reset value
7:0	SPI1_MISO_IO	SPI1_MISO function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	SPI1_SSEL0_IO	SPI1_SSEL0 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	SPI1_SSEL1_IO	SPI1_SSEL1 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	I2C0_SDA_IO	I2C0_SDA function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.8 Pin assign register 7

Table 135. Pin assign register 7 (PINASSIGN7, address 0x4000 C01C) bit description

Bit	Symbol	Description	Reset value
7:0	I2C0_SCL_IO	I2C0_SCL function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	I3C0_SDA_IO	I3C0_SDA function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	I3C0_SCL_IO	I3C0_SCL function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	I3C0_PUR_IO	I3C0_PUR function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF

11.5.9 Pin assign register 8

Table 136. Pin assign register 8 (PINASSIGN8, address 0x4000 C020) bit description

Bit	Symbol	Description	Reset value
7:0	ACMP0_OUT_O	ACMP0_OUT function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
15:8	CLKOUT _O	CLKOUT function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
23:16	GPIO_INT_BMAT_O	GPIO_INT_BMAT function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_31 (= 0x1F) and from PIO1_0 (= 0x20) to PIO1_21(= 0x35).	0xFF
31:24	-	Reserved	-

11.5.10 FlexTimer Pin assign register 0

Table 137. FlexTimer Pin assign register 0 (FTM_PINASSIGN0, address 0x4000 C180) bit description

Bit	Symbol	Description	Reset value
1:0	FTM0_EXTCLK	Assign movable function FTM0_EXTCLK 00 = Selection 0 function pins 01 = Selection 1 function pins 10 = Selection 2 function pins 11 = Selection 3 function pins See Table 126 “Flextimer pin assignments”	0xFF
3:2	FTM0_CH0	Assign movable function FTM0_CH0	0xFF
5:4	FTM0_CH1	Assign movable function FTM0_CH1	0xFF
7:6	FTM0_CH2	Assign movable function FTM0_CH2	0xFF
9:8	FTM0_CH3	Assign movable function FTM0_CH3	0xFF
11:10	FTM0_CH4	Assign movable function FTM0_CH4	0xFF
13:12	FTM0_CH5	Assign movable function FTM0_CH5	0xFF
15:14	FTM0_FAULT0	Assign movable function FTM0_FAULT0	0xFF
17:16	FTM0_FAULT1	Assign movable function FTM0_FAULT1	0xFF
19:18	FTM0_FAULT2	Assign movable function FTM0_FAULT2	0xFF
21:20	FTM0_FAULT3	Assign movable function FTM0_FAULT3	0xFF
23:22	FTM1_EXTCLK	Assign movable function FTM1_EXTCLK	0xFF
25:24	FTM1_CH0	Assign movable function FTM1_CH0	0xFF
27:26	FTM1_CH1	Assign movable function FTM1_CH1	0xFF
29:28	FTM1_CH2	Assign movable function FTM1_CH2	0xFF
31:30	FTM1_CH3	Assign movable function FTM1_CH3	0xFF

11.5.11 FlexTimer Pin assign register 1

Table 138. FlexTimer Pin assign register 0 (FTM_PINASSIGN1, address 0x4000 C184) bit description

Bit	Symbol	Description	Reset value
1:0	FTM1_QD_PHA	Assign movable function FTM1_QD_PHA	0xFF
3:2	FTM1_QD_PHB	Assign movable function FTM1_QD_PHB	0xFF
31:4	-	Reserved	0

11.5.12 PINENABLE 0

Table 139. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description

Bit	Symbol	Value	Description	Reset value
0	ACMP_I1		ACMP_I1 function select.	1
		0	ACMP_I1 enabled on pin PIO0_00.	
		1	ACMP_I1 disabled.	

Table 139. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description

Bit	Symbol	Value	Description	Reset value
1	ACMP_I2		ACMP_I2 function select.	1
		0	ACMP_I2 enabled on pin PIO0_1.	
		1	ACMP_I2 disabled.	
2	ACMP_I3		ACMP_I3 function select.	1
		0	ACMP_I3 enabled on pin PIO0_14.	
		1	ACMP_I3 disabled.	
3	ACMP_I4		ACMP_I4 function select.	1
		0	ACMP_I4 enabled on pin PIO0_23.	
		1	ACMP_I4 disabled.	
4	ACMP_I5		ACMP_I5 function select.	1
		0	ACMP_I5 enabled on pin PIO0_30.	
		1	ACMP_I5 disabled.	
5	SWCLK		SWCLK function select.	0
		0	SWCLK enabled on pin PIO0_3.	
		1	SWCLK disabled.	
6	SWDIO		SWDIO function select.	0
		0	SWDIO enabled on pin PIO0_2.	
		1	SWDIO disabled.	
7	RESETN		RESETN function select.	0
		0	RESETN enabled on pin PIO0_5.	
		1	RESETN disabled.	
8	CLKIN		CLKIN function select.	1
		0	CLKIN enabled on pin PIO0_1.	
		1	CLKIN disabled.	
9	CMPVREF		CMPVREF function select.	1
		0	CMPVREF enabled on pin PIO0_6.	
		1	CMPVREF disabled.	
10	XTALIN		XTALIN function select.	1
		0	XTALIN enabled on pin PIO0_8.	
		1	XTALIN disabled.	
11	XTALOUT		XTALOUT function select.	1
		0	XTALOUT enabled on pin PIO0_9.	
		1	XTALOUT disabled.	
12	ADC_0		ADC_0 function select.	1
		0	ADC_0 enabled on pin PIO0_7.	
		1	ADC_0 disabled.	
13	ADC_1		ADC_1 function select.	1
		0	ADC_1 enabled on pin PIO0_6.	
		1	ADC_1 disabled.	

Table 139. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description

Bit	Symbol	Value	Description	Reset value
14	ADC_2		ADC_2 function select.	1
		0	ADC_2 enabled on pin PIO0_14.	
		1	ADC_2 disabled.	
15	ADC_3		ADC_3 function select.	1
		0	ADC_3 enabled on pin PIO0_23.	
		1	ADC_3 disabled.	
16	ADC_4		ADC_4 function select.	1
		0	ADC_4 enabled on pin PIO0_22.	
		1	ADC_4 disabled.	
17	ADC_5		ADC_5 function select.	1
		0	ADC_5 enabled on pin PIO0_21.	
		1	ADC_5 disabled.	
18	ADC_6		ADC_6 function select.	1
		0	ADC_6 enabled on pin PIO0_20.	
		1	ADC_6 disabled.	
19	ADC_7		ADC_7 function select.	1
		0	ADC_7 enabled on pin PIO0_19.	
		1	ADC_7 disabled.	
20	ADC_8		ADC_8 function select.	1
		0	ADC_8 enabled on pin PIO0_18.	
		1	ADC_8 disabled.	
21	ADC_9		ADC_9 function select.	1
		0	ADC_9 enabled on pin PIO0_17.	
		1	ADC_9 disabled.	
22	ADC_10		ADC_10 function select.	1
		0	ADC_10 enabled on pin PIO0_13.	
		1	ADC_10 disabled.	
23	ADC_11		ADC_11 function select.	1
		0	ADC_11 enabled on pin PIO0_4.	
		1	ADC_11 disabled.	
31:24	-		Reserved	

Remark: In analog mode, the internal pull-up must be disabled via the IOCON register

UM11607

Chapter 12: I/O Configuration (IOCON)

Rev. 3 — April 2023 **User manual**

12.1 How to read this chapter

The IOCON block is identical for all LPC86x parts. Registers for pins that are not available on a specific package are reserved.

Table 140. Pinout summary

Package	Pins/configuration registers available
HVQFN48	PIO0_0 to PIO0_31; PIO1_0 to PIO1_9
LQFP64	PIO0_0 to PIO0_31; PIO1_0 to PIO1_21
HVQFN32	PIO0_0 to PIO0_28

12.2 Features

The following electrical properties are configurable for each pin:

- Pull-up/pull-down resistor
- Open-drain mode
- Hysteresis
- Digital glitch filter with programmable time constant
- Analog mode (for a subset of pins, see the LPC86x data sheet)

One I2C supports Fast-mode Plus with 1 Mbit/s data rates on two true open-drain pins and listen mode. Three I2Cs support data rates up to 400 kbit/s on standard digital pins.

12.3 Basic configuration

Enable the clock to the IOCON in the SYSAHBCLKCTRL register (Table 94, bit 18). Once the pins are configured, you can disable the IOCON clock to conserve power.

Remark: If the open-drain pins PIO0_10 and PIO0_11 are not available on the package, prevent the pins from internally floating as follows: Set bits 10 and 11 in the GPIO DIR0 register to 1 to enable the output driver and write 1 to bits 10 and 11 in the GPIO CLR0 register to drive the outputs LOW internally.

12.4 General description

12.4.1 Pin configuration

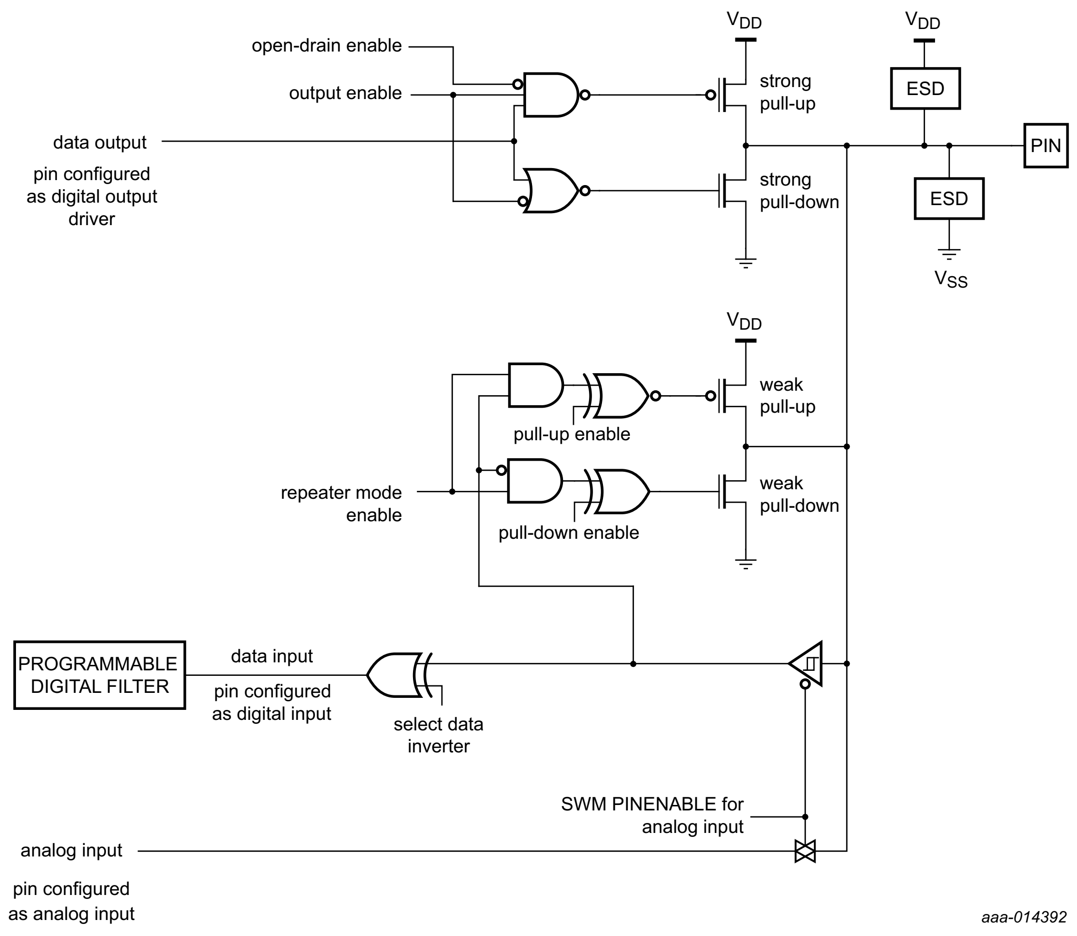


Fig 13. Pin configuration

12.4.2 Pin function

The pin function is determined entirely through the switch matrix. By default one of the GPIO functions is assigned to each pin. The switch matrix can assign all functions from the movable function table to any pin in the IOCON block or enable a special function like an analog input on a specific pin.

Related links:

Table 125 “Movable functions (assign to pins PIO0_0 to PIO0_31 and PIO1_0 to PIO1_21 through switch matrix)”

12.4.3 Pin mode

The MODE bit in the IOCON register allows enabling or disabling an on-chip pull-up resistor for each pin.

The repeater mode enables the pull-up resistor if the pin is high and enables the pull-down resistor if the pin is low. This causes the pin to retain its last known state if it is configured as an input and is not driven externally. Repeater mode may typically be used to prevent a pin from floating (and potentially using significant power if it floats to an indeterminate state) if it is temporarily not driven.

12.4.4 Open-drain mode

An open-drain mode can be enabled for all digital I/O pins that are not the I2C-bus pins. This mode is not a true open-drain mode. The input cannot be pulled up above V_{DD}.

Remark: As opposed to the true open-drain I2C-bus pins, digital pins with configurable open-drain mode are **not** 5 V tolerant when V_{DD} = 0.

12.4.5 Analog mode

The switch matrix automatically configures the pin in analog mode whenever an analog input or output is selected as the pin’s function. In analog mode, the internal pull-up should be disabled via IOCON register.

12.4.6 I^{2}C-bus mode

The I^{2}C-bus pins PIO0_10 and PIO0_11 can be programmed to support a true open-drain mode independently of whether the I2C function is selected or another digital function. If the I^{2}C function is selected, all three I^{2}C modes, Standard mode, Fast-mode, and Fast-mode plus, are supported. A digital glitch filter can be configured for all functions. Pins PIO0_10 and PIO0_11 operate as high-current sink drivers (20 mA) independently of the programmed function.

Remark: Pins PIO0_10 and PIO0_11 are 5 V tolerant when V_{DD} = 0 and when V_{DD} is at operating voltage level.

12.4.7 Programmable digital filter

All GPIO pins are equipped with a programmable, digital glitch filter. The filter rejects input pulses with a selectable duration of shorter than one, two, or three cycles of a filter clock (S_MODE = 1, 2, or 3). For each individual pin, the filter clock can be selected from one of seven peripheral clocks PCLK0 to 6, which are derived from the main clock using the IOCONCLKDIV0 to 6 registers. The filter can also be bypassed entirely.

Any input pulses of duration T_{pulse} of either polarity will be rejected if:
$T_{pulse} < T_{PCLKn} \times S\_MODE$

Input pulses of one filter clock cycle longer may also be rejected:

$T_{pulse} = T_{PCLKn}$ ´ (S_MODE + 1)

Remark: The filtering effect is accomplished by requiring that the input signal be stable for (S_MODE +1) successive edges of the filter clock before being passed on to the chip. Enabling the filter results in delaying the signal to the internal logic and should be done only if specifically required by an application. For high-speed or time critical functions ensure that the filter is bypassed.

If the delay of the input signal must be minimized, select a faster PCLK and a higher sample mode (S_MODE) to minimize the effect of the potential extra clock cycle.

If the sensitivity to noise spikes must be minimized, select a slower PCLK and lower sample mode.

Related registers and links:

Table 108 “IOCON glitch filter clock divider registers 6 to 0 (IOCONCLKDIV[6:0], address 0x4004 8134 (IOCONCLKDIV6) to 0x004 814C (IOCONFILTCLKDIV0)) bit description”

12.5 Register description

Each port pin PIO0_m and PIO1_m have one IOCON register assigned to control the function and electrical characteristics of the pin.

Remark: See Table 142 for the IOCON register map ordered by pin name.

Table 141. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reference
PIO0_17	R/W	0x000	I/O configuration for pin PIO0_17/ADC_9	Table 143
PIO0_13	R/W	0x004	I/O configuration for pin PIO0_13/ADC_10	Table 144
PIO0_12	R/W	0x008	I/O configuration for pin PIO0_12	Table 145
PIO0_5	R/W	0x00C	I/O configuration for pin PIO0_5/$\overline{\text{RESET}}$	Table 146
PIO0_4	R/W	0x010	I/O configuration for pin PIO0_4/ADC_11/TRSTN/WAKEUP	Table 147
PIO0_3	R/W	0x014	I/O configuration for pin PIO0_3/SWCLK	Table 148
PIO0_2	R/W	0x018	I/O configuration for pin PIO0_2/SWDIO	Table 149
PIO0_11	R/W	0x01C	I/O configuration for pin PIO0_11/I2C0_SDA. This is the pin configuration for the true open-drain pin.	Table 150
PIO0_10	R/W	0x020	I/O configuration for pin PIO0_10/I2C0_SCL. This is the pin configuration for the true open-drain pin.	Table 151
PIO0_16	R/W	0x024	I/O configuration for pin PIO0_16	Table 152
PIO0_15	R/W	0x028	I/O configuration for pin PIO0_15	Table 153
PIO0_1	R/W	0x02C	I/O configuration for pin PIO0_1/ACMP_I2/CLKIN	Table 154
-	-	0x030	Reserved	-
PIO0_9	R/W	0x034	I/O configuration for pin PIO0_9/XTALOUT	Table 155
PIO0_8	R/W	0x038	I/O configuration for pin PIO0_8/XTALIN	Table 156
PIO0_7	R/W	0x03C	I/O configuration for pin PIO0_7/ADC_0	Table 157
PIO0_6	R/W	0x040	I/O configuration for pin PIO0_6/ADC_1/ CMPVREF	Table 158
PIO0_0	R/W	0x044	I/O configuration for pin PIO0_0/ACMP_I1	Table 159
PIO0_14	R/W	0x048	I/O configuration for pin PIO0_14/ ACMP_I3/ADC_2	Table 160
-	-	0x04C	Reserved.	-
PIO0_28	R/W	0x050	I/O configuration for pin PIO0_28/WKTCLKIN	Table 161
PIO0_27	R/W	0x054	I/O configuration for pin PIO0_27	Table 162
PIO0_26	R/W	0x058	I/O configuration for pin PIO0_26	Table 163

Table 141. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reference
PIO0_25	R/W	0x05C	I/O configuration for pin PIO0_25	Table 164
PIO0_24	R/W	0x060	I/O configuration for pin PIO0_24	Table 165
PIO0_23	R/W	0x064	I/O configuration for pin PIO0_23/ADC_3/ACMP_I4	Table 166
PIO0_22	R/W	0x068	I/O configuration for pin PIO0_22/ADC_4	Table 167
PIO0_21	R/W	0x06C	I/O configuration for pin PIO0_21/ADC_5	Table 168
PIO0_20	R/W	0x070	I/O configuration for pin PIO0_20/ADC_6	Table 169
PIO0_19	R/W	0x074	I/O configuration for pin PIO0_19/ADC_7	Table 170
PIO0_18	R/W	0x078	I/O configuration for pin PIO0_18/ADC_8	Table 171
PIO1_8	R/W	0x07C	I/O configuration for pin PIO1_8	Table 172
PIO1_9	R/W	0x080	I/O configuration for pin PIO1_9	Table 173
PIO1_12	R/W	0x084	I/O configuration for pin PIO1_12	Table 174
PIO1_13	R/W	0x088	I/O configuration for pin PIO1_13	Table 175
PIO0_31	R/W	0x08C	I/O configuration for pin PIO0_31	Table 176
PIO1_0	R/W	0x090	I/O configuration for pin PIO1_0	Table 177
PIO1_1	R/W	0x094	I/O configuration for pin PIO1_1	Table 178
PIO1_2	R/W	0x098	I/O configuration for pin PIO1_2	Table 179
PIO1_14	R/W	0x09C	I/O configuration for pin PIO1_14	Table 180
PIO1_15	R/W	0x0A0	I/O configuration for pin PIO1_15	Table 181
PIO1_3	R/W	0x0A4	I/O configuration for pin PIO1_3	Table 182
PIO1_4	R/W	0x0A8	I/O configuration for pin PIO1_4	Table 183
PIO1_5	R/W	0x0AC	I/O configuration for pin PIO1_5	Table 184
PIO1_16	R/W	0x0B0	I/O configuration for pin PIO1_16	Table 185
PIO1_17	R/W	0x0B4	I/O configuration for pin PIO1_17	Table 186
PIO1_6	R/W	0x0B8	I/O configuration for pin PIO1_6	Table 187
PIO1_18	R/W	0x0BC	I/O configuration for pin PIO1_18	Table 188
PIO1_19	R/W	0x0C0	I/O configuration for pin PIO1_19	Table 189
PIO1_7	R/W	0x0C4	I/O configuration for pin PIO1_7	Table 190
PIO0_29	R/W	0x0C8	I/O configuration for pin PIO0_29	Table 191
PIO0_30	R/W	0x0CC	I/O configuration for pin PIO0_30/ACMP_I5	Table 192
PIO1_20	R/W	0x0D0	I/O configuration for pin PIO1_20	Table 193
PIO1_21	R/W	0x0D4	I/O configuration for pin PIO1_21	Table 194
PIO1_11	R/W	0x0D8	I/O configuration for pin PIO1_11	Table 195
PIO1_10	R/W	0x0DC	I/O configuration for pin PIO1_10	Table 196

Table 142. I/O configuration registers ordered by pin name

Name	Address offset	True open-drain	Analog[1]	Digital filter	High-drive output	Reference
PIO0_0	0x044	no	yes	yes	no	Table 159
PIO0_1	0x02C	no	yes	yes	no	Table 154
PIO0_2	0x018	no	no	yes	yes	Table 149
PIO0_3	0x014	no	no	yes	yes	Table 148

Table 142. I/O configuration registers ordered by pin name

Name	Address offset	True open-drain	Analog[1]	Digital filter	High-drive output	Reference
PIO0_4	0x010	no	yes	yes	no	Table 147
PIO0_5	0x00C	no	no	yes	no	Table 146
PIO0_6	0x040	no	yes	yes	no	Table 158
PIO0_7	0x03C	no	yes	yes	no	Table 157
PIO0_8	0x038	no	yes	yes	no	Table 156
PIO0_9	0x034	no	yes	yes	no	Table 155
PIO0_10	0x020	yes	no	yes	no	Table 151
PIO0_11	0x01C	yes	no	yes	no	Table 150
PIO0_12	0x008	no	no	yes	yes	Table 145
PIO0_13	0x004	no	yes	yes	no	Table 144
PIO0_14	0x048	no	yes	yes	no	Table 160
PIO0_15	0x028	no	no	yes	no	Table 153
PIO0_16	0x024	no	no	yes	yes	Table 152
PIO0_17	0x000	no	yes	yes	no	Table 143
PIO0_18	0x078	no	yes	yes	no	Table 171
PIO0_19	0x074	no	yes	yes	no	Table 170
PIO0_20	0x070	no	yes	yes	no	Table 169
PIO0_21	0x06C	no	yes	yes	no	Table 168
PIO0_22	0x068	no	yes	yes	no	Table 167
PIO0_23	0x064	no	yes	yes	no	Table 166
PIO0_24	0x060	no	no	yes	no	Table 165
PIO0_25	0x05C	no	no	yes	no	Table 164
PIO0_26	0x058	no	no	yes	no	Table 163
PIO0_27	0x054	no	no	yes	no	Table 162
PIO0_28	0x050	no	no	yes	no	Table 161
PIO0_29	0x0C8	no	yes	yes	no	Table 191
PIO0_30	0x0CC	no	yes	yes	no	Table 192
PIO0_31	0x08C	no	no	yes	no	Table 176
PIO1_0	0x090	no	no	yes	no	Table 177
PIO1_1	0x094	no	no	yes	no	Table 178
PIO1_2	0x098	no	no	yes	no	Table 179
PIO1_3	0x0A4	no	no	yes	no	Table 182
PIO1_4	0x0A8	no	no	yes	no	Table 183
PIO1_5	0x0AC	no	no	yes	no	Table 184
PIO1_6	0x0B8	no	no	yes	no	Table 187
PIO1_7	0x0C4	no	no	yes	no	Table 190
PIO1_8	0x07C	no	no	yes	no	Table 172
PIO1_9	0x080	no	no	yes	no	Table 173
PIO1_10	0x0DC	no	no	yes	no	Table 196
PIO1_11	0x0D8	no	no	yes	no	Table 195

Table 142. I/O configuration registers ordered by pin name

Name	Address offset	True open-drain	Analog[1]	Digital filter	High-drive output	Reference
PIO1_12	0x084	no	no	yes	no	Table 174
PIO1_13	0x088	no	no	yes	no	Table 175
PIO1_14	0x09C	no	no	yes	no	Table 180
PIO1_15	0x0A0	no	no	yes	no	Table 181
PIO1_16	0x0B0	no	no	yes	no	Table 185
PIO1_17	0x0B4	no	no	yes	no	Table 186
PIO1_18	0x0BC	no	no	yes	no	Table 188
PIO1_19	0x0C0	no	no	yes	no	Table 189
PIO1_20	0x0D0	no	no	yes	no	Table 193
PIO1_21	0x0D4	no	no	yes	no	Table 194

[1] The analog pad is enabled when the analog function is selected in the switch matrix through the PINENABLE register.

12.5.1 PIO0_17 register

Table 143. PIO0_17 register (PIO0_17, address 0x4004 4000) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 143. PIO0_17 register (PIO0_17, address 0x4004 4000) bit description

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.2 PIO0_13 register

Table 144. PIO0_13 register (PIO0_13, address 0x4004 4004) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 144. PIO0_13 register (PIO0_13, address 0x4004 4004) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.3 PIO0_12 register

Table 145. PIO0_12 register (PIO0_12, address 0x4004 4008) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 145. PIO0_12 register (PIO0_12, address 0x4004 4008) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.4 PIO0_5 register

Table 146. PIO0_5 register (PIO0_5, address 0x4004 400C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 146. PIO0_5 register (PIO0_5, address 0x4004 400C) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.5 PIO0_4 register

Table 147. PIO0_4 register (PIO0_4, address 0x4004 4010) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 147. PIO0_4 register (PIO0_4, address 0x4004 4010) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.6 PIO0_3 register

Table 148. PIO0_3 register (PIO0_3, address 0x4004 4014) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input.	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 148. PIO0_3 register (PIO0_3, address 0x4004 4014) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.7 PIO0_2 register

Table 149. PIO0_2 register (PIO0_2, address 0x4004 4018) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input.	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	

Table 149. PIO0_2 register (PIO0_2, address 0x4004 4018) bit description

Bit	Symbol	Value	Description	Reset value
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.8 PIO0_11 register

Table 150. PIO0_11 register (PIO0_11, address 0x4004 401C) bit description

Bit	Symbol	Value	Description	Reset value
5:0	-		Reserved.	0
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
7	-		Reserved.	1
9:8	I2CMODE		Selects I2C mode. Select Standard mode (I2CMODE = 00, default) or Standard I/O functionality (I2CMODE = 01) if the pin function is GPIO.	00
		0x0	Standard mode/ Fast-mode I2C.	
		0x1	Standard GPIO functionality. Requires external pull-up for GPIO output function.	
		0x2	Fast-mode Plus I2C	
		0x3	Reserved.	
10	-	-	Reserved.	-
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	

Table 150. PIO0_11 register (PIO0_11, address 0x4004 401C) bit description

Bit	Symbol	Value	Description	Reset value
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	-

12.5.9 PIO0_10 register

Table 151. PIO0_10 register (PIO0_10, address 0x4004 4020) bit description

Bit	Symbol	Value	Description	Reset value
5:0	-		Reserved.	0
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
7	-		Reserved.	1
9:8	I2CMODE		Selects I2C mode. Select Standard mode (I2CMODE = 00, default) or Standard I/O functionality (I2CMODE = 01) if the pin function is GPIO.	00
		0x0	Standard mode/ Fast-mode I2C.	
		0x1	Standard GPIO functionality. Requires external pull-up for GPIO output function.	
		0x2	Fast-mode Plus I2C	
		0x3	Reserved.	
10	-	-	Reserved.	-
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	

Table 151. PIO0_10 register (PIO0_10, address 0x4004 4020) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	-

12.5.10 PIO0_16 register

Table 152. PIO0_16 register (PIO0_16, address 0x4004 4024) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	

Table 152. PIO0_16 register (PIO0_16, address 0x4004 4024) bit description

Bit	Symbol	Value	Description	Reset value
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.11 PIO0_15 register

Table 153. PIO0_15 register (PIO0_15, address 0x4004 4028) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	

Table 153. PIO0_15 register (PIO0_15, address 0x4004 4028) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.12 PIO0_1 register

Table 154. PIO0_1 register (PIO0_1, address 0x4004 402C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	

Table 154. PIO0_1 register (PIO0_1, address 0x4004 402C) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.13 PIO0_9 register

Table 155. PIO0_9 register (PIO0_9, address 0x4004 4034) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	

Table 155. PIO0_9 register (PIO0_9, address 0x4004 4034) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.14 PIO0_8 register

Table 156. PIO0_8 register (PIO0_8, address 0x4004 4038) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.15 PIO0_7 register

Table 157. PIO0_7 register (PIO0_7, address 0x4004 403C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.16 PIO0_6 register

Table 158. PIO0_6 register (PIO0_6, address 0x4004 4040) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.17 PIO0_0 register

Table 159. PIO0_0 register (PIO0_0, address 0x4004 4044) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.18 PIO0_14 register

Table 160. PIO0_14 register (PIO0_14, address 0x4004 4048) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.19 PIO0_28 register

Table 161. PIO0_28 register (PIO0_28, address 0x4004 4050) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.20 PIO0_27 register

Table 162. PIO0_27 register (PIO0_27, address 0x4004 4054) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.21 PIO0_26 register

Table 163. PIO0_26 register (PIO0_26, address 0x4004 4058) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.22 PIO0_25 register

Table 164. PIO0_25 register (PIO0_25, address 0x4004 405C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.23 PIO0_24 register

Table 165. PIO0_24 register (PIO0_24, address 0x4004 4060) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.24 PIO0_23 register

Table 166. PIO0_23 register (PIO0_23, address 0x4004 4064) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.25 PIO0_22 register

Table 167. PIO0_22 register (PIO0_22, address 0x4004 4068) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.26 PIO0_21 register

Table 168. PIO0_21 register (PIO0_21, address 0x4004 406C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.27 PIO0_20 register

Table 169. PIO0_20 register (PIO0_20, address 0x4004 4070) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.28 PIO0_19 register

Table 170. PIO0_19 register (PIO0_19, address 0x4004 4074) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.29 PIO0_18 register

Table 171. PIO0_18 register (PIO0_18, address 0x4004 4078) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.30 PIO1_8 register

Table 172. PIO1_8 register (PIO1_8, address 0x4004 407C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.31 PIO1_9 register

Table 173. PIO1_9 register (PIO1_9, address 0x4004 4080) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.32 PIO1_12 register

Table 174. PIO1_12 register (PIO1_12, address 0x4004 4084) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.33 PIO1_13 register

Table 175. PIO1_13 register (PIO1_13, address 0x4004 4088) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.34 PIO0_31 register

Table 176. PIO0_31 register (PIO0_31, address 0x4004 408C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.35 PIO1_0 register

Table 177. PIO1_0 register (PIO1_0, address 0x4004 4090) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.36 PIO1_1 register

Table 178. PIO1_1 register (PIO1_1, address 0x4004 4094) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.37 PIO1_2 register

Table 179. PIO1_2 register (PIO1_2, address 0x4004 4098) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.38 PIO1_14 register

Table 180. PIO1_14 register (PIO1_14, address 0x4004 409C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.39 PIO1_15 register

Table 181. PIO1_15 register (PIO1_15, address 0x4004 40A0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.40 PIO1_3 register

Table 182. PIO1_3 register (PIO1_3, address 0x4004 40A4) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.41 PIO1_4 register

Table 183. PIO1_4 register (PIO1_4, address 0x4004 40A8) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.42 PIO1_5 register

Table 184. PIO1_5 register (PIO1_5, address 0x4004 40AC) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.43 PIO1_16 register

Table 185. PIO1_16 register (PIO1_16, address 0x4004 40B0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.44 PIO1_17 register

Table 186. PIO1_17 register (PIO1_17, address 0x4004 40B4) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.45 PIO1_6 register

Table 187. PIO1_6 register (PIO1_6, address 0x4004 40B8) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.46 PIO1_18 register

Table 188. PIO1_18 register (PIO1_18, address 0x4004 40BC) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.47 PIO1_19 register

Table 189. PIO1_19 register (PIO1_19, address 0x4004 40C0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.48 PIO1_7 register

Table 190. PIO1_7 register (PIO1_7, address 0x4004 40C4) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.49 PIO0_29 register

Table 191. PIO0_29 register (PIO0_29, address 0x4004 40C8) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.50 PIO1_30 register

Table 192. PIO0_30 register (PIO0_30, address 0x4004 40CC) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.51 PIO1_20 register

Table 193. PIO1_20 register (PIO1_20, address 0x4004 40D0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.52 PIO1_21 register

Table 194. PIO1_21 register (PIO1_21, address 0x4004 40D4) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.53 PIO1_11 register

Table 195. PIO1_11 register (PIO1_11, address 0x4004 40D8) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

12.5.54 PIO1_10 register

Table 196. PIO1_10 register (PIO1_10, address 0x4004 40DC) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	
5	HYS		Hysteresis.	1
		0	Disable.	
		1	Enable.	
6	INV		Invert input	0
		0	Input not inverted (HIGH on pin reads as 1; LOW on pin reads as 0).	
		1	Input inverted (HIGH on pin reads as 0, LOW on pin reads as 1).	
9:7	-	-	Reserved.	0b001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. **Remark:** This is not a true open-drain mode.	
12:11	S_MODE		Digital filter sample mode.	0
		0x0	Bypass input filter.	
		0x1	1 clock cycle. Input pulses shorter than one filter clock are rejected.	
		0x2	2 clock cycles. Input pulses shorter than two filter clocks are rejected.	
		0x3	3 clock cycles. Input pulses shorter than three filter clocks are rejected.	
15:13	CLK_DIV		Select peripheral clock divider for input filter sampling clock. Value 0x7 is reserved.	0
		0x0	IOCONCLKDIV0.	
		0x1	IOCONCLKDIV1.	
		0x2	IOCONCLKDIV2.	
		0x3	IOCONCLKDIV3.	
		0x4	IOCONCLKDIV4.	
		0x5	IOCONCLKDIV5.	
		0x6	IOCONCLKDIV6.	
31:16	-	-	Reserved.	0

UM11607

Chapter 13: General Purpose I/O (GPIO)

Rev. 3 — April 2023 **User manual**

13.1 How to read this chapter

GPIO pins are placed in logical groups by port. A port may contain up to 32 pins. Typically, these pins have other functions. See Chapter 12 “I/O Configuration (IOCON)” for more details. Table 197 shows the GPIO pins that are available for each package type.

Table 197. GPIO pins available

Package	GPIO Port 0	GPIO Port 1
HVQFN48	PIO0_0 to PIO0_31	PIO1_0 to PIO1_9
LQFP64	PIO0_0 to PIO0_31	PIO1_0 to PIO1_21
HVQFN32	PIO0_0 to PIO0_28	N/A

13.2 Basic configuration

For the GPIO port registers, enable the clock to the GPIO port in the SYSAHBCLKCTRL register (Table 94).

13.3 Features

- GPIO pins can be configured as input or output by software.
- All GPIO pins default to inputs with interrupt disabled at reset.
- Pin registers allow pins to be sensed and set individually.
- Direction (input/output) can be set and cleared individually.

13.4 General description

The GPIO pins can be used in several ways to set pins as inputs or outputs and use the inputs as combinations of level and edge sensitive interrupts.

The GPIOs can be used as external interrupts together with the pin interrupt block.

The GPIO port registers configure each GPIO pin as input or output and read the state of each pin if the pin is configured as input or set the state of each pin if the pin is configured as output.

13.5 Register description

Note: In all GPIO registers, bits that are not shown are reserved.

GPIO port addresses can be read and written as bytes, halfwords, or words.

Remark: ext in this table and subsequent tables indicates that the data read after reset depends on the state of the pin, which in turn may depend on an external source.

Table 198. Register overview: GPIO port (base address 0xA000 0000)

Name	Access	Address offset	Description	Reset value	Width	Reference
B0 to B31	R/W	0x0000 to 0x001F	Byte pin registers port 0	ext	byte (8 bit)	Table 199
B32 to B53	R/W	0x0020 to 0x0035	Byte pin registers port1; PIO1_0 to PIO1_21	ext	byte (8 bit)	Table 199
-	-	0x0036 to 0x003F	Reserved	-	-	-
W0 to W31	R/W	0x1000 to 0x107C	Word pin registers port 0	ext	word (32 bit)	Table 200
W32 to W53	R/W	0x1080 to 0x10D4	Word pin registers port 1; PIO1_0 to PIO1_21	ext	word (32 bit)	Table 200
DIR0	R/W	0x2000	Direction registers port 0	0	word (32 bit)	Table 201
DIR1	R/W	0x2004	Direction registers port 1	0	word (32 bit)	Table 201
MASK0	R/W	0x2080	Mask register port 0	0	word (32 bit)	Table 202
MASK1	R/W	0x2084	Mask register port 1	0	word (32 bit)	Table 202
PIN0	R/W	0x2100	Port pin register port 0	ext	word (32 bit)	Table 203
PIN1	R/W	0x2104	Port pin register port 1	ext	word (32 bit)	Table 203
MPIN0	R/W	0x2180	Masked port register port 0	ext	word (32 bit)	Table 204
SET0	R/W	0x2200	Write: Set register for port 0 Read: output bits for port 0	0	word (32 bit)	Table 205
SET1	R/W	0x2204	Write: Set register for port 1 Read: output bits for port 1	0	word (32 bit)	Table 205
CLR0	WO	0x2280	Clear port 0	-	word (32 bit)	Table 206
CLR1	WO	0x2284	Clear port 1	-	word (32 bit)	Table 206
NOT0	WO	0x2300	Toggle port 0	-	word (32 bit)	Table 207
NOT1	WO	0x2304	Toggle port 1	-	word (32 bit)	Table 207
DIRSET0	WO	0x2380	Set pin direction bits for port 0.	0	word (32 bit)	Table 208
DIRSET1	WO	0x2384	Set pin direction bits for port 1.	0	word (32 bit)	Table 208
DIRCLR0	WO	0x2400	Clear pin direction bits for port 0.	-	word (32 bit)	Table 209
DIRCLR1	WO	0x2404	Clear pin direction bits for port 1.	-	word (32 bit)	Table 209
DIRNOT0	WO	0x2480	Toggle pin direction bits for port 0.	-	word (32 bit)	Table 210
DIRNOT1	WO	0x2484	Toggle pin direction bits for port 1.	-	word (32 bit)	Table 210

13.5.1 GPIO port byte pin registers

Each GPIO pin has a byte register in this address range. Software typically reads and writes bytes to access individual pins, but can read or write halfwords to sense or set the state of two pins, and read or write words to sense or set the state of four pins.

Table 199. GPIO port byte pin registers (B[0:53], addresses 0xA000 0000 (B0) to 0xA000 0035 (B53)) bit description

Bit	Symbol	Description	Reset value	Access
0	PBYTE	Read: state of the pin PIO0m_n, regardless of direction, masking, or alternate function, except that pins configured as analog I/O always read as 0. One register for each port pin: m = port 0 to 1; n = pin 0 to 31 for port 0, and pin 0 to 21 for port 1. Write: loads the pin's output bit.	ext	R/W
7:1		Reserved (0 on read, ignored on write)	0	-

13.5.2 GPIO port word pin registers

Each GPIO pin has a word register in this address range. Any byte, halfword, or word read in this range will be all zeros if the pin is low or all ones if the pin is high, regardless of direction, masking, or alternate function, except that pins configured as analog I/O always read as zeros. Any write will clear the pin's output bit if the value written is all zeros, else it will set the pin's output bit.

Table 200. GPIO port word pin registers (W[0:53], addresses 0xA000 1000 (W0) to 0xA000 10D4 (W53)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	PWORD	Read 0: pin PIOm_n is LOW. Write 0: clear output bit. Read 0xFFFF FFFF: pin PIOm_n is HIGH. Write any value 0x0000 0001 to 0xFFFF FFFF: set output bit. **Remark:** Only 0 or 0xFFFF FFFF can be read. Writing any value other than 0 will set the output bit. One register for each port pin: m = port 0 to 1; n = pin 0 to 31 for port 0; n = pin 0 to 21 for port 1.	ext	R/W

13.5.3 GPIO port direction registers

Each GPIO port has one direction register for configuring the port pins as inputs or outputs.

Table 201. GPIO direction port register (DIR[0:1], address 0xA000 2000 (DIR0) to 0xA00002004 (DIR1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	DIRP	Selects pin direction for pin PIOm_n (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = input. 1 = output.	0	R/W

13.5.4 GPIO port mask registers

These registers affect writing and reading the MPORT registers. Zeroes in these registers enable reading and writing; ones disable writing and result in zeros in corresponding positions when reading.

Table 202. GPIO mask port register (MASK[0:1], address 0xA000 2080 (MASK0) to 0xA0002084(MASK1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	MASKP	Controls which bits corresponding to PIOm_n are active in the MPORT register (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = Read MPORT: pin state; write MPORT: load output bit. 1 = Read MPORT: 0; write MPORT: output bit not affected.	0	R/W

13.5.5 GPIO port pin registers

Reading these registers returns the current state of the pins read, regardless of direction, masking, or alternate functions, except that pins configured as analog I/O always read as 0s. Writing these registers loads the output bits of the pins written to, regardless of the Mask register.

Table 203. GPIO port pin register (PIN[0:1], address 0xA000 2100 (PIN0) to 0xA0000 2104 (PIN1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	PORT	Reads pin states or loads output bits (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = Read: pin is low; write: clear output bit. 1 = Read: pin is high; write: set output bit.	ext	R/W

13.5.6 GPIO masked port pin registers

These registers are similar to the PORT registers, except that the value read is masked by ANDing with the inverted contents of the corresponding MASK register, and writing to one of these registers only affects output register bits that are enabled by zeros in the corresponding MASK register

Table 204. GPIO masked port pin register (MPIN[0:1], address 0xA000 2180 (MPIN0) to 0xA0000 2184 (MPIN1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	MPORTP	Masked port register (bit 0 = PIOm_0, bit 1 =PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = Read: pin is LOW and/or the corresponding bit in the MASK register is 1; write: clear output bit if the corresponding bit in the MASK register is 0. 1 = Read: pin is HIGH and the corresponding bit in the MASK register is 0; write: set output bit if the corresponding bit in the MASK register is 0.	ext	R/W

13.5.7 GPIO port set registers

Output bits can be set by writing ones to these registers, regardless of MASK registers. Reading from these register returns the port's output bits, regardless of pin directions.

Table 205. GPIO port set register (SET[0:1], address 0xA000 2200 (SET0) to 0xA000 2204 (SET1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	SETP	Read or set output bits (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = Read: output bit: write: no operation. 1 = Read: output bit; write: set output bit.	0	R/W

13.5.8 GPIO port clear registers

Output bits can be cleared by writing ones to these write-only registers, regardless of MASK registers.

Table 206. GPIO port clear register (CLR[0:1], address 0xA000 2280 (CLR0) to 0xA000 2284 (CLR1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	CLRP	Clear output bits (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = No operation. 1 = Clear output bit.	NA	WO

13.5.9 GPIO port toggle registers

Output bits can be set by writing ones to these write-only registers, regardless of MASK registers.

Table 207. GPIO port toggle register (NOT[0:1], address 0xA000 2300 (NOT0) to 0xA000 2304(NOT1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	NOTP	Toggle output bits (bit 0 = PIOm_0, bit 1 =PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = no operation. 1 = Toggle output bit.	NA	WO

13.5.10 GPIO port direction set registers

Direction bits can be set by writing ones to these registers.

Table 208. GPIO port direction set register (DIRSET[0:1], address 0xA000 2380 (DIRSET0) to 0xA000 2384 (DIRSET1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	DIRSETP	Set direction bits (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = No operation. 1 = Set direction bit.	0	WO

13.5.11 GPIO port direction clear registers

Direction bits can be cleared by writing ones to these write-only registers.

Table 209. GPIO port direction clear register (DIRCLR[0:1], 0xA000 2400 (DIRCLR0) to 0xA000 2404 (DIRCLR1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	DIRCLRP	Clear direction bits (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = No operation. 1 = Clear direction bit.	NA	WO

13.5.12 GPIO port direction toggle registers

Direction bits can be set by writing ones to these write-only registers.

Table 210. GPIO port direction toggle register (DIRNOT[0:1], address 0xA000 2480 (DIRNOT0) to 0xA000 2484 (DIRNOT1)) bit description

Bit	Symbol	Description	Reset value	Access
31:0	DIRNOTP	Toggle direction bits (bit 0 = PIOm_0, bit 1 = PIOm_1, ..., bit 31 = PIOm_31). m = port 0 to 1; n = pin 0 to 31 for port 0 and pin 0 to 21 for port 1. 0 = no operation. 1 = Toggle direction bit.	NA	WO

13.6 Functional description

13.6.1 Reading pin state

Software can read the state of all GPIO pins except those selected for analog input or output in the “I/O Configuration” logic. A pin does not have to be selected for GPIO in “I/O Configuration” in order to read its state. There are four ways to read pin state:

- The state of a single pin can be read with 7 high-order zeros from a Byte Pin register.
- The state of a single pin can be read in all bits of a byte, halfword, or word from a Word Pin register.

- The state of multiple pins in a port can be read as a byte, halfword, or word from a PORT register.
- The state of a selected subset of the pins in a port can be read from a Masked Port (MPORT) register. Pins having a 1 in the port's Mask register will read as 0 from its MPORT register.

13.6.2 GPIO output

Each GPIO pin has an output bit in the GPIO block. These output bits are the targets of write operations to the pins. Two conditions must be met in order for a pin's output bit to be driven onto the pin:

1. The pin must be selected for GPIO operation in the switch matrix (this is the default), and
2. the pin must be selected for output by a 1 in its port's DIR register.

If either or both of these conditions is (are) not met, writing to the pin has no effect.

There are multiple ways to change GPIO output bits:

- Writing to a Byte Pin register loads the output bit from the least significant bit.
- Writing to a Word Pin register loads the output bit with the OR of all of the bits written. (This feature follows the definition of truth of a multi-bit value in programming languages.)
- Writing to a port's PORT register loads the output bits of all the pins written to.
- Writing to a port's MPORT register loads the output bits of pins identified by zeros in corresponding positions of the port's MASK register.
- Writing ones to a port's SET register sets output bits.
- Writing ones to a port's CLR register clears output bits.
- Writing ones to a port's NOT register toggles/complements/inverts output bits.

The state of a port's output bits can be read from its SET register. Reading any of the registers described in Section 13.6.1 returns the state of pins, regardless of their direction or alternate functions.

13.6.3 Masked I/O

A port's MASK register defines which of its pins should be accessible in its MPORT register. Zeroes in MASK enable the corresponding pins to be read from and written to MPORT. Ones in MASK force a pin to read as 0 and its output bit to be unaffected by writes to MPORT. When a port's MASK register contains all zeros, its PORT and MPORT registers operate identically for reading and writing.

Applications in which interrupts can result in Masked GPIO operation, or in task switching among tasks that do Masked GPIO operation, must treat code that uses the Mask register as a protected/restricted region. This can be done by interrupt disabling or by using a semaphore.

The simpler way to protect a block of code that uses a MASK register is to disable interrupts before setting the MASK register, and re-enable them after the last operation that uses the MPORT or MASK register.

More efficiently, software can dedicate a semaphore to the MASK registers, and set/capture the semaphore controlling exclusive use of the MASK registers before setting the MASK registers, and release the semaphore after the last operation that uses the MPORT or MASK registers.

13.6.4 GPIO direction

Each pin in a GPIO port can be configured as input or output using the DIR registers. The direction of individual pins can be set, cleared, or toggled using the DIRSET, DIRCLR, and DIRNOT registers.

13.6.5 Recommended practices

The following lists some recommended uses for using the GPIO port registers:

- For initial setup after Reset or re-initialization, write the PORT registers.
- To change the state of one pin, write a Byte Pin or Word Pin register.
- To change the state of multiple pins at a time, write the SET and/or CLR registers.
- To change the state of multiple pins in a tightly controlled environment like a software state machine, consider using the NOT register. This can require less write operations than SET and CLR.
- To read the state of one pin, read a Byte Pin or Word Pin register.
- To make a decision based on multiple pins, read and mask a PORT register.

UM11607

Chapter 14: Pin interrupts/pattern match engine

Rev. 3 — April 2023 User manual

14.1 How to read this chapter

The pin interrupt generator and the pattern match engine are available on all LPC86x parts.

14.2 Features

- Pin interrupts
  - Up to eight pins can be selected from all GPIO pins as edge- or level-sensitive interrupt requests. Each request creates a separate interrupt in the NVIC.
  - Edge-sensitive interrupt pins can interrupt on rising or falling edges or both.
  - Level-sensitive interrupt pins can be HIGH- or LOW-active.
- Pattern match engine
  - Up to eight pins can be selected from all GPIO pins to contribute to a boolean expression. The boolean expression consists of specified levels and/or transitions on various combinations of these pins.
  - Each bit slice minterm (product term) comprising the specified boolean expression can generate its own, dedicated interrupt request.
  - Any occurrence of a pattern match can be programmed to also generate an RXEV notification to the ARM CPU. The RXEV signal can be connected to a pin.
  - Pattern match can be used, in conjunction with software, to create complex state machines based on pin inputs.

14.3 Basic configuration

- Pin interrupts:
  - Select up to eight external interrupt pins from all GPIO port pins in the SYSCON block (Table 113). The pin selection process is the same for pin interrupts and the pattern match engine. The two features are mutually exclusive.
  - Enable the clock to the pin interrupt register block in the SYSAHBCLKCTRL register (Table 94, bit 28).
  - If you want to use the pin interrupts to wake up the part from deep-sleep mode or power-down mode, enable the pin interrupt wake-up feature in the STARTERP0 register (Table 114).
  - Each selected pin interrupt is assigned to one interrupt in the NVIC (interrupts #24 to #31 for pin interrupts 0 to 7).
- Pattern match engine:
  - Select up to eight external pins from all GPIO port pins in the SYSCON block (Table 113). The pin selection process is the same for pin interrupts and the pattern match engine. The two features are mutually exclusive.

- Enable the clock to the pin interrupt register block in the SYSAHBCLKCTRL register (Table 94, bit 28).
- Each bit slice of the pattern match engine is assigned to one interrupt in the NVIC (interrupts #24 to #31 for slices 0 to 7).
- The combined interrupt from all slices or slice combinations can be connected to the ARM RXEV request and to pin function GPIO_INT_BMAT through the switch matrix movable function register (PINASSIGN11, Table 136).

14.3.1 Configure pins as pin interrupts or as inputs to the pattern match engine

Follow these steps to configure pins as pin interrupts:

1. Determine the pins that serve as pin interrupts on the LPC86x package. See the data sheet for determining the GPIO port pin number associated with the package pin.
2. For each pin interrupt, program the GPIO port pin number into one of the eight PINTSEL registers in the SYSCON block.

   Remark: The port pin number serves to identify the pin to the PINTSEL register. Any function, including GPIO, can be assigned to this pin through the switch matrix.
3. Enable each pin interrupt in the NVIC.

Once the pin interrupts or pattern match inputs are configured, you can set up the pin interrupt detection levels or the pattern match boolean expression.

See Section 8.6.42 “Pin interrupt select registers” in the SYSCON block for the PINTSEL registers.

14.4 Pin description

The inputs to the pin interrupt and pattern match engine are determined by the pin interrupt select registers in the SYSCON block. See Section 8.6.42 “Pin interrupt select registers”.

The pattern match engine output is assigned to an external pin through the switch matrix.

See Section 11.3.1 “Connect an internal signal to a package pin” for the steps that you need to follow to assign the GPIO pattern match function to a pin.

Table 211. Pin interrupt/pattern match engine pin description

Function	Direction	Pin	Description	SWM register	Reference
GPIO_INT_BMAT	O	any	GPIO pattern match output	PINASSIGN11	Table 136

14.5 General description

Pins with configurable functions can serve as external interrupts or inputs to the pattern match engine. You can configure up to eight pins total using the PINTSEL registers in the SYSCON block for these features.

14.5.1 Pin interrupts

From all available GPIO pins, up to eight pins can be selected in the system control block to serve as external interrupt pins (see Table 113). The external interrupt pins are connected to eight individual interrupts in the NVIC and are created based on rising or falling edges or on the input level on the pin.

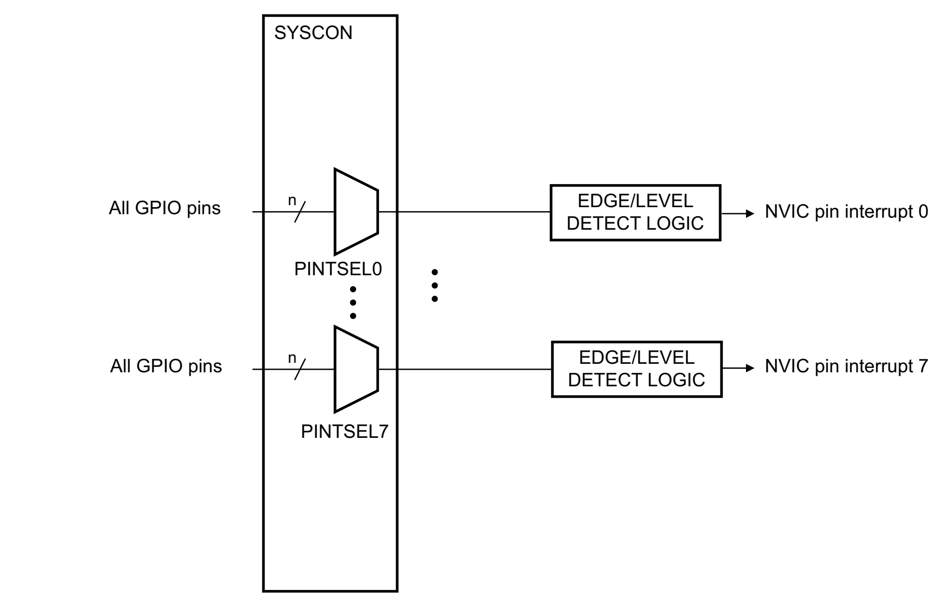
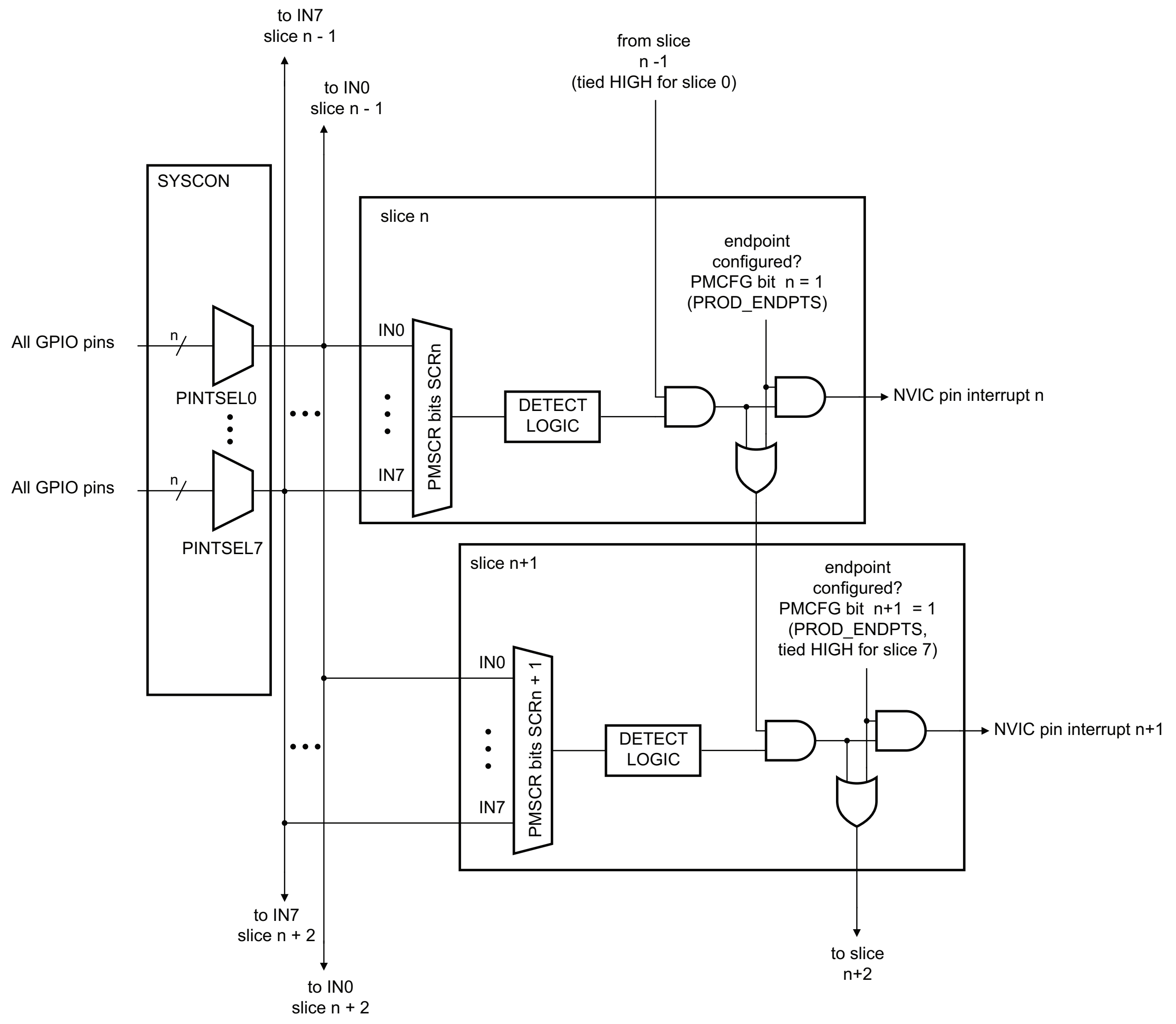


Fig 14. Pin interrupt connections

14.5.2 Pattern match engine

The pattern match feature allows complex boolean expressions to be constructed from the same set of eight GPIO pins that were selected for the GPIO pin interrupts. Each term in the boolean expression is implemented as one slice of the pattern match engine. A slice consists of an input selector and a detect logic. The slice input selector selects one input from the available eight inputs with each input connected to a pin by the input's PINTSEL register.

The detect logic monitors the selected input continuously and creates a HIGH output if the input qualifies as detected. Several terms can be combined to a minterm by designating a slice as an endpoint of the expression. A pin interrupt for this slice is asserted when the minterm evaluates as true.



See Figure 16 for the detect logic block.

Fig 15. Pattern match engine connections

The detect logic of each slice can detect the following events on the selected input:

- Edge with memory (sticky): A rising edge, a falling edge, or a rising or falling edge that is detected at any time after the edge-detection mechanism has been cleared. The input qualifies as detected (the detect logic output remains HIGH) until the pattern match engine detect logic is cleared again.
- Event (non-sticky): Every time an edge (rising or falling) is detected, the detect logic output for this pin goes HIGH. This bit is cleared after one clock cycle, and the detect logic can detect another edge,
- Level: A HIGH or LOW level on the selected input.

Figure 16 shows the details of the edge detection logic for each slice.

You can combine a sticky event with non-sticky events to create a pin interrupt whenever a rising or falling edge occurs after a qualifying edge event.

You can create a time window during which rising or falling edges can create a pin interrupt by combining a level detect with an event detect. See Section 14.7.3 for details.

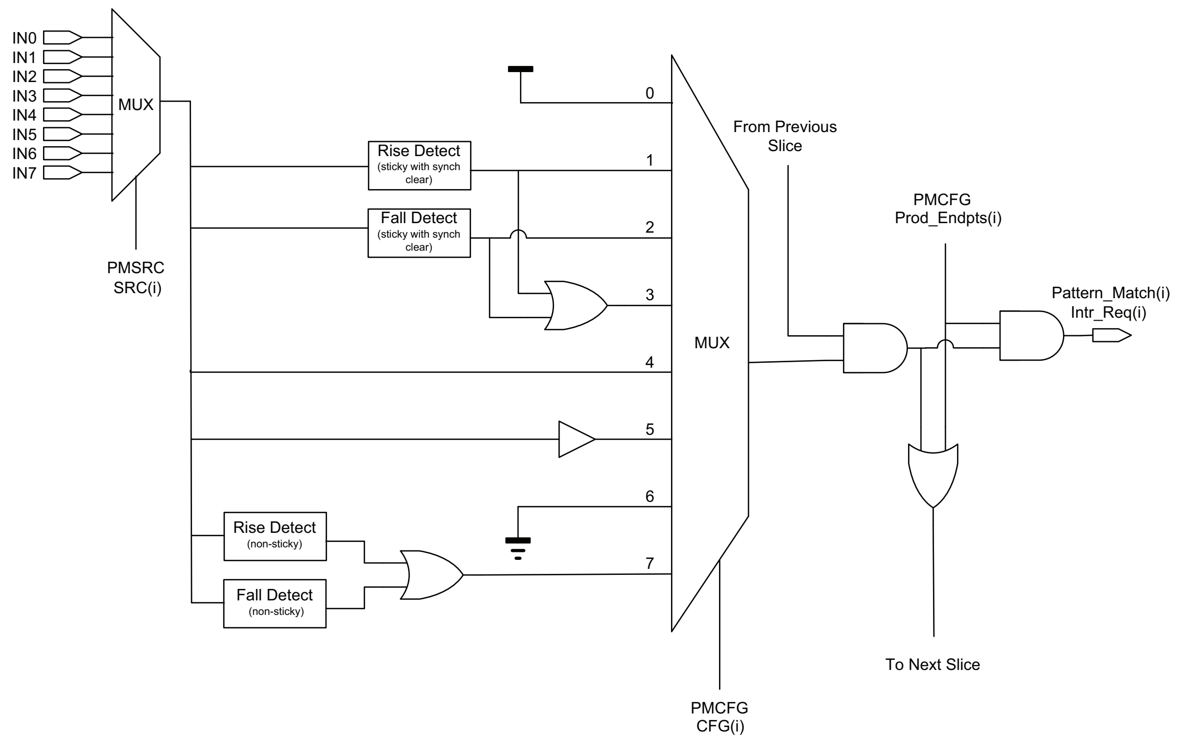


Fig 16. Pattern match bit slice with detect logic

14.5.2.1 Inputs and outputs of the pattern match engine

The connections between the pins and the pattern match engine are shown in Figure 15. All inputs to the pattern match engine are selected in the SYSCON block and can be GPIO port pins or another pin function depending on the switch matrix configuration.

The pattern match logic continuously monitors the eight inputs and generates interrupts when any one or more minterms (product terms) of the specified boolean expression is matched. A separate interrupt request is generated for each individual minterm.

In addition, the pattern match module can be enabled to generate a Receive Event (RXEV) output to the ARM core when a boolean expression is true (i.e. when any minterm is matched).

The RXEV output is also be routed to GPIO_INT_BMAT pin. This allows the GPIO module to provide a rudimentary programmable logic capability employing up to eight inputs and one output.

The pattern match function utilizes the same eight interrupt request lines as the pin interrupts, so these two features are mutually exclusive as far as interrupt generation is concerned. A control bit is provided to select whether interrupt requests are generated in response to the standard pin interrupts or to pattern matches. Note that, if the pin interrupts are selected, the RXEV request to the CPU can still be enabled for pattern matches.

Remark: Pattern matching cannot be used to wake the part up from Deep-sleep or power-down mode. Pin interrupts must be selected in order to use the pins for wake-up.

14.5.2.2 Boolean expressions

The pattern match module is constructed of eight bit-slice elements. Each bit slice is programmed to represent one component of one minterm (product term) within the boolean expression. The interrupt request associated with the last bit slice for a particular minterm will be asserted whenever that minterm is matched.
(See bit slice drawing Figure 16).

The pattern match capability can be used to create complex software state machines. Each minterm (and its corresponding individual interrupt) represents a different transition event to a new state. Software can then establish the new set of conditions (that is a new boolean expression) that will cause a transition out of the current state.

Example:

Assume the expression: (IN0)~(IN1)(IN3)^ + (IN1)(IN2) + (IN0)~(IN3)~(IN4) is specified through the registers PMSRC (Table 224) and PMCFG (Table 225). Each term in the boolean expression, (IN0), ~(IN1), (IN3)^, etc., represents one bit slice of the pattern match engine.

- In the first minterm (IN0)~(IN1)(IN3)^, bit slice 0 monitors for a high-level on input (IN0), bit slice 1 monitors for a low level on input (IN1) and bit slice 2 monitors for a rising-edge on input (IN3). If this combination is detected, that is if all three terms are true, the interrupt associated with bit slice 2 (PININT2_IRQ) will be asserted.
- In the second minterm (IN1)(IN2), bit slice 3 monitors input (IN1) for a high level, bit slice 4 monitors input (IN2) for a high level. If this combination is detected, the interrupt associated with bit slice 4 (PININT4_IRQ) will be asserted.
- In the third minterm (IN0)~(IN3)~(IN4), bit slice 5 monitors input (IN0) for a high level, bit slice 6 monitors input (IN3) for a low level, and bit slice 7 monitors input (IN4) for a low level. If this combination is detected, the interrupt associated with bit slice 7(PININT7_IRQ) will be asserted.

- The ORed result of all three minterms asserts the RXEV request to the CPU and the GPIO_INT_BMAT output. That is, if any of the three minterms are true, the output is asserted.

Related links:

Section 14.7.2

14.6 Register description

Table 212. Register overview: Pin interrupts and pattern match engine (base address: 0xA000 4000)

Name	Access	Address offset	Description	Reset value	Reference
ISEL	R/W	0x000	Pin Interrupt Mode register	0	Table 213
IENR	R/W	0x004	Pin interrupt level or rising edge interrupt enable register	0	Table 214
SIENR	WO	0x008	Pin interrupt level or rising edge interrupt set register	NA	Table 215
CIENR	WO	0x00C	Pin interrupt level (rising edge interrupt) clear register	NA	Table 216
IENF	R/W	0x010	Pin interrupt active level or falling edge interrupt enable register	0	Table 217
SIENF	WO	0x014	Pin interrupt active level or falling edge interrupt set register	NA	Table 218
CIENF	WO	0x018	Pin interrupt active level or falling edge interrupt clear register	NA	Table 219
RISE	R/W	0x01C	Pin interrupt rising edge register	0	Table 220
FALL	R/W	0x020	Pin interrupt falling edge register	0	Table 221
IST	R/W	0x024	Pin interrupt status register	0	Table 222
PMCTRL	R/W	0x028	Pattern match interrupt control register	0	Table 223
PMSRC	R/W	0x02C	Pattern match interrupt bit-slice source register	0	Table 224
PMCFG	R/W	0x030	Pattern match interrupt bit slice configuration register	0	Table 225

14.6.1 Pin interrupt mode register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the ISEL register determines whether the interrupt is edge or level sensitive.

Table 213. Pin interrupt mode register (ISEL, address 0xA000 4000) bit description

Bit	Symbol	Description	Reset value	Access
7:0	PMODE	Selects the interrupt mode for each pin interrupt. Bit n configures the pin interrupt selected in PINTSELn. 0 = Edge sensitive 1 = Level sensitive	0	R/W
31:8	-	Reserved.	-	-

14.6.2 Pin interrupt level or rising edge interrupt enable register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the IENR register enables the interrupt depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the rising edge interrupt is enabled.
- If the pin interrupt mode is level sensitive (PMODE = 1), the level interrupt is enabled. The IENF register configures the active level (HIGH or LOW) for this interrupt.

Table 214. Pin interrupt level or rising edge interrupt enable register (IENR, address 0xA000 4004) bit description

Bit	Symbol	Description	Reset value	Access
7:0	ENRL	Enables the rising edge or level interrupt for each pin interrupt. Bit n configures the pin interrupt selected in PINTSELn. 0 = Disable rising edge or level interrupt. 1 = Enable rising edge or level interrupt.	0	R/W
31:8	-	Reserved.	-	-

14.6.3 Pin interrupt level or rising edge interrupt set register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the SIENR register sets the corresponding bit in the IENR register depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the rising edge interrupt is set.
- If the pin interrupt mode is level sensitive (PMODE = 1), the level interrupt is set.

Table 215. Pin interrupt level or rising edge interrupt set register (SIENR, address 0xA000 4008) bit description

Bit	Symbol	Description	Reset value	Access
7:0	SETENRL	Ones written to this address set bits in the IENR, thus enabling interrupts. Bit n sets bit n in the IENR register. 0 = No operation. 1 = Enable rising edge or level interrupt.	NA	WO
31:8	-	Reserved.	-	-

14.6.4 Pin interrupt level or rising edge interrupt clear register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the CIENR register clears the corresponding bit in the IENR register depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the rising edge interrupt is cleared.
- If the pin interrupt mode is level sensitive (PMODE = 1), the level interrupt is cleared.

Table 216. Pin interrupt level or rising edge interrupt clear register (CIENR, address 0xA000 400C) bit description

Bit	Symbol	Description	Reset value	Access
7:0	CENRL	Ones written to this address clear bits in the IENR, thus disabling the interrupts. Bit n clears bit n in the IENR register. 0 = No operation. 1 = Disable rising edge or level interrupt.	NA	WO
31:8	-	Reserved.	-	-

14.6.5 Pin interrupt active level or falling edge interrupt enable register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the IENF register enables the falling edge interrupt or the configures the level sensitivity depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the falling edge interrupt is enabled.
- If the pin interrupt mode is level sensitive (PMODE = 1), the active level of the level interrupt (HIGH or LOW) is configured.

Table 217. Pin interrupt active level or falling edge interrupt enable register (IENF, address 0xA000 4010) bit description

Bit	Symbol	Description	Reset value	Access
7:0	ENAF	Enables the falling edge or configures the active level interrupt for each pin interrupt. Bit n configures the pin interrupt selected in PINTSELn. 0 = Disable falling edge interrupt or set active interrupt level LOW. 1 = Enable falling edge interrupt enabled or set active interrupt level HIGH.	0	R/W
31:8	-	Reserved.	-	-

14.6.6 Pin interrupt active level or falling edge interrupt set register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the SIENF register sets the corresponding bit in the IENF register depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the falling edge interrupt is set.
- If the pin interrupt mode is level sensitive (PMODE = 1), the HIGH-active interrupt is selected.

Table 218. Pin interrupt active level or falling edge interrupt set register (SIENF, address 0xA000 4014) bit description

Bit	Symbol	Description	Reset value	Access
7:0	SETENAF	Ones written to this address set bits in the IENF, thus enabling interrupts. Bit n sets bit n in the IENF register. 0 = No operation. 1 = Select HIGH-active interrupt or enable falling edge interrupt.	NA	WO
31:8	-	Reserved.	-	-

14.6.7 Pin interrupt active level or falling edge interrupt clear register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.42), one bit in the CIENF register sets the corresponding bit in the IENF register depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the falling edge interrupt is cleared.
- If the pin interrupt mode is level sensitive (PMODE = 1), the LOW-active interrupt is selected.

Table 219. Pin interrupt active level or falling edge interrupt clear register (CIENF, address 0xA000 4018) bit description

Bit	Symbol	Description	Reset value	Access
7:0	CENAF	Ones written to this address clears bits in the IENF, thus disabling interrupts. Bit n clears bit n in the IENF register. 0 = No operation. 1 = LOW-active interrupt selected or falling edge interrupt disabled.	NA	WO
31:8	-	Reserved.	-	-

14.6.8 Pin interrupt rising edge register

This register contains ones for pin interrupts selected in the PINTSELn registers (see Section 8.6.42) on which a rising edge has been detected. Writing ones to this register clears rising edge detection. Ones in this register assert an interrupt request for pins that are enabled for rising-edge interrupts. All edges are detected for all pins selected by the PINTSELn registers, regardless of whether they are interrupt-enabled.

Table 220. Pin interrupt rising edge register (RISE, address 0xA000 401C) bit description

Bit	Symbol	Description	Reset value	Access
7:0	RDET	Rising edge detect. Bit n detects the rising edge of the pin selected in PINTSELn. Read 0: No rising edge has been detected on this pin since Reset or the last time a one was written to this bit. Write 0: no operation. Read 1: a rising edge has been detected since Reset or the last time a one was written to this bit. Write 1: clear rising edge detection for this pin.	0	R/W
31:8	-	Reserved.	-	-

14.6.9 Pin interrupt falling edge register

This register contains ones for pin interrupts selected in the PINTSELn registers (see Section 8.6.42) on which a falling edge has been detected. Writing ones to this register clears falling edge detection. Ones in this register assert an interrupt request for pins that are enabled for falling-edge interrupts. All edges are detected for all pins selected by the PINTSELn registers, regardless of whether they are interrupt-enabled.

Table 221. Pin interrupt falling edge register (FALL, address 0xA000 4020) bit description

Bit	Symbol	Description	Reset value	Access
7:0	FDET	Falling edge detect. Bit n detects the falling edge of the pin selected in PINTSELn. Read 0: No falling edge has been detected on this pin since Reset or the last time a one was written to this bit. Write 0: no operation. Read 1: a falling edge has been detected since Reset or the last time a one was written to this bit. Write 1: clear falling edge detection for this pin.	0	R/W
31:8	-	Reserved.	-	-

14.6.10 Pin interrupt status register

Reading this register returns ones for pin interrupts that are currently requesting an interrupt. For pins identified as edge-sensitive in the Interrupt Select register, writing ones to this register clears both rising- and falling-edge detection for the pin. For level-sensitive pins, writing ones inverts the corresponding bit in the Active level register, thus switching the active level on the pin.

Table 222. Pin interrupt status register (IST, address 0xA000 4024) bit description

Bit	Symbol	Description	Reset value	Access
7:0	PSTAT	Pin interrupt status. Bit n returns the status, clears the edge interrupt, or inverts the active level of the pin selected in PINTSELn. Read 0: interrupt is not being requested for this interrupt pin. Write 0: no operation. Read 1: interrupt is being requested for this interrupt pin. Write 1 (edge-sensitive): clear rising- and falling-edge detection for this pin. Write 1 (level-sensitive): switch the active level for this pin (in the IENF register).	0	R/W
31:8	-	Reserved.	-	-

14.6.11 Pattern Match Interrupt Control Register

The pattern match control register contains one bit to select pattern-match interrupt generation (as opposed to pin interrupts which share the same interrupt request lines), and another to enable the RXEV output to the CPU. This register also allows the current state of any pattern matches to be read.

If the pattern match feature is not used (either for interrupt generation or for RXEV assertion) bits SEL_PMATCH and ENA_RXEV of this register should be left at 0 to conserve power.

Remark: Set up the pattern-match configuration in the PMSRC and PMCFG registers before writing to this register to enable (or re-enable) the pattern-match functionality. This eliminates the possibility of spurious interrupts as the feature is being enabled.

Table 223. Pattern match interrupt control register (PMCTRL, address 0xA000 4028) bit description

Bit	Symbol	Value	Description	Reset value
0	SEL_PMATCH		Specifies whether the 8 pin interrupts are controlled by the pin interrupt function or by the pattern match function.	0
		0	Pin interrupt. Interrupts are driven in response to the standard pin interrupt function	
		1	Pattern match. Interrupts are driven in response to pattern matches.	
1	ENA_RXEV		Enables the RXEV output to the ARM CPU and/or to a GPIO output when the specified boolean expression evaluates to true.	0
		0	Disabled. RXEV output to the CPU is disabled.	
		1	Enabled. RXEV output to the CPU is enabled.	
23:2	-		Reserved. Do not write 1s to unused bits.	0
31:24	PMAT	-	This field displays the current state of pattern matches. A 1 in any bit of this field indicates that the corresponding product term is matched by the current state of the appropriate inputs.	0x0

14.6.12 Pattern Match Interrupt Bit-Slice Source register

The bit-slice source register specifies the input source for each of the eight pattern match bit slices.

Each of the possible eight inputs is selected in the pin interrupt select registers in the SYSCON block. See Section 8.6.42. Input 0 corresponds to the pin selected in the PINTSEL0 register, input 1 corresponds to the pin selected in the PINTSEL1 register, and so forth.

Remark: Writing any value to either the PMCFG register or the PMSRC register, or disabling the pattern-match feature (by clearing both the SEL_PMATCH and ENA_RXEV bits in the PMCTRL register to zeros) will erase all edge-detect history.

Table 224. Pattern match bit-slice source register (PMSRC, address 0xA000 402C) bit description

Bit	Symbol	Value	Description	Reset value
7:0	Reserved		Software should not write 1s to unused bits.	0

Table 224. Pattern match bit-slice source register (PMSRC, address 0xA000 402C) bit description

Bit	Symbol	Value	Description	Reset value
10:8	SRC0		Selects the input source for bit slice 0	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 0.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 0.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 0.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 0.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 0.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 0.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 0.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 0.	
13:11	SRC1		Selects the input source for bit slice 1	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 1.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 1.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 1.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 1.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 1.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 1.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 1.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 1.	

Table 224. Pattern match bit-slice source register (PMSRC, address 0xA000 402C) bit description

Bit	Symbol	Value	Description	Reset value
16:14	SRC2		Selects the input source for bit slice 2	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 2.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 2.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 2.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 2.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 2.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 2.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 2.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 2.	
19:17	SRC3		Selects the input source for bit slice 3	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 3.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 3.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 3.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 3.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 3.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 3.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 3.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 3.	

Table 224. Pattern match bit-slice source register (PMSRC, address 0xA000 402C) bit description

Bit	Symbol	Value	Description	Reset value
22:20	SRC4		Selects the input source for bit slice 4	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 4.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 4.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 4.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 4.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 4.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 4.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 4.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 4.	
25:23	SRC5		Selects the input source for bit slice 5	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 5.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 5.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 5.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 5.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 5.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 5.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 5.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 5.	

Table 224. Pattern match bit-slice source register (PMSRC, address 0xA000 402C) bit description

Bit	Symbol	Value	Description	Reset value
28:26	SRC6		Selects the input source for bit slice 6	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 6.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 6.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 6.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 6.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 6.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 6.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 6.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 6.	
31:29	SRC7		Selects the input source for bit slice 7	0
		0x0	Input 0. Selects the pin selected in the PINTSEL0 register as the source to bit slice 7.	
		0x1	Input 1. Selects the pin selected in the PINTSEL1 register as the source to bit slice 7.	
		0x2	Input 2. Selects the pin selected in the PINTSEL2 register as the source to bit slice 7.	
		0x3	Input 3. Selects the pin selected in the PINTSEL3 register as the source to bit slice 7.	
		0x4	Input 4. Selects the pin selected in the PINTSEL4 register as the source to bit slice 7.	
		0x5	Input 5. Selects the pin selected in the PINTSEL5 register as the source to bit slice 7.	
		0x6	Input 6. Selects the pin selected in the PINTSEL6 register as the source to bit slice 7.	
		0x7	Input 7. Selects the pin selected in the PINTSEL7 register as the source to bit slice 7.	

14.6.13 Pattern Match Interrupt Bit Slice Configuration register

The bit-slice configuration register configures the detect logic and contains bits to select from among eight alternative conditions for each bit slice that cause that bit slice to contribute to a pattern match. The seven LSBs of this register specify which bit-slices are the end-points of product terms in the boolean expression (i.e. where OR terms are to be inserted in the expression).

Two types of edge detection on each input are possible:

- Sticky: A rising edge, a falling edge, or a rising or falling edge that is detected at any time after the edge-detection mechanism has been cleared. The input qualifies as detected (the detect logic output remains HIGH) until the pattern match engine detect logic is cleared again.
- Non-sticky: Every time an edge (rising or falling) is detected, the detect logic output for this pin goes HIGH. This bit is cleared after one clock cycle, and the edge detect logic can detect another edge,

Remark: To clear the pattern match engine detect logic, write any value to either the PMCFG register or the PMSRC register, or disable the pattern-match feature (by clearing both the SEL_PMATCH and ENA_RXEV bits in the PMCTRL register to zeros). This will erase all edge-detect history.

To select whether a slice marks the final component in a minterm of the boolean expression, write a 1 in the corresponding PROD_ENPTSn bit. Setting a term as the final component has two effects:

1. The interrupt request associated with this bit slice will be asserted whenever a match to that product term is detected.
2. The next bit slice will start a new, independent product term in the boolean expression (i.e. an OR will be inserted in the boolean expression following the element controlled by this bit slice).

Table 225. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description

Bit	Symbol	Value	Description	Reset value
0	PROD_ENDPTS0		Determines whether slice 0 is an endpoint.	0
		0	No effect. Slice 0 is not an endpoint.	
		1	endpoint. Slice 0 is the endpoint of a product term (minterm). Pin interrupt 0 in the NVIC is raised if the minterm evaluates as true.	
1	PROD_ENDPTS1		Determines whether slice 1 is an endpoint.	0
		0	No effect. Slice 1 is not an endpoint.	
		1	endpoint. Slice 1 is the endpoint of a product term (minterm). Pin interrupt 1 in the NVIC is raised if the minterm evaluates as true.	
2	PROD_ENDPTS2		Determines whether slice 2 is an endpoint.	0
		0	No effect. Slice 2 is not an endpoint.	
		1	endpoint. Slice 2 is the endpoint of a product term (minterm). Pin interrupt 2 in the NVIC is raised if the minterm evaluates as true.	
3	PROD_ENDPTS3		Determines whether slice 3 is an endpoint.	0
		0	No effect. Slice 3 is not an endpoint.	
		1	endpoint. Slice 3 is the endpoint of a product term (minterm). Pin interrupt 3 in the NVIC is raised if the minterm evaluates as true.	
4	PROD_ENDPTS4		Determines whether slice 4 is an endpoint.	0
		0	No effect. Slice 4 is not an endpoint.	
		1	endpoint. Slice 4 is the endpoint of a product term (minterm). Pin interrupt 4 in the NVIC is raised if the minterm evaluates as true.	

Table 225. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
5	PROD_ENDPTS5		Determines whether slice 5 is an endpoint.	0
		0	No effect. Slice 5 is not an endpoint.	
		1	endpoint. Slice 5 is the endpoint of a product term (minterm). Pin interrupt 5 in the NVIC is raised if the minterm evaluates as true.	
6	PROD_ENDPTS6		Determines whether slice 6 is an endpoint.	0
		0	No effect. Slice 6 is not an endpoint.	
		1	endpoint. Slice 6 is the endpoint of a product term (minterm). Pin interrupt 6 in the NVIC is raised if the minterm evaluates as true.	
7	-		Reserved. Bit slice 7 is automatically considered a product end point.	0
10:8	CFG0		Specifies the match contribution condition for bit slice 0.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	

Table 225. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
13:11	CFG1		Specifies the match contribution condition for bit slice 1.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	
16:14	CFG2		Specifies the match contribution condition for bit slice 2.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	

Table 225. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
19:17	CFG3		Specifies the match contribution condition for bit slice 3.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	
22:20	CFG4		Specifies the match contribution condition for bit slice 4.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	

Table 225. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
25:23	CFG5		Specifies the match contribution condition for bit slice 5.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	
28:26	CFG6		Specifies the match contribution condition for bit slice 6.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	

Table 225. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
31:29	CFG7		Specifies the match contribution condition for bit slice 7.	0b000
		0x0	Constant HIGH. This bit slice always contributes to a product term match.	
		0x1	Sticky rising edge. Match occurs if a rising edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x2	Sticky falling edge. Match occurs if a falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x3	Sticky rising or falling edge. Match occurs if either a rising or falling edge on the specified input has occurred since the last time the edge detection for this bit slice was cleared. This bit is only cleared when the PMCFG or the PMSRC registers are written to.	
		0x4	High level. Match (for this bit slice) occurs when there is a high level on the input specified for this bit slice in the PMSRC register.	
		0x5	Low level. Match occurs when there is a low level on the specified input.	
		0x6	Constant 0. This bit slice never contributes to a match (should be used to disable any unused bit slices).	
		0x7	Event. Non-sticky rising or falling edge. Match occurs on an event - i.e. when either a rising or falling edge is first detected on the specified input (this is a non-sticky version of value 0x3). This bit is cleared after one clock cycle.	

14.7 Functional description

14.7.1 Pin interrupts

In this interrupt facility, up to 8 pins are identified as interrupt sources by the Pin Interrupt Select registers (PINTSEL0-7). All registers in the pin interrupt block contain 8 bits, corresponding to the pins called out by the PINTSEL0-7 registers. The ISEL register defines whether each interrupt pin is edge- or level-sensitive. The RISE and FALL registers detect edges on each interrupt pin, and can be written to clear (and set) edge detection. The IST register indicates whether each interrupt pin is currently requesting an interrupt, and this register can also be written to clear interrupts.

The other pin interrupt registers play different roles for edge-sensitive and level-sensitive pins, as described in Table 226.

Table 226. Pin interrupt registers for edge- and level-sensitive pins

Name	Edge-sensitive function	Level-sensitive function
IENR	Enables rising-edge interrupts.	Enables level interrupts.
SIENR	Write to enable rising-edge interrupts.	Write to enable level interrupts.
CIENR	Write to disable rising-edge interrupts.	Write to disable level interrupts.
IENF	Enables falling-edge interrupts.	Selects active level.
SIENF	Write to enable falling-edge interrupts.	Write to select high-active.
CIENF	Write to disable falling-edge interrupts.	Write to select low-active.

14.7.2 Pattern Match engine example

Suppose the desired boolean pattern to be matched is:
(IN1) + (IN1 * IN2) + (~IN2 * ~IN3 * IN6fe) + (IN5 * IN7ev)

with:

IN6fe = (sticky) falling-edge on input 6

IN7ev = (non-sticky) event (rising or falling edge) on input 7

Each individual term in the expression shown above is controlled by one bit-slice. To specify this expression, program the pattern match bit slice source and configuration register fields as follows:

- PMSRC register (Table 224):
  - Since bit slice 5 will be used to detect a sticky event on input 6, you can write a 1 to the SRC5 bits to clear any pre-existing edge detects on bit slice 5.
  - SRC0: 001 - select input 1 for bit slice 0
  - SRC1: 001 - select input 1 for bit slice 1
  - SRC2: 010 - select input 2 for bit slice 2
  - SRC3: 010 - select input 2 for bit slice 3
  - SRC4: 011 - select input 3 for bit slice 4
  - SRC5: 110 - select input 6 for bit slice 5
  - SRC6: 101 - select input 5 for bit slice 6
  - SRC7: 111 - select input 7 for bit slice 7
- PMCFG register (Table 225):
  - PROD_ENDPTS0 = 1
  - PROD_ENDPTS02 = 1
  - PROD_ENDPTS5 = 1
  - All other slices are not product term endpoints and their PROD_ENDPTS bits are 0. Slice 7 is always a product term endpoint and does not have a register bit associated with it.
  - = 0100101 - bit slices 0, 2, 5, and 7 are product-term endpoints. (Bit slice 7 is an endpoint by default - no associated register bit).
  - CFG0: 000 - high level on the selected input (input 1) for bit slice 0
  - CFG1: 000 - high level on the selected input (input 1) for bit slice 1
  - CFG2: 000 - high level on the selected input (input 2) for bit slice 2
  - CFG3: 101 - low level on the selected input (input 2) for bit slice 3
  - CFG4: 101 - low level on the selected input (input 3) for bit slice 4
  - CFG5: 010 - (sticky) falling edge on the selected input (input 6) for bit slice 5
  - CFG6: 000 - high level on the selected input (input 5) for bit slice 6
  - CFG7: 111 - event (any edge, non-sticky) on the selected input (input 7) for bit slice 7
- PMCTRL register (Table 223):

– Bit0: Setting this bit will select pattern matches to generate the pin interrupts in place of the normal pin interrupt mechanism.

  For this example, pin interrupt 0 will be asserted when a match is detected on the first product term (which, in this case, is just a high level on input 1).

  Pin interrupt 2 will be asserted in response to a match on the second product term.

  Pin interrupt 5 will be asserted when there is a match on the third product term.

  Pin interrupt 7 will be asserted on a match on the last term.

– Bit1: Setting this bit will cause the RxEv signal to the ARM CPU to be asserted whenever a match occurs on ANY of the product terms in the expression. Otherwise, the RXEV line will not be used.

– Bit31:24: At any given time, bits 0, 2, 5 and/or 7 may be high if the corresponding product terms are currently matching.

– The remaining bits will always be low.

14.7.3 Pattern match engine edge detect examples

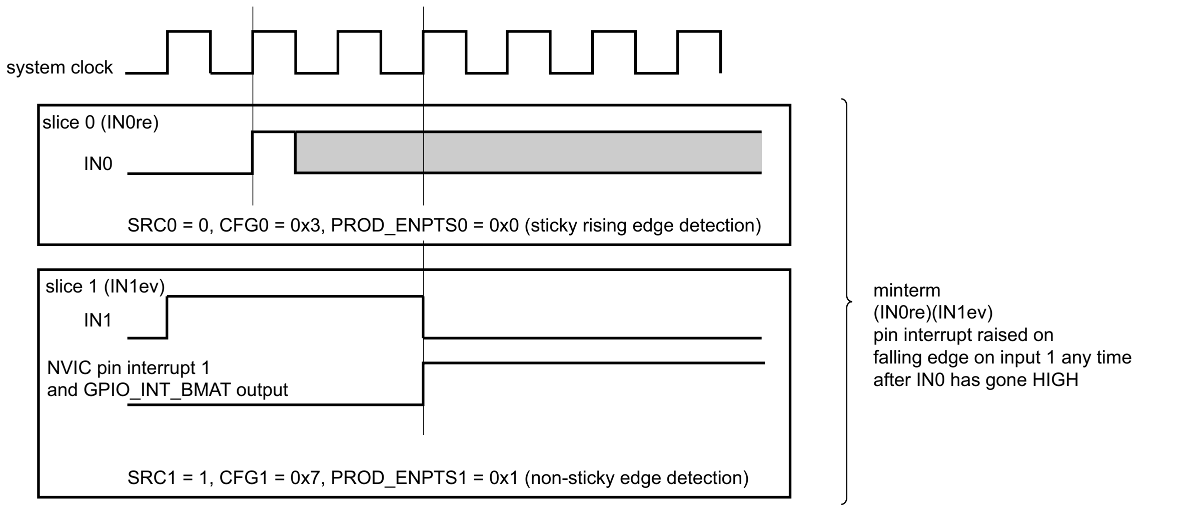

system clock
slice 0 (IN0re)
IN0
SRC0 = 0, CFG0 = 0x3, PROD_ENPTS0 = 0x0 (sticky rising edge detection)
slice 1 (IN1ev)
IN1
NVIC pin interrupt 1
and GPIO_INT_BMAT output
SRC1 = 1, CFG1 = 0x7, PROD_ENPTS1 = 0x1 (non-sticky edge detection)
minterm
(IN0re)(IN1ev)
pin interrupt raised on
falling edge on input 1 any time
after IN0 has gone HIGH

Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

Fig 17. Pattern match engine examples: sticky edge detect

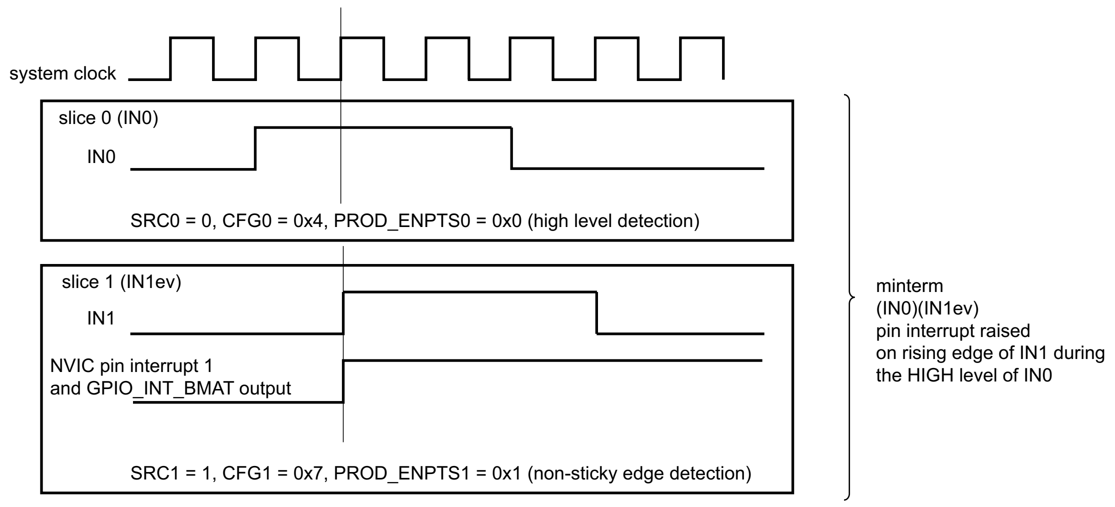

system clock
slice 0 (IN0)
IN0
SRC0 = 0, CFG0 = 0x4, PROD_ENPTS0 = 0x0 (high level detection)
slice 1 (IN1ev)
IN1
NVIC pin interrupt 1
and GPIO_INT_BMAT output
SRC1 = 1, CFG1 = 0x7, PROD_ENPTS1 = 0x1 (non-sticky edge detection)
minterm
(IN0)(IN1ev)
pin interrupt raised
on rising edge of IN1 during
the HIGH level of IN0

Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

Fig 18. Pattern match engine examples: Windowed non-sticky edge detect evaluates as true

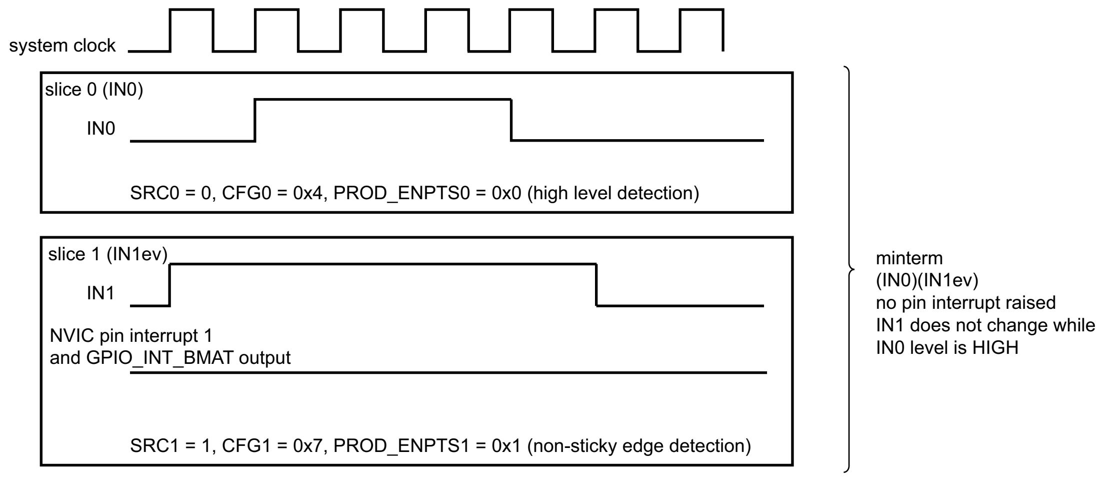

system clock
slice 0 (IN0)
IN0
SRC0 = 0, CFG0 = 0x4, PROD_ENPTS0 = 0x0 (high level detection)
slice 1 (IN1ev)
IN1
NVIC pin interrupt 1
and GPIO_INT_BMAT output
SRC1 = 1, CFG1 = 0x7, PROD_ENPTS1 = 0x1 (non-sticky edge detection)
minterm
(IN0)(IN1ev)
no pin interrupt raised
IN1 does not change while
IN0 level is HIGH

Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

Fig 19. Pattern match engine examples: Windowed non-sticky edge detect evaluates as false

UM11607

Chapter 15: Input multiplexing and DMA trigger multiplexing (INPUT MUX, DMA TRIGMUX)

Rev. 3 — April 2023 **User manual**

15.1 How to read this chapter

DMA input multiplexing is available for all parts.

15.2 Features

- Configures the inputs to the DMA triggers.

15.3 Basic configuration

- In the SYSAHBCLKCTRL register, enable the DMA clock to write to the DMA TRIGMUX registers. See Table 94.

15.4 Pin description

The input multiplexer has no dedicated pins. External pins can be selected as inputs to the DMA triggers. Multiplexer inputs from external pins are assigned through the switch matrix to pins.

15.5 General description

The inputs to the DMA triggers are multiplexed from multiple input sources. The sources can be external pins, interrupts, or output signals of other peripherals.

The input multiplexing makes it possible to design complex event-driven processes without CPU intervention by connecting peripherals like the ADC or the analog comparator.

The DMA can use trigger input multiplexing to sequence DMA transactions without the use of interrupt service routines.

15.5.1 DMA trigger input multiplexing

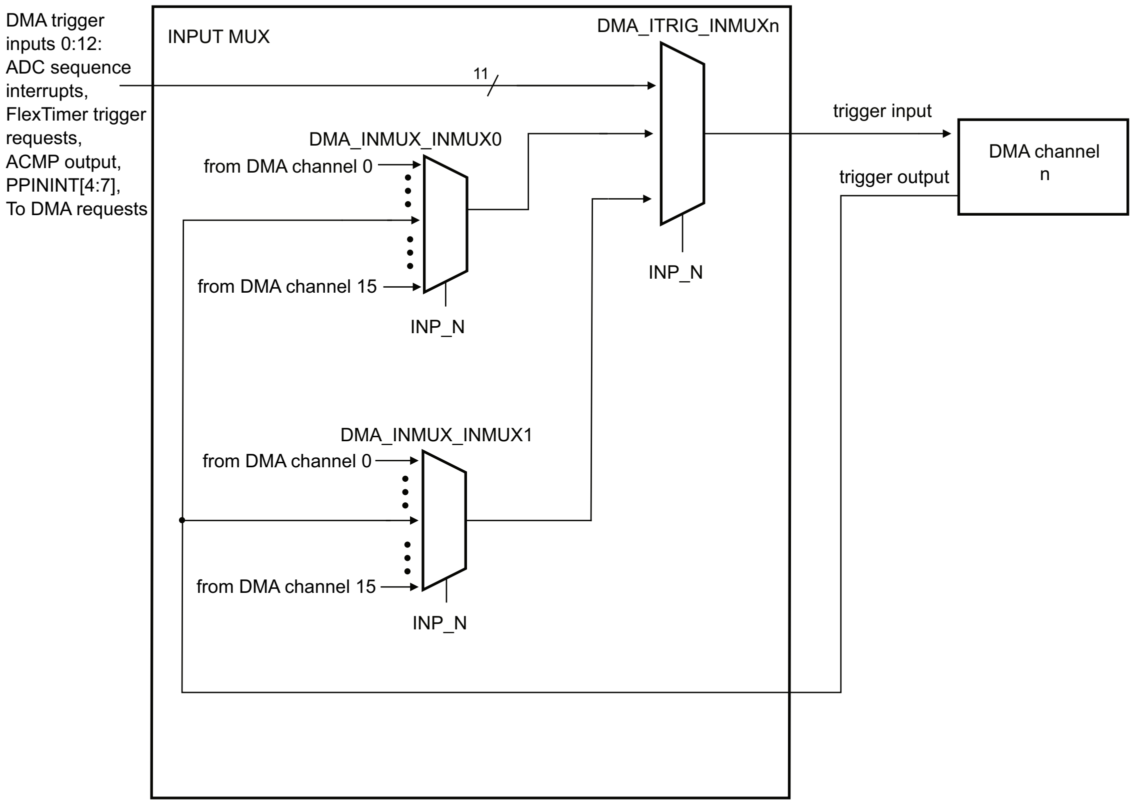


Fig 20. DMA trigger multiplexing

15.6 Register description

All input multiplexer registers reside on word address boundaries. Details of the registers appear in the description of each function.

All address offsets not shown in Table 227 are reserved and should not be written to.

Table 227. Register overview: Input multiplexing (base address 0x4002 C000)

Name	Access	Offset	Description	Reset value	Reference
DMA_INMUX_INMUX0	R/W	0x000	Input mux register for DMA trigger input 11. Selects from 16 DMA trigger outputs.	0x1F	Table 228
DMA_INMUX_INMUX1	R/W	0x004	Input mux register for DMA trigger input 12. Selects from 16 DMA trigger outputs.	0x1F	Table 228
FTM0_INMUX0	-	0x020	Input mux register for FTM0 input 0	0x0F	Table 229
FTM0_INMUX1	-	0x024	Input mux register for FTM0 input 1	0x0F	Table 229
FTM0_INMUX2	-	0x028	Input mux register for FTM0 input 2	0x0F	Table 229
-	-	0x02C	Reserved	0	
DMA_ITRIG_INMUX0	R/W	0x040	Input mux register for trigger inputs 0 to 12 connected to DMA channel 0. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229

Table 227. Register overview: Input multiplexing (base address 0x4002 C000) *…continued*

Name	Access	Offset	Description	Reset value	Reference
DMA_ITRIG_INMUX1	R/W	0x044	Input mux register for trigger inputs 0 to 12 connected to DMA channel 1. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX2	R/W	0x048	Input mux register for trigger inputs 0 to 12 connected to DMA channel 2. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX3	R/W	0x04C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 3. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX4	R/W	0x050	Input mux register for trigger inputs 0 to 12 connected to DMA channel 4. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX5	R/W	0x054	Input mux register for trigger inputs 0 to 12 connected to DMA channel 5. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX6	R/W	0x058	Input mux register for trigger inputs 0 to 12 connected to DMA channel 6. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX7	R/W	0x05C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 7. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX8	R/W	0x060	Input mux register for trigger inputs 0 to 12 connected to DMA channel 8. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX9	R/W	0x064	Input mux register for trigger inputs 0 to 12 connected to DMA channel 9. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX10	R/W	0x068	Input mux register for trigger inputs 0 to 12 connected to DMA channel 10. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX11	R/W	0x06C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 11. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX12	R/W	0x070	Input mux register for trigger inputs 0 to 12 connected to DMA channel 12. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX13	R/W	0x074	Input mux register for trigger inputs 0 to 12 connected to DMA channel 13. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX14	R/W	0x078	Input mux register for trigger inputs 0 to 12 connected to DMA channel 14. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
DMA_ITRIG_INMUX15	R/W	0x07C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 15. Selects from ADC, ACMP, pin interrupts, and DMA requests.	0x0F	Table 229
FTM1_INMUX0	R/W	0x0A0	Input mux register for FTM1 input 0	0x0F	Table 229
FTM1_INMUX1	R/W	0x0A4	Input mux register for FTM1 input 1	0x0F	Table 229
FTM1_INMUX2	R/W	0x0A8	Input mux register for FTM1 input 2	0x0F	Table 229

Table 227. Register overview: Input multiplexing (base address 0x4002 C000) *…continued*

Name	Access	Offset	Description	Reset value	Reference
FTM0_FLT_INMUX0	R/W	0x0C0	Input mux register for FTM0 FAULT input 0	0x0F	Table 229
FTM0_FLT_INMUX1	R/W	0x0C4	Input mux register for FTM0 FAULT input 1	0x0F	Table 229
FTM0_FLT_INMUX2	R/W	0x0C8	Input mux register for FTM0 FAULT input 2	0x0F	Table 229
FTM0_FLT_INMUX3	R/W	0x0CC	Input mux register for FTM0 FAULT input 3	0x0F	Table 229

15.6.1 DMA trigger input mux input registers 0 to 1

This register provides a multiplexer for inputs 11 to 12 of each DMA trigger input mux register DMA_ITRIG_INMUX. These inputs can be selected from the 16 trigger outputs generated by the DMA (one trigger output per channel).

By default, none of the triggers are selected.

Table 228. DMA input trigger input mux input registers 0 to 1 (DMA_INMUX_INMUX[0:1], address 0x4002 C000 (DMA_INMUX_INMUX0) to 0x4002 C004 (DMA_INMUX_INMUX1)) bit description

Bit	Symbol	Description	Reset value
4:0	INP	DMA trigger output number (decimal value) for DMA channel n (n = 0 to 15).	0x1F
31:5	-	Reserved.	-

15.6.2 DMA input trigger input mux registers 0 to 15

With the DMA input trigger input mux registers you can select one trigger input for each of the 16 DMA channels from multiple internal sources.

By default, none of the triggers are selected.

Table 229. DMA input trigger Input mux registers 0 to 15(DMA_ITRIG_INMUX[0:15], address 0x4002 C040 (DMA_ITRIG_INMUX0) to 0x4002 C07C (DMA_ITRIG_INMUX15)) bit description

Bit	Symbol	Value	Description	Reset value
3:0	INP		Trigger input number (decimal value) for DMA channel n (n = 0 to 12). All other values are reserved.	0x0F
		0x0	GPIO_INT4	
		0x1	GPIO_INT5	
		0x2	GPIO_INT6	
		0x3	GPIO_INT7	
		0x4	ADC0_SEQA_IRQ	
		0x5	ADC0_SEQB_IRQ	
		0x6	COMP0_OUT	
		0x7	FTM0_INIT_TRIG ORed with FTM0_EXT_TRIG	
		0x8	FTM1_INIT_TRIG ORed with FTM1_EXT_TRIG	
		0x9	Ored (FTM0_CH0, FTM0_CH1,.., FTM0_CH5)	
		0xA	Ored (FTM1_CH0, FTM1_CH1,.., FTM1_CH3)	
		0xB	DMA trigger mux 0 (DMA_INMUX_INMUX0)	
		0xC	DMA trigger mux 1(DMA_INMUX_INMUX1)	
31:4	-		Reserved.	-

UM11607

Chapter 16: Reduced power modes and power management

Rev. 3 — April 2023 User manual

16.1 How to read this chapter

The LPC86x provides an on-chip API in the boot ROM to optimize power consumption in active and sleep modes.

Read this chapter to configure the reduced power modes deep-sleep mode, power-down mode, and deep power-down mode.

16.2 Features

- Reduced power modes control
- Ultra low-power oscillator control
- Five general purpose backup registers to retain data in deep power-down mode

16.3 Basic configuration

The PMU is always on as long as V_{DD} is present.

If using the WAKEUP or RESET function, disable the hysteresis for the WAKEUP pad or RESET pad in the DPDCTRL register when the supply voltage VDD is below 2.2 V. See Table 235.

If using the WKTCLKIN function, disable the hysteresis for that pin in the DPDCTRL register. See Table 235.

16.3.1 Low power modes in the ARM Cortex-M0+ core

Entering and exiting the low power modes is always controlled by the ARM Cortex-M0+ core. The SCR register is the software interface for controlling the core’s actions when entering a low power mode. The SCR register is located on the ARM private peripheral bus. For details, see Ref. 3.

16.3.1.1 System control register

The System control register (SCR) controls entry to and exit from a low power state. This register is located on the private peripheral bus and is a R/W register with reset value of 0x0000 0000. The SCR register allows to put the ARM core into sleep mode or the entire system in deep-sleep or power-down mode. To set the low power state with SLEEPDEEP = 1 to either deep-sleep or power-down or to enter the deep power-down mode, use the PCON register (Table 233).

Table 230. System control register (SCR, address 0xE000 ED10) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved.	0
1	SLEEPONEXIT	Indicates sleep-on-exit when returning from Handler mode to Thread mode: 0 = do not sleep when returning to Thread mode. 1 = enter sleep, or deep sleep, on return from an ISR to Thread mode. Setting this bit to 1 enables an interrupt driven application to avoid returning to an empty main application.	0
2	SLEEPDEEP	Controls whether the processor uses sleep or deep-sleep as its low power mode: 0 = sleep 1 = deep sleep.	0
3	-	Reserved.	0
4	SEVONPEND	Send Event on Pending bit: 0 = only enabled interrupts or events can wake-up the processor, disabled interrupts are excluded 1 = enabled events and all interrupts, including disabled interrupts, can wake up the processor. 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.	0
31:5	-	Reserved.	0

16.4 Pin description

In deep power-down the WAKEUP pin PIO0_4, RESET pin PIO0_5, and the self-wake-up timer clock input WKTCLKIN on pin PIO0_28 are functional (if enabled). The WAKEUP and the RESET functions can be disabled in the DPDCTRL register to lower the power consumption even more. In this case, enable the self-wake-up timer to provide an internal wake-up signal. See Section 16.6.3 “Deep power-down control register”.

Remark: When entering deep power-down mode, an external pull-up resistor is required on the WAKEUP pin or the RESET pin to hold it HIGH.

16.5 General description

Power on the LPC86x is controlled by the PMU, by the SYSCON block, and the ARM Cortex-M0+ core. The following reduced power modes are supported in order from highest to lowest power consumption:

1. Sleep mode:

   The sleep mode affects the ARM Cortex-M0+ core only. Peripherals and memories are active.

2. Deep sleep and power-down modes:

   The deep sleep and power-down modes affect the core and the entire system with memories and peripherals. Before entering deep-sleep or power-down, you must switch the main clock to the FRO to provide a clock signal that can be shut down cleanly.

   a. In deep sleep mode, the peripherals receive no internal clocks. The flash is in standby mode. The SRAM memory and all peripheral registers as well as the processor maintain their internal states. The WWDT, WKT, and BOD can remain active to wake up the system on an interrupt.

   b. In power-down mode, the peripherals receive no internal clocks. The internal SRAM memory and all peripheral registers as well as the processor maintain their internal states. The flash memory is powered down. The WWDT, WKT, and BOD can remain active to wake up the system on an interrupt.

3. Deep power-down mode:

   For maximal power savings, the entire system is shut down except for the general purpose registers in the PMU and the self-wake-up timer. Only the general purpose registers in the PMU maintain their internal states. The part can wake up on a pulse on either the WAKEUP pin or the RESET pin, or when the self-wake-up timer times out. On wake-up, the part reboots.

Remark: The part is in active mode when it is fully powered and operational after booting.

16.5.1 Wake-up process

If the part receives a wake-up signal in any of the reduced power modes, it wakes up to the active mode.

See these links for related registers and wake-up instructions:

- To configure the system after wake-up: Table 117 “Wake-up configuration register (PDAWAKECFG, address 0x4004 8234) bit description”.
- To use external interrupts for wake-up: Table 114 “Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description” and Table 113 “Pin interrupt select registers (PINTSEL[0:7], address 0x4004 8178 (PINTSEL0) to 0x4004 8194 (PINTSEL7)) bit description”
- To enable external or internal signals to wake up the part from deep-sleep or power-down modes: Table 115 “Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description”
- To configure the USART to wake up the part: Section 18.3.2 “Configure the USART for wake-up”

- For configuring the self-wake-up timer: Section 23.5
- For a list of all wake-up sources: Table 231 "Wake-up sources for reduced power modes".

Table 231. Wake-up sources for reduced power modes

power mode	Wake-up source	Conditions
Sleep	Any interrupt	Enable interrupt in NVIC.
	$\overline{\text{RESET}}$ pin PIO0_5	Enable the reset function in the PINENABLE0 register via switch matrix.
Deep-sleep and power-down	Pin interrupts	Enable pin interrupts in NVIC and STARTERP0 registers.
	BOD interrupt	• Enable interrupt in NVIC and STARTERP1 registers. • Enable interrupt in BODCTRL register. • BOD powered in PDSLEEPCFG register.
	BOD reset	• Enable reset in BODCTRL register. • BOD powered in PDSLEEPCFG register.
	WWDT interrupt	• Enable interrupt in NVIC and STARTERP1 registers. • WWDT running. Enable WWDT in WWDT MOD register and feed. • Enable interrupt in WWDT MOD register. • WDOsc powered in PDSLEEPCFG register.
	WWDT reset	• WWDT running. • Enable reset in WWDT MOD register. • WDOsc powered in PDSLEEPCFG register.
	Self-Wake-up Timer (WKT) time-out	• Enable interrupt in NVIC and STARTERP1 registers. • Enable ultra low-power oscillator in the DPDCTRL register in the PCON block. • Select low-power clock for WKT clock in the WKT CTRL register. • Start the WKT by writing a time-out value to the WKT COUNT register.
	Interrupt from USART/SPI/I2C peripheral	• Enable interrupt in NVIC and STARTERP1 registers. • Enable USART/I2C/SPI interrupts. • Provide an external clock signal to the peripheral. • Configure the USART in synchronous slave mode and I2C and SPI in slave mode.
	Interrupt from peripheral	• Enable interrupt in NVIC and STARTERP1 registers. • Switch FCLK clock source to the WDOsc.
	$\overline{\text{RESET}}$ pin PIO0_5	Enable the reset function in the PINENABLE0 register via switch matrix.
Deep power-down	$\overline{\text{WAKEUP}}$ pin PIO0_4	Enable the WAKEUP function in the DPDCTRL register in the PMU.
	$\overline{\text{RESET}}$ pin PIO0_5	Enable the reset function in the DPDCTRL register in the PMU to allow wake-up in deep power-down mode.
	WKT time-out	• Enable the ultra low-power oscillator in the DPDCTRL register in the PMU. • Enable the ultra low-power oscillator to keep running in deep power-down mode in the DPDCTRL register in the PMU. • Select low-power clock for WKT clock in the WKT CTRL register. • Start WKT by writing a time-out value to the WKT COUNT register.

16.6 Register description

Table 232. Register overview: PMU (base address 0x4002 0000)

Name	Access	Address offset	Description	Reset value	Reference
PCON	R/W	0x000	Power control register	0x0	Table 233
GPREG0	R/W	0x004	General purpose register 0	0x0	Table 234
GPREG1	R/W	0x008	General purpose register 1	0x0	Table 234
GPREG2	R/W	0x00C	General purpose register 2	0x0	Table 234
GPREG3	R/W	0x010	General purpose register 3	0x0	Table 234
DPDCTRL	R/W	0x014	Deep power-down control register. Also includes bits for general purpose storage.	0x0	Table 235

16.6.1 Power control register

The power control register selects whether one of the ARM Cortex-M0+ controlled power-down modes (sleep mode or deep-sleep/power-down mode) or the deep power-down mode is entered and provides the flags for sleep or deep-sleep/power-down modes and deep power-down modes respectively.

Table 233. Power control register (PCON, address 0x4002 0000) bit description

Bit	Symbol	Value	Description	Reset value
2:0	PM		Power mode	000
		0x0	Default. The part is in active or sleep mode.	
		0x1	Deep-sleep mode. ARM WFI will enter deep-sleep mode.	
		0x2	Power-down mode. ARM WFI will enter power-down mode.	
		0x3	Deep power-down mode. ARM WFI will enter deep-power down mode (ARM Cortex-M0+ core powered-down).	
3	NODPD		A 1 in this bit prevents entry to deep power-down mode when 0x3 is written to the PM field above, the SLEEPDEEP bit is set, and a WFI is executed. This bit is cleared only by power-on reset, so writing a one to this bit locks the part in a mode in which deep power-down mode is blocked.	0
7:4	-	-	Reserved. Do not write ones to this bit.	0
8	SLEEPFLAG		Sleep mode flag	0
		0	Active mode. Read: No power-down mode entered. Part is in Active mode. Write: No effect.	
		1	Low power mode. Read: sleep, deep-sleep or power-down mode entered. Write: Writing a 1 clears the SLEEPFLAG bit to 0.	
10:9	-	-	Reserved. Do not write ones to this bit.	0

Table 233. Power control register (PCON, address 0x4002 0000) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
11	DPDFLAG		Deep power-down flag	0
		0	Not deep power-down. Read: deep power-down mode **not** entered. Write: No effect.	0
		1	Deep power-down. Read: deep power-down mode entered. Write: Clear the deep power-down flag.	
31:12	-	-	Reserved. Do not write ones to this bit.	0

16.6.2 General purpose registers 0 to 3

The general purpose registers retain data through the deep power-down mode when power is still applied to the V_{DD} pin but the chip has entered deep power-down mode. Only a cold boot - when all power has been completely removed from the chip - will reset the general purpose registers.

Table 234. General purpose registers 0 to 3 (GPREG[0:3], address 0x4002 0004 (GPREG0) to 0x4002 0010 (GPREG3)) bit description

Bit	Symbol	Description	Reset value
31:0	GPDATA	Data retained during deep power-down mode.	0x0

16.6.3 Deep power-down control register

The deep power-down control register controls the ultra low-power oscillator that can be used by the self-wake-up timer to wake up from Deep power-down mode. In addition, this2 register configures the functionality of the WAKEUP pin (PIO0_4) and the RESET pin (PIO0_5).

The bits in the register not used for deep power-down control (bits 31:4) can be used for storing additional data which are retained in deep power-down mode in the same way as registers GPREG0 to GPREG3.

Remark: If there is a possibility that the external voltage applied on pin V_{DD} drops below 2.2 V during deep power-down, the hysteresis of the WAKEUP or the RESET input pin has to be disabled in this register before entering deep power-down mode in order for the chip to wake up.

Remark: Enabling the ultra low-power oscillator in deep power-down mode increases the power consumption. Only enable this oscillator if you need the self-wake-up timer to wake up the part from deep power-down mode. You may need the self-wake-up timer if the wake-up pin is used for other purposes and the wake-up function is not available.

Table 235. Deep power down control register (DPDCTRL, address 0x4002 0014) bit description

Bit	Symbol	Value	Description	Reset value
0	WAKEUPHYS		WAKEUP pin hysteresis enable	0
		0	Disabled. Hysteresis for WAKEUP pin disabled.	
		1	Enabled. Hysteresis for WAKEUP pin enabled.	

Table 235. Deep power down control register (DPDCTRL, address 0x4002 0014) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
1	WAKEPAD_ DISABLE		WAKEUP pin disable. Setting this bit disables the wake-up pin, so it can be used for other purposes. **Remark:** Setting this bit is not necessary if deep power-down mode is not used.	0
		0	Enabled. The wake-up function is enabled on pin PIO0_4.	
		1	Disabled. Setting this bit disables the wake-up function on pin PIO0_4.	
2	ULPOSCEN		Enable the ultra low-power oscillator for use with the 10 kHz self-wake-up timer clock. You must set this bit if the CLKSEL bit in the self-wake-up timer CTRL bit is set. Do not enable the ultra low-power oscillator if the self-wake-up timer is clocked by the divided FRO or the external clock input.	0
		0	Disabled.	
		1	Enabled.	
3	ULPOSCDPDEN		Enable the ultra low-power oscillator in deep power-down mode. Setting this bit causes the ultra low-power oscillator to remain running during deep power-down mode provided that bit 2 in this register is set as well. You must set this bit for the self-wake-up timer to be able to wake up the part from deep power-down mode. **Remark:** Do not set this bit unless you use the self-wake-up timer with the ultra low-power oscillator clock source to wake up from deep power-down mode.	0
		0	Disabled.	
		1	Enabled.	
4	WAKEUPCLKHYS		External clock input for the self-wake-up timer WKTCLKIN hysteresis enable.	0
		0	Disabled. Hysteresis for WAKEUP clock pin disabled.	
		1	Enabled. Hysteresis for WAKEUP clock pin enabled.	
5	WAKECLKPAD_ DISABLE		Disable the external clock input for the self-wake-up timer. Setting this bit enables the self-wake-up timer clock pin WKTCLKLIN. To minimize power consumption, especially in deep power-down mode, disable this clock input when not using the external clock option for the self-wake-up timer.	0
		0	Disabled. Setting this bit disables external clock input on pin PIO0_28.	
		1	Enabled. The external clock input for the self-wake-up timer is enabled on pin PIO0_28.	
6	RESETHYS		RESET pin hysteresis enable.	0
		0	Disabled. Hysteresis for RESET pin disabled.	
		1	Enabled. Hysteresis for RESET pin enabled.	
7	RESET_ DISABLE		RESET pin disable. Setting this bit disables the reset wake-up function, so the pin can be used for other purposes. **Remark:** Setting this bit is not necessary if deep power-down mode is not used.	0
		0	Enabled. The reset wake-up function is enabled on pin PIO0_5.	
		1	Disabled. Setting this bit disables the wake-up function on pin PIO0_5.	
31:8	GPDATA		Data retained during deep power-down mode.	0x0

Remark: Do not set bit 1 and bit 7 if you intend to use a pin to wake up the part from deep power-down mode. You can only disable both wake-up pins if the self-wake-up timer is enabled and configured.

16.7 Functional description

16.7.1 Power management

The part supports a variety of power control features. In Active mode, when the chip is running, power and clocks to selected peripherals can be optimized for power consumption. In addition, there are four special modes of processor power reduction with different peripherals running: sleep mode, deep-sleep mode, power-down mode, and deep power-down mode.

Table 236. Peripheral configuration in reduced power modes

Peripheral	Sleep mode	Deep-sleep mode	Power-down mode	Deep power-down mode
FRO	software configurable	on	off	off
FRO output	software configurable	off	off	off
Flash	software configurable	on	off	off
BOD	software configurable	software configurable	software configurable	off
PLL	software configurable	off	off	off
SysOsc	software configurable	off	off	off
WDosc/WWDT	software configurable	software configurable	software configurable	off
Digital peripherals	software configurable	off	off	off
WKT/ultra low-power oscillator	software configurable	software configurable	software configurable	software configurable
ADC	software configurable	off	off	off
Comparator	software configurable	off	off	off

Remark: The Debug mode is not supported in sleep, deep-sleep, power-down, or deep power-down modes.

16.7.2 Reduced power modes and WWDT lock features

The WWDT lock feature influences the power consumption in any of the power modes because locking the WWDT clock source forces the watchdog oscillator to be on independently of the deep-sleep and power-down mode software configuration through the PDSLEEPCFG register. For details see Section 22.5.3 “Using the WWDT lock features”.

16.7.3 Active mode

In Active mode, the ARM Cortex-M0+ core, memories, and peripherals are clocked by the system clock or main clock.

The chip is in Active mode after reset and the default power configuration is determined by the reset values of the PDRUNCFG and SYSAHBCLKCTRL registers. The power configuration can be changed during run time.

16.7.3.1 Power configuration in Active mode

Power consumption in Active mode is determined by the following configuration choices:

- The SYSAHBCLKCTRL register controls which memories and peripherals are running (Table 94).
- The power to various analog blocks (PLL, oscillators, the BOD circuit, the ADC block, the comparator block, and the flash block) can be controlled at any time individually through the PDRUNCFG register (Table 118 "Power configuration register (PDRUNCFG, address 0x4004 8238) bit description").
- The clock source for the system clock can be selected from the FRO (default), the system oscillator, external clock, the watchdog oscillator, or the divided FRO (see Figure 7 and related registers).
- The system clock frequency can be selected by the SYSPLLCTRL (Table 73) and the SYSAHBCLKDIV register (Table 86).
- The USART, ADC, and CLKOUT use individual peripheral clocks with their own clock dividers. The peripheral clocks can be shut down through the corresponding clock divider registers.

16.7.4 Sleep mode

In sleep mode, the system clock to the ARM Cortex-M0+ core is stopped and execution of instructions is suspended until either a reset or an interrupt occurs.

Peripheral functions, if selected to be clocked in the SYSAHBCLKCTRL register, continue operation during sleep mode and may generate interrupts to cause the processor to resume execution. Sleep mode eliminates dynamic power used by the processor itself, memory systems and related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins retain static.

16.7.4.1 Power configuration in sleep mode

Power consumption in sleep mode is configured by the same settings as in Active mode:

- The clock remains running.
- The system clock frequency remains the same as in Active mode, but the processor is not clocked.
- Analog and digital peripherals are selected as in Active mode.

16.7.4.2 Programming sleep mode

The following steps must be performed to enter sleep mode:

1. The PM bits in the PCON register must be set to the default value 0x0.
2. The SLEEPDEEP bit in the ARM Cortex-M0+ SCR register must be set to zero (Table 230).
3. Use the ARM Cortex-M0+ Wait-For-Interrupt (WFI) instruction.

16.7.4.3 Wake-up from sleep mode

Sleep mode is exited automatically when an interrupt enabled by the NVIC arrives at the processor or a reset occurs. After wake-up due to an interrupt, the microcontroller returns to its original power configuration defined by the contents of the PDRUNCFG and the SYSAHBCLKDIV registers. If a reset occurs, the microcontroller enters the default configuration in Active mode.

16.7.5 Deep-sleep mode

In deep-sleep mode, the system clock to the processor is disabled as in sleep mode. All analog blocks are powered down, except for the BOD circuit and the low power oscillator, which can be selected or deselected during deep-sleep mode in the PDSLEEPCFG register. The main clock, and therefore all peripheral clocks, are disabled except for the clock to the watchdog timer if the low power oscillator is selected. The FRO is running, but its output is disabled. The flash is in standby mode.

Deep-sleep mode eliminates all power used by analog peripherals and all dynamic power used by the processor itself, memory systems and related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static.

16.7.5.1 Power configuration in deep-sleep mode

Power consumption in deep-sleep mode is determined by the deep-sleep power configuration setting in the PDSLEEPCFG (Table 116) register:

- The watchdog oscillator can be left running in deep-sleep mode if required for the WWDT.
- The BOD circuit can be left running in deep-sleep mode if required by the application.

16.7.5.2 Programming deep-sleep mode

The following steps must be performed to enter deep-sleep mode:

1. The PM bits in the PCON register must be set to 0x1 (Table 233).
2. Select the power configuration in deep-sleep mode in the PDSLEEPCFG (Table 116) register.
3. Select the power configuration after wake-up in the PDAWAKECFG (Table 117) register.
4. If any of the available wake-up interrupts are needed for wake-up, enable the interrupts in the interrupt wake-up registers (Table 114, Table 115) and in the NVIC.
5. Select the FRO as the main clock. See Table 83.
6. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register (Table 230).
7. Use the ARM WFI instruction.

16.7.5.3 Wake-up from deep-sleep mode

The microcontroller can wake up from deep-sleep mode in the following ways:

- Signal on one of the eight pin interrupts selected in Table 113. Each pin interrupt must also be enabled in the STARTERP0 register (Table 114) and in the NVIC.
- BOD signal, if the BOD is enabled in the PDSLEEPCFG register:

- BOD interrupt using the deep-sleep interrupt wake-up register 1 (Table 115). The BOD interrupt must be enabled in the NVIC. The BOD interrupt must be selected in the BODCTRL register.
- Reset from the BOD circuit. In this case, the BOD circuit must be enabled in the PDSLEEPCFG register, and the BOD reset must be enabled in the BODCTRL register (Table 109).

- WWDT signal, if the ultra low power oscillator is enabled in the PDSLEEPCFG register:
  - WWDT interrupt using the interrupt wake-up register 1 (Table 115). The WWDT interrupt must be enabled in the NVIC. The WWDT interrupt must be set in the WWDT MOD register, and the WWDT must be enabled in the SYSAHBCLKCTRL register.
  - Reset from the watchdog timer. The WWDT reset must be set in the WWDT MOD register. In this case, the ultra low power oscillator must be running in deep-sleep mode (see PDSLEEPCFG register), and the WDT must be enabled in the SYSAHBCLKCTRL register.
- Via any of the USART blocks if the USART is configured in synchronous mode. See Section 18.3.2 “Configure the USART for wake-up”.
- Via the I2C. See Section 20.3.3.
- Via any of the SPI blocks. See Section 19.3.1.

Remark: If the BOD is enabled in active mode and the user needs to disable the BOD in deep-sleep mode, disable the BOD reset (bit 4 in the BODCTRL register) before entering power-down mode. After wake-up, enable the BOD reset (bit 4 in the BODCTRL register).

16.7.6 Power-down mode

In power-down mode, the system clock to the processor is disabled as in sleep mode. All analog blocks are powered down, except for the BOD circuit and the ultra low-power oscillator, which must be selected or deselected during power-down mode in the PDSLEEPCFG register. The main clock and therefore all peripheral clocks are disabled except for the clock to the watchdog timer if the ultra low-power oscillator is selected. The FRO itself and the flash are powered down, decreasing power consumption compared to deep-sleep mode.

Power-down mode eliminates all power used by analog peripherals and all dynamic power used by the processor itself, memory systems and related controllers, and internal buses. The processor state and registers, peripheral registers, and internal SRAM values are maintained, and the logic levels of the pins remain static. Wake-up times are longer compared to the deep-sleep mode.

16.7.6.1 Power configuration in power-down mode

Power consumption in power-down mode can be configured by the power configuration setting in the PDSLEEPCFG (Table 116) register in the same way as for deep-sleep mode (see Section 16.7.5.1):

- The ultra low-power oscillator can be left running in power-down mode if required for the WWDT.
- The BOD circuit can be left running in power-down mode if required by the application.

16.7.6.2 Programming power-down mode

The following steps must be performed to enter power-down mode:

1. The PM bits in the PCON register must be set to 0x2 (Table 233).
2. Select the power configuration in power-down mode in the PDSLEEPCFG (Table 116) register.
3. Select the power configuration after wake-up in the PDAWAKECFG (Table 117) register.
4. If any of the available wake-up interrupts are used for wake-up, enable the interrupts in the interrupt wake-up registers (Table 114, Table 115) and in the NVIC.
5. Select the FRO as the main clock. See Table 83.
6. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register (Table 230).
7. Use the ARM WFI instruction.

16.7.6.3 Wake-up from power-down mode

The microcontroller can wake up from power-down mode in the same way as from deep-sleep mode:

- Signal on one of the eight pin interrupts selected in Table 113. Each pin interrupt must also be enabled in the STARTERP0 register (Table 114) and in the NVIC.
- BOD signal, if the BOD is enabled in the PDSLEEPCFG register:
  - BOD interrupt using the interrupt wake-up register 1 (Table 115). The BOD interrupt must be enabled in the NVIC. The BOD interrupt must be selected in the BODCTRL register.
  - Reset from the BOD circuit. In this case, the BOD reset must be enabled in the BODCTRL register (Table 109).
- WWDT signal, if the ultra low-power oscillator is enabled in the PDSLEEPCFG register:
  - WWDT interrupt using the interrupt wake-up register 1 (Table 115). The WWDT interrupt must be enabled in the NVIC. The WWDT interrupt must be set in the WWDT MOD register.
  - Reset from the watchdog timer. The WWDT reset must be set in the WWDT MOD register.
  - Via any of the USART blocks. See Section 18.3.2 “Configure the USART for wake-up”.
  - Via the I2C. See Section 20.3.3.
  - Via any of the SPI blocks. See Section 19.3.1.

  Remark: If the BOD is enabled in active mode and the user needs to disable the BOD in power-down mode, disable the BOD reset (bit 4 in the BODCTRL register) before entering power-down mode. After wake-up, enable the BOD reset (bit 4 in the BODCTRL register).

16.7.7 Deep power-down mode

In deep power-down mode, power and clocks are shut off to the entire chip with the exception of the WAKEUP pin, RESET pin, and the self-wake-up timer.

During deep power-down mode, the contents of the SRAM and registers are not retained except for a small amount of data which can be stored in the general purpose registers of the PMU block.

All functional pins are tri-stated in deep power-down mode except for the WAKEUP pin and the RESET pin.

Remark: Setting bit 3 in the PCON register (Table 233) prevents the part from entering deep-power down mode.

16.7.7.1 Power configuration in deep power-down mode

Deep power-down mode has no configuration options. All clocks, the core, and all peripherals are powered down. Only the WAKEUP pin, RESET pin, and the self-wake-up timer are powered.

16.7.7.2 Programming deep power-down mode using the WAKEUP or RESET pin

The following steps must be performed to enter deep power-down mode when using the WAKEUP pin or the RESET pin for waking up:

1. Pull the WAKEUP pin or the RESET pin externally HIGH depending on which WAKEUP source is used.
2. Ensure that bit 3 in the PCON register (Table 233) is cleared.
3. Write 0x3 to the PM bits in the PCON register (see Table 233).
4. Store data to be retained in the general purpose registers (Section 16.6.2).
5. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register (Table 230).
6. Use the ARM WFI instruction.

16.7.7.3 Wake-up from deep power-down mode using the WAKEUP pin or RESET pin

Pulling the WAKEUP pin or RESET pin LOW wakes up the LPC86x from deep power-down, and the part goes through the entire reset process.

1. On the WAKEUP pin or RESET pin, transition from HIGH to LOW.
   – The PMU will turn on the on-chip voltage regulator. When the core voltage reaches the power-on-reset (POR) trip point, a system reset will be triggered and the chip re-boots.
   – All registers except the DPDCTRL and GPREG0 to GPREG3 registers and PCON will be in their reset state.
2. Once the chip has booted, read the deep power-down flag in the PCON register (Table 233) to verify that the reset was caused by a wake-up event from deep power-down and was not a cold reset.
3. Clear the deep power-down flag in the PCON register (Table 233).
4. (Optional) Read the stored data in the general purpose registers (Section 16.6.2).
5. Set up the PMU for the next deep power-down cycle.

16.7.7.4 Programming deep power-down mode using the self-wake-up timer:

The following steps must be performed to enter deep power-down mode when using the self-wake-up timer for waking up:

1. Enable the ultra low-power oscillator to run in deep power-down mode by setting bits 2 and 3 in the DPDCTRL register to 1 (see Table 235)
2. Ensure that bit 3 in the PCON register (Table 233) is cleared.
3. Write 0x3 to the PM bits in the PCON register (see Table 233).
4. Store data to be retained in the general purpose registers (Section 16.6.2).
5. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register.
6. Start the self-wake-up timer by writing a value to the WKT COUNT register (Table 375).
7. Use the ARM WFI instruction.

16.7.7.5 Wake-up from deep power-down mode using the self-wake-up timer:

The part goes through the entire reset process when the self-wake-up timer times out:

1. When the WKT count reaches 0, the following happens:
   – The PMU will turn on the on-chip voltage regulator. When the core voltage reaches the power-on-reset (POR) trip point, a system reset will be triggered and the chip re-boots.
   – All registers except the DPDCTRL and GPREG0 to GPREG3 registers and PCON are in their reset state.
2. Once the chip has booted, read the deep power-down flag in the PCON register (Table 233) to verify that the reset was caused by a wake-up event from deep power-down and was not a cold reset.
3. Clear the deep power-down flag in the PCON register (Table 233).
4. (Optional) Read the stored data in the general purpose registers (Section 16.6.2).
5. Set up the PMU for the next deep power-down cycle.

Chapter 17: DMA controller

17.1 How to read this chapter

The DMA controller is available on all parts.

17.2 Features

- 16 channels supported with 14 channels connected to peripheral request inputs and outputs of the USART, SPI, I^{2}C, and I^{3}C.
- DMA operations can be triggered by on- or off-chip events. Each DMA channel can select one trigger input from 13 sources.
- Priority is user selectable for each channel.
- Continuous priority arbitration.
- Address cache with four entries.
- Efficient use of data bus.
- Supports single transfers up to 1,024 words.
- Address increment options allow packing and/or unpacking data.

17.3 Basic configuration

Configure the DMA as follows:

- Use the SYSAHBCLKCTRL register (Table 94) to enable the clock to the DMA registers interface.
- The DMA interrupt is connected to slot #20 in the NVIC.
- Each DMA channel has one DMA request line associated and can also select one of 13 input triggers through the input multiplexer registers DMA_ITRIG_INMUX[0:24].
- Trigger outputs are connected to DMA_INMUX_INMUX[0:1] as inputs to DMA triggers.

For details on the trigger input and output multiplexing, see Section 15.5.1 “DMA trigger input multiplexing”.

17.3.1 Hardware triggers

Each DMA channel can use one trigger that is independent of the request input for this channel. The trigger input is selected in the DMA_ITRIG_INMUX registers. There are 13 possible internal trigger sources for each channel with each trigger signal issued by the output of a peripheral. In addition, the DMA trigger output can be routed to the trigger input of another channel through the trigger input multiplexing. See Section 15.5.1 “DMA trigger input multiplexing”.

See Table 237 for the connection of input multiplexers to DMA channels.

See Table 230 for a list of possible trigger input sources.

17.3.2 Trigger outputs

Each channel of the DMA controller provides a trigger output. This allows the possibility of using the trigger outputs as a trigger source to a different channel in order to support complex transfers on selected peripherals. This kind of transfer can, for example, use more than one peripheral DMA request. An example use would be to input data to a holding buffer from one peripheral, and then output the data to another peripheral, with both transfers being paced by the appropriate peripheral DMA request. This kind of an operation is called “chained operation” or “channel chaining”.

17.3.3 DMA requests

DMA requests are directly connected to the peripherals. Each channel supports one DMA request line and one trigger input which is multiplexed to many possible input sources.

For each trigger multiplexer DMA_ITRIG_INMUXn, the following sources are supported:

- ADC sequence A interrupt ADC_SEQA_IRQ
- ADC sequence B interrupt ADC_SEQB_IRQ
- ACMP_O comparator output
- GPIO pin interrupt 4 (PININT4)
- GPIO pin interrupt 5 (PININT5)
- GPIO pin interrupt 6 (PININT6)
- GPIO pin interrupt 7 (PININT7)
- FTM0_INIT_TRIG ORed with FTM0_EXT_TRIG
- FTM1_INIT_TRIG ORed with FTM1_EXT_TRIG
- Ored (FTM0_CH0, FTM0_CH1,.., FTM0_CH5)
- Ored (FTM1_CH0, FTM1_CH1,.., FTM1_CH3)
- Two choices of DMA output triggers, SDMA_TRIGOUT_A and SDMA_TRIGOUT_B

Table 237. DMA requests

DMA channel #	Request input	DMA trigger multiplexer
0	UART0_RX_DMA	DMA_ITRIG_INMUX0
1	UART0_TX_DMA	DMA_ITRIG_INMUX1
2	UART1_RX_DMA	DMA_ITRIG_INMUX2
3	UART1_TX_DMA	DMA_ITRIG_INMUX3
4	UART2_RX_DMA	DMA_ITRIG_INMUX4
5	UART2_TX_DMA	DMA_ITRIG_INMUX5
6	LPSPI0_RX_DMA	DMA_ITRIG_INMUX6
7	LPSPI0_TX_DMA	DMA_ITRIG_INMUX7
8	LPSPI1_RX_DMA	DMA_ITRIG_INMUX8
9	LPSPI1_TX_DMA	DMA_ITRIG_INMUX9
10	I2C0_RX_DMA	DMA_ITRIG_INMUX10
11	I2C0_TX_DMA	DMA_ITRIG_INMUX11
12	I3C0_RX_DMA	DMA_ITRIG_INMUX12

Table 237. DMA requests

DMA channel #	Request input	DMA trigger multiplexer
13	I3C0_TX_DMA	DMA_ITRIG_INMUX13
14	Reserved	DMA_ITRIG_INMUX14
15	Reserved	DMA_ITRIG_INMUX15

17.3.4 DMA in sleep mode

The DMA can operate and access all SRAM blocks in sleep mode.

17.4 Pin description

The DMA controller has no configurable pins.

17.5 General description

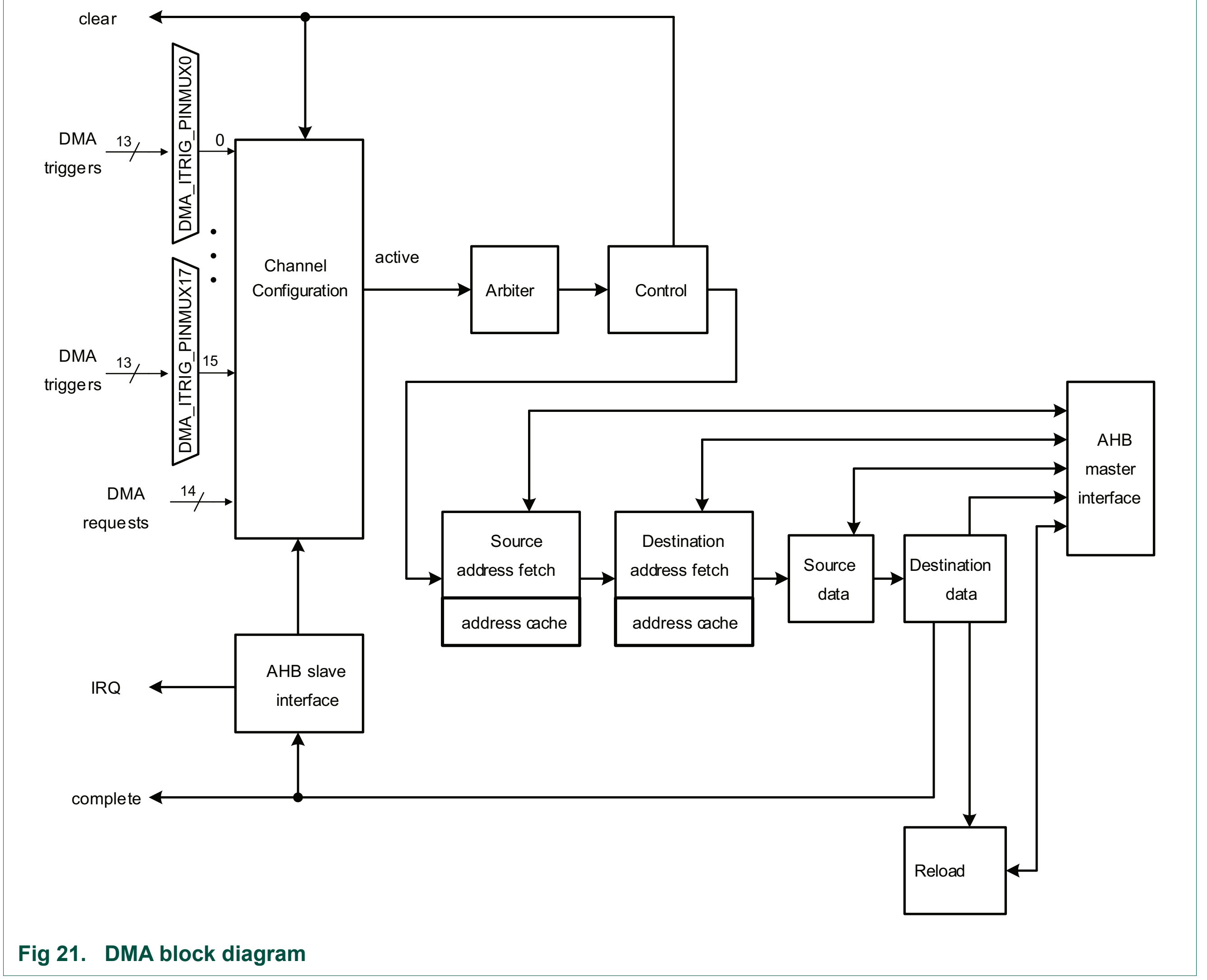


Fig 21. DMA block diagram

17.5.1 DMA requests and triggers

An operation on a DMA channel can be initiated by either a DMA request or a trigger event. DMA requests come from peripherals and specifically indicate when a peripheral either needs input data to be read from it, or that output data may be sent to it. DMA requests are created by the UART, SPI, I2C, and I3C.

A trigger initiates a DMA operation and can be a signal from an unrelated peripheral. Peripherals that generate triggers are the ADC and the analog comparator. In addition, the DMA triggers also create a trigger output that can trigger DMA transactions on another channel. Triggers can be used to send a character or a string to a UART or other serial output at a fixed time interval or when an event occurs.

A DMA channel using a trigger can respond by moving data from any memory address to any other memory address. This can include fixed peripheral data registers, or incrementing through RAM buffers. The amount of data moved by a single trigger event can range from a single transfer to many transfers. A transfer that is started by a trigger can still be paced using the channel's DMA request. This allows sending a string to a serial peripheral, for instance, without overrunning the peripheral's transmit buffer.

Each trigger input to the DMA has a corresponding output that can be used as a trigger input to another channel. The trigger outputs appear in the trigger source list for each channel and can be selected through the DMA_INMUX registers as inputs to other channels.

17.5.2 DMA Modes

The DMA controller doesn't really have separate operating modes, but there are ways of using the DMA controller that have commonly used terminology in the industry.

Once the DMA controller is set up for operation, using any specific DMA channel requires initializing the registers associated with that channel (see Table 237), and supplying at least the channel descriptor, which is located somewhere in memory, typically in on-chip SRAM (see Section 17.6.3). The channel descriptor is shown in Table 238.

Table 238. Channel descriptor

Offset	Description
+ 0x0	Reserved
+ 0x4	Source data end address
+ 0x8	Destination end address
+ 0xC	Link to next descriptor

The source and destination end addresses, as well as the link to the next descriptor are just memory addresses that can point to any valid address on the device. The starting address for both source and destination data is the specified end address minus the transfer length (XFERCOUNT * the address increment as defined by SRCINC and DSTINC). The link to the next descriptor is used only if it is a linked transfer.

After the channel has had a sufficient number of DMA requests and/or triggers, depending on its configuration, the initial descriptor will be exhausted. At that point, if the transfer configuration directs it, the channel descriptor will be reloaded with data from memory pointed to by the "Link to next descriptor" entry of the initial channel descriptor. Descriptors loaded in this manner look slightly different the channel descriptor, as shown in Table 239. The difference is that a new transfer configuration is specified in the reload descriptor instead of being written to the XFERCFG register for that channel.

This process repeats as each descriptor is exhausted as long as reload is selected in the transfer configuration for each new descriptor.

Table 239. Reload descriptors

Offset	Description
+ 0x0	Transfer configuration.
+ 0x4	Source end address. This points to the address of the last entry of the source address range if the address is incremented. The address to be used in the transfer is calculated from the end address, data width, and transfer size.
+ 0x8	Destination end address. This points to the address of the last entry of the destination address range if the address is incremented. The address to be used in the transfer is calculated from the end address, data width, and transfer size.
+ 0xC	Link to next descriptor. If used, this address must be aligned to a multiple of 16 bytes (i.e., the size of a descriptor).

17.5.3 Single buffer

This generally applies to memory to memory moves, and peripheral DMA that occurs only occasionally and is set up for each transfer. For this kind of operation, only the initial channel descriptor shown in Table 240 is needed.

Table 240. Channel descriptor for a single transfer

Offset	Description
+ 0x0	Reserved
+ 0x4	Source data end address
+ 0x8	Destination data end address
+ 0xC	(not used)

This case is identified by the Reload bit in the XFERCFG register = 0. When the DMA channel receives a DMA request or trigger (depending on how it is configured), it performs one or more transfers as configured, then stops. Once the channel descriptor is exhausted, additional DMA requests or triggers will have no effect until the channel configuration is updated by software.

17.5.4 Ping-Pong

Ping-pong is a special case of a linked transfer. It is described separately because it is typically used more frequently than more complicated versions of linked transfers.

A ping-pong transfer uses two buffers alternately. At any one time, one buffer is being loaded or unloaded by DMA operations. The other buffer has the opposite operation being handled by software, readying the buffer for use when the buffer currently being used by the DMA controller is full or empty. Table 241 shows an example of descriptors for ping-pong from a peripheral to two buffers in memory.

Table 241. Example descriptors for ping-pong operation: peripheral to buffer

Channel Descriptor			Descriptor B			Descriptor A	
+ 0x0	(not used)		+ 0x0	Buffer B transfer configuration		+ 0x0	Buffer A transfer configuration
+ 0x4	Peripheral data end address		+ 0x4	Peripheral data end address		+ 0x4	Peripheral data end address
+ 0x8	Buffer A memory end address		+ 0x8	Buffer B memory end address		+ 0x8	Buffer A memory end address
+ 0xC	Address of descriptor B		+ 0xC	Address of descriptor A		+ 0xC	Address of descriptor B

In this example, the channel descriptor is used first, with a first buffer in memory called buffer A. The configuration of the DMA channel must have been set to indicate a reload. Similarly, both descriptor A and descriptor B must also specify reload. When the channel descriptor is exhausted, descriptor B is loaded using the link to descriptor B, and a transfer interrupt informs the CPU that buffer A is available.

Descriptor B is then used until it is also exhausted, when descriptor A is loaded using the link to descriptor A contained in descriptor B. Then a transfer interrupt informs the CPU that buffer B is available for processing. The process repeats when descriptor A is exhausted, alternately using each of the 2 memory buffers.

17.5.5 Linked transfers (linked list)

A linked transfer can use any number of descriptors to define a complicated transfer. This can be configured such that a single transfer, a portion of a transfer, one whole descriptor, or an entire structure of links can be initiated by a single DMA request or trigger.

An example of a linked transfer could start out like the example for a ping-pong transfer (Table 241). The difference would be that descriptor B would not link back to descriptor A, but would continue on to another different descriptor. This could continue as long as desired, and can be ended anywhere, or linked back to any point to repeat a sequence of descriptors. Of course, any descriptor not currently in use can be altered by software as well.

17.5.6 Address alignment for data transfers

Transfers of 16 bit width require an address alignment to a multiple of 2 bytes. Transfers of 32 bit width require an address alignment to a multiple of 4 bytes. Transfers of 8 bit width can be at any address.

17.5.7 Channel chaining

Channel chaining is a feature, which allows completion of a DMA transfer on channel x to trigger a DMA transfer on channel y. This feature can for example be used to have DMA channel x reading n bytes from UART to memory, and then have DMA channel y transferring the received bytes to the CRC engine, without any action required from the ARM core.

To use channel chaining, first configure DMA channels x and y as if no channel chaining would be used. Then:

- For channel x:

  - If channel x is configured to auto reload the descriptor on exhausting of the descriptor (bit RELOAD in the transfer configuration of the descriptor is set), then enable 'clear trigger on descriptor exhausted' by setting bit CLRTRIG in the channel's transfer configuration in the descriptor.
- For channel y:
  - Configure the input trigger input mux register (DMA_ITRIG_INMUX[0:24]) for channel y to use any of the available DMA trigger muxes (DMA trigger mux 0/1).
  - Configure the chosen DMA trigger mux to select DMA channel x.
  - Enable hardware triggering by setting bit HWTRIGEN in the channel configuration register.
  - Set the trigger type to edge sensitive by clearing bit TRIGTYPE in the channel configuration register.
  - Configure the trigger edge to falling edge by clearing bit TRIGPOL in the channel configuration register.

Note that after completion of channel x the descriptor may be reloaded (if configured so), but remains un-triggered. To configure the chain to auto-trigger itself, setup channels x and y for channel chaining as described above. In addition to that:

- A ping-pong configuration for both channel x and y is recommended, so that data currently moved by channel y is not altered by channel x.
- For channel x:
  - Configure the input trigger input mux register (DMA_ITRIG_INMUX[0:24]) for channel y to use the same DMA trigger mux as chosen for channel y.
  - Enable hardware triggering by setting bit HWTRIGEN in the channel configuration register.
  - Set the trigger type to edge sensitive by clearing bit TRIGTYPE in the channel configuration register.
  - Configure the trigger edge to falling edge by clearing bit TRIGPOL in the channel configuration register.

17.6 Register description

The DMA registers are grouped into DMA control, interrupt and status registers and DMA channel registers. DMA transfers are controlled by a set of three registers per channel, the CFG[0:24], CTRLSTAT[0:24], and XFERCFG[0:24] registers.

The reset value reflects the data stored in used bits only. It does not include the content of reserved bits.

Table 242. Register overview: DMA controller (base address 0x5000 5000)

Name	Access	Address offset	Description	Reset Value	Reference
Global control and status registers					
CTRL	R/W	0x000	DMA control.	0	Table 243
INTSTAT	RO	0x004	Interrupt status.	0	Table 244
SRAMBASE	R/W	0x008	SRAM address of the channel configuration table.	0	Table 245

Table 242. Register overview: DMA controller (base address 0x5000 5000)

Name	Access	Address offset	Description	Reset Value	Reference
Shared registers					
ENABLESET0	RO/W1	0x020	Channel Enable read and Set for all DMA channels.	0	Table 247
ENABLECLR0	W1	0x028	Channel Enable Clear for all DMA channels.	NA	Table 248
ACTIVE0	RO	0x030	Channel Active status for all DMA channels.	0	Table 249
BUSY0	RO	0x038	Channel Busy status for all DMA channels.	0	Table 250
ERRINT0	RO/W1	0x040	Error Interrupt status for all DMA channels.	0	Table 251
INTENSET0	RO/W1	0x048	Interrupt Enable read and Set for all DMA channels.	0	Table 252
INTENCLR0	W1	0x050	Interrupt Enable Clear for all DMA channels.	NA	Table 253
INTA0	RO/W1	0x058	Interrupt A status for all DMA channels.	0	Table 254
INTB0	RO/W1	0x060	Interrupt B status for all DMA channels.	0	Table 255
SETVALID0	W1	0x068	Set ValidPending control bits for all DMA channels.	NA	Table 256
SETTRIG0	W1	0x070	Set Trigger control bits for all DMA channels.	NA	Table 257
ABORT0	W1	0x078	Channel Abort control for all DMA channels.	NA	Table 258
Channel0 registers					
CFG0	R/W	0x400	Configuration register for DMA channel 0.		Table 259
CTLSTAT0	RO	0x404	Control and status register for DMA channel 0.		Table 261
XFERCFG0	R/W	0x408	Transfer configuration register for DMA channel 0.		Table 262
Channel1 registers					
CFG1	R/W	0x410	Configuration register for DMA channel 1.		Table 259
CTLSTAT1	RO	0x414	Control and status register for DMA channel 1.		Table 261
XFERCFG1	R/W	0x418	Transfer configuration register for DMA channel 1.		Table 262
Channel2 registers					
CFG2	R/W	0x420	Configuration register for DMA channel 2.		Table 259
CTLSTAT2	RO	0x424	Control and status register for DMA channel 2.		Table 261
XFERCFG2	R/W	0x428	Transfer configuration register for DMA channel 2.		Table 262
Channel3 registers					
CFG3	R/W	0x430	Configuration register for DMA channel 3.		Table 259
CTLSTAT3	RO	0x434	Control and status register for DMA channel 3.		Table 261
XFERCFG3	R/W	0x438	Transfer configuration register for DMA channel 3.		Table 262
Channel4 registers					
CFG4	R/W	0x440	Configuration register for DMA channel 4.		Table 259
CTLSTAT4	RO	0x444	Control and status register for DMA channel 4.		Table 261
XFERCFG4	R/W	0x448	Transfer configuration register for DMA channel 4.		Table 262
Channel5 registers					
CFG5	R/W	0x450	Configuration register for DMA channel 5.		Table 259
CTLSTAT5	RO	0x454	Control and status register for DMA channel 5.		Table 261
XFERCFG5	R/W	0x458	Transfer configuration register for DMA channel 5.		Table 262
Channel6 registers					
CFG6	R/W	0x460	Configuration register for DMA channel 6.		Table 259
CTLSTAT6	RO	0x464	Control and status register for DMA channel 6.		Table 261

Table 242. Register overview: DMA controller (base address 0x5000 5000)

Name	Access	Address offset	Description	Reset Value	Reference
XFERCFG6	R/W	0x468	Transfer configuration register for DMA channel 6.		Table 262
Channel7 registers					
CFG7	R/W	0x470	Configuration register for DMA channel 7.		Table 259
CTLSTAT7	RO	0x474	Control and status register for DMA channel 7.		Table 261
XFERCFG7	R/W	0x478	Transfer configuration register for DMA channel 7.		Table 262
Channel8 registers					
CFG8	R/W	0x480	Configuration register for DMA channel 8.		Table 259
CTLSTAT8	RO	0x484	Control and status register for DMA channel 8.		Table 261
XFERCFG8	R/W	0x488	Transfer configuration register for DMA channel 8.		Table 262
Channel9 registers					
CFG9	R/W	0x490	Configuration register for DMA channel 9.		Table 259
CTLSTAT9	RO	0x494	Control and status register for DMA channel 9.		Table 261
XFERCFG9	R/W	0x498	Transfer configuration register for DMA channel 9.		Table 262
Channel10 registers					
CFG10	R/W	0x4A0	Configuration register for DMA channel 10.		Table 259
CTLSTAT10	RO	0x4A4	Control and status register for DMA channel 10.		Table 261
XFERCFG10	R/W	0x4A8	Transfer configuration register for DMA channel 10.		Table 262
Channel11 registers					
CFG11	R/W	0x4B0	Configuration register for DMA channel 11.		Table 259
CTLSTAT11	RO	0x4B4	Control and status register for DMA channel 11.		Table 261
XFERCFG11	R/W	0x4B8	Transfer configuration register for DMA channel 11.		Table 262
Channel12 registers					
CFG12	R/W	0x4C0	Configuration register for DMA channel 12.		Table 259
CTLSTAT12	RO	0x4C4	Control and status register for DMA channel 12.		Table 261
XFERCFG12	R/W	0x4C8	Transfer configuration register for DMA channel 12.		Table 262
Channel13 registers					
CFG13	R/W	0x4D0	Configuration register for DMA channel 13.		Table 259
CTLSTAT13	RO	0x4D4	Control and status register for DMA channel 13.		Table 261
XFERCFG13	R/W	0x4D8	Transfer configuration register for DMA channel 13.		Table 262
Channel14 registers					
CFG14	R/W	0x4E0	Configuration register for DMA channel 14.		Table 259
CTLSTAT14	RO	0x4E4	Control and status register for DMA channel 14.		Table 261
XFERCFG14	R/W	0x4E8	Transfer configuration register for DMA channel 14.		Table 262
Channel15 registers					
CFG15	R/W	0x4F0	Configuration register for DMA channel 15.		Table 259
CTLSTAT15	RO	0x4F4	Control and status register for DMA channel 15.		Table 261
XFERCFG15	R/W	0x4F8	Transfer configuration register for DMA channel 15.		Table 262

17.6.1 Control register

The CTRL register contains global the control bit for a enabling the DMA controller.

Table 243. Control register (CTRL, address 0x5000 8000) bit description

Bit	Symbol	Value	Description	Reset value
0	ENABLE		DMA controller master enable.	0
		0	Disabled. The DMA controller is disabled. This clears any triggers that were asserted at the point when disabled, but does not prevent re-triggering when the DMA controller is re-enabled.	
		1	Enabled. The DMA controller is enabled.	
31:1	-		Reserved. Read value is undefined, only zero should be written.	NA

17.6.2 Interrupt Status register

The Read-Only INTSTAT register provides an overview of DMA status. This allows quick determination of whether any enabled interrupts are pending. Details of which channels are involved are found in the interrupt type specific registers.

Table 244. Interrupt Status register (INTSTAT, address 0x5000 8004) bit description

Bit	Symbol	Value	Description	Reset value
0	-		Reserved. Read value is undefined, only zero should be written.	NA
1	ACTIVEINT		Summarizes whether any enabled interrupts are pending (except pending error interrupts).	0
		0	Not pending. No enabled interrupts are pending.	
		1	Pending. At least one enabled interrupt is pending.	
2	ACTIVEERRINT		Summarizes whether any error interrupts are pending.	0
		0	Not pending. No error interrupts are pending.	
		1	Pending. At least one error interrupt is pending.	
31:3	-		Reserved. Read value is undefined, only zero should be written.	NA

17.6.3 SRAM Base address register

The SRAMBASE register must be configured with an address (preferably in on-chip SRAM) where DMA descriptors will be stored. Software must set up the descriptors for those DMA channels that will be used in the application.

Table 245. SRAM Base address register (SRAMBASE, address 0x5000 8008) bit description

Bit	Symbol	Description	Reset value
8:0	-	Reserved. Read value is undefined, only zero should be written.	NA
31:9	OFFSET	Address bits 31:9 of the beginning of the DMA descriptor table. For 18 channels, the table must begin on a 512 byte boundary.	0

Each DMA channel has an entry for the channel descriptor in the SRAM table. The values for each channel start at the address offsets found in Table 246. Only the descriptors for channels defined at extraction are used. The contents of each channel descriptor are described in Table 238.

Table 246. Channel descriptor map

Descriptor	Table offset
Channel descriptor for DMA channel 0	0x000
Channel descriptor for DMA channel 1	0x010
Channel descriptor for DMA channel 2	0x020
Channel descriptor for DMA channel 3	0x030
Channel descriptor for DMA channel 4	0x040
Channel descriptor for DMA channel 5	0x050
Channel descriptor for DMA channel 6	0x060
Channel descriptor for DMA channel 7	0x070
Channel descriptor for DMA channel 8	0x080
Channel descriptor for DMA channel 9	0x090
Channel descriptor for DMA channel 10	0x0A0
Channel descriptor for DMA channel 11	0x0B0
Channel descriptor for DMA channel 12	0x0C0
Channel descriptor for DMA channel 13	0x0D0
Channel descriptor for DMA channel 14	0x0E0
Channel descriptor for DMA channel 15	0x0F0

17.6.4 Enable read and Set registers

The ENABLESET0 register determines whether each DMA channel is enabled or disabled. Disabling a DMA channel does not reset the channel in any way. A channel can be paused and restarted by clearing, then setting the Enable bit for that channel.

Reading ENABLESET0 provides the current state of all of the DMA channels represented by that register. Writing a 1 to a bit position in ENABLESET0 that corresponds to an implemented DMA channel sets the bit, enabling the related DMA channel. Writing a 0 to any bit has no effect. Enables are cleared by writing to ENABLECLR0.

Table 247. Enable read and Set register 0 (ENABLESET0, address 0x5000 8020) bit description

Bit	Symbol	Description	Reset value
24:0	ENA	Enable for DMA channels 15:0. Bit n enables or disables DMA channel n. 0 = disabled. 1 = enabled.	0
31:25	-	Reserved.	-

17.6.5 Enable Clear register

The ENABLECLR0 register is used to clear the channel enable bits in ENABLESET0. This register is write-only.

Table 248. Enable Clear register 0 (ENABLECLR0, address 0x5000 8028) bit description

Bit	Symbol	Description	Reset value
24:0	CLR	Writing ones to this register clears the corresponding bits in ENABLESET0. Bit n clears the channel enable bit n.	NA
31:25	-	Reserved.	-

17.6.6 Active status register

The ACTIVE0 register indicates which DMA channels are active at the point when the read occurs. The register is read-only.

A DMA channel is considered active when a DMA operation has been started but not yet fully completed. The Active status will persist from a DMA operation being started, until the pipeline is empty after end of the last descriptor (when there is no reload). An active channel may be aborted by software by setting the appropriate bit in one of the Abort register (see Section 17.6.15).

Table 249. Active status register 0 (ACTIVE0, address 0x5000 8030) bit description

Bit	Symbol	Description	Reset value
24:0	ACT	Active flag for DMA channel n. Bit n corresponds to DMA channel n. 0 = not active. 1 = active.	0
31:25	-	Reserved.	-

17.6.7 Busy status register

The BUSY0 register indicates which DMA channels is busy at the point when the read occurs. This registers is read-only.

A DMA channel is considered busy when there is any operation related to that channel in the DMA controller's internal pipeline. This information can be used after a DMA channel is disabled by software (but still active), allowing confirmation that there are no remaining operations in progress for that channel.

Table 250. Busy status register 0 (BUSY0, address 0x5000 8038) bit description

Bit	Symbol	Description	Reset value
24:0	BSY	Busy flag for DMA channel n. Bit n corresponds to DMA channel n. 0 = not busy. 1 = busy.	0
31:25	-	Reserved.	-

17.6.8 Error Interrupt register

The ERRINT0 register contains flags for each DMA channel's Error Interrupt. Any pending interrupt flag in the register will be reflected on the DMA interrupt output.

Reading the registers provides the current state of all DMA channel error interrupts. Writing a 1 to a bit position in ERRINT0 that corresponds to an implemented DMA channel clears the bit, removing the interrupt for the related DMA channel. Writing a 0 to any bit has no effect.

Table 251. Error Interrupt register 0 (ERRINT0, address 0x5000 8040) bit description

Bit	Symbol	Description	Reset value
24:0	ERR	Error Interrupt flag for DMA channel n. Bit n corresponds to DMA channel n. 0 = error interrupt is not active. 1 = error interrupt is active.	0
31:25	-	Reserved.	-

17.6.9 Interrupt Enable read and Set register

The INTENSET0 register controls whether the individual Interrupts for DMA channels contribute to the DMA interrupt output.

Reading the registers provides the current state of all DMA channel interrupt enables. Writing a 1 to a bit position in INTENSET0 that corresponds to an implemented DMA channel sets the bit, enabling the interrupt for the related DMA channel. Writing a 0 to any bit has no effect. Interrupt enables are cleared by writing to INTENCLR0.

Table 252. Interrupt Enable read and Set register 0 (INTENSET0, address 0x5000 8048) bit description

Bit	Symbol	Description	Reset value
24:0	INTEN	Interrupt Enable read and set for DMA channel n. Bit n corresponds to DMA channel n. 0 = interrupt for DMA channel is disabled. 1 = interrupt for DMA channel is enabled.	0
31:25	-	Reserved.	-

17.6.10 Interrupt Enable Clear register

The INTENCLR0 register is used to clear interrupt enable bits in INTENSET0. The register is write-only.

Table 253. Interrupt Enable Clear register 0 (INTENCLR0, address 0x5000 8050) bit description

Bit	Symbol	Description	Reset value
24:0	CLR	Writing ones to this register clears corresponding bits in the INTENSET0. Bit n corresponds to DMA channel n.	NA
31:25	-	Reserved.	-

17.6.11 Interrupt A register

The IntA0 register contains the interrupt A status for each DMA channel. The status will be set when the SETINTA bit is 1 in the transfer configuration for a channel, when the descriptor becomes exhausted. Writing a 1 to a bit in these registers clears the related INTA flag. Writing 0 has no effect. Any interrupt pending status in this register will be reflected on the DMA interrupt output if it is enabled in the related INTENSET register.

Remark: The error status is not included in this register. The error status is reported in the ERRINT0 status register.

Table 254. Interrupt A register 0 (INTA0, address 0x5000 8058) bit description

Bit	Symbol	Description	Reset value
24:0	IA	Interrupt A status for DMA channel n. Bit n corresponds to DMA channel n. 0 = the DMA channel interrupt A is not active. 1 = the DMA channel interrupt A is active.	0
31:25	-	Reserved.	-

17.6.12 Interrupt B register

The INTB0 register contains the interrupt B status for each DMA channel. The status will be set when the SETINTB bit is 1 in the transfer configuration for a channel, when the descriptor becomes exhausted. Writing a 1 to a bit in the register clears the related INTB flag. Writing 0 has no effect. Any interrupt pending status in this register will be reflected on the DMA interrupt output if it is enabled in the INTENSET register.

Remark: The error status is not included in this register. The error status is reported in the ERRINT0 status register.

Table 255. Interrupt B register 0 (INTB0, address 0x5000 8060) bit description

Bit	Symbol	Description	Reset value
24:0	IB	Interrupt B status for DMA channel n. Bit n corresponds to DMA channel n. 0 = the DMA channel interrupt B is not active. 1 = the DMA channel interrupt B is active.	0
31:25	-	Reserved.	-

17.6.13 Set Valid register

The SETVALID0 register allows setting the Valid bit in the CTRLSTAT register for one or more DMA channels. See Section 17.6.17 for a description of the VALID bit.

The CFGVALID and SV (set valid) bits allow more direct DMA block timing control by software. Each Channel Descriptor, in a sequence of descriptors, can be validated by either the setting of the CFGVALID bit or by setting the channel's SETVALID flag. Normally, the CFGVALID bit is set. This tells the DMA that the Channel Descriptor is active and can be executed. The DMA will continue sequencing through descriptor blocks whose CFGVALID bit are set without further software intervention. Leaving a CFGVALID bit set to 0 allows the DMA sequence to pause at the Descriptor until software triggers the continuation. If, during DMA transmission, a Channel Descriptor is found with CFGVALID set to 0, the DMA checks for a previously buffered SETVALID0 setting for the channel. If found, the DMA will set the descriptor valid, clear the SV setting, and resume processing the descriptor. Otherwise, the DMA pauses until the channels SETVALID0 bit is set.

Table 256. Set Valid 0 register (SETVALID0, address 0x5000 8068) bit description

Bit	Symbol	Description	Reset value
24:0	SV	SETVALID control for DMA channel n. Bit n corresponds to DMA channel n. 0 = no effect. 1 = sets the VALIDPENDING control bit for DMA channel n.	NA
31:25	-	Reserved.	-

17.6.14 Set Trigger register

The SETTRIG0 register allows setting the TRIG bit in the CTRLSTAT register for one or more DMA channel. See Section 17.6.17 for a description of the TRIG bit, and Section 17.5.1 for a general description of triggering.

Table 257. Set Trigger 0 register (SETTRIG0, address 0x5000 8070) bit description

Bit	Symbol	Description	Reset value
24:0	TRIG	Set Trigger control bit for DMA channel 0. Bit n corresponds to DMA channel n. 0 = no effect. 1 = sets the TRIG bit for DMA channel n.	NA
31:25	-	Reserved.	-

17.6.15 Abort registers

The Abort0 register allows aborting operation of a DMA channel if needed. To abort a selected channel, the channel should first be disabled by clearing the corresponding Enable bit by writing a 1 to the proper bit ENABLECLR. Then wait until the channel is no longer busy by checking the corresponding bit in BUSY. Finally, write a 1 to the proper bit of ABORT. This prevents the channel from restarting an incomplete operation when it is enabled again.

Table 258. Abort 0 register (ABORT0, address 0x5000 8078) bit description

Bit	Symbol	Description	Reset value
24:0	ABORTCTRL	Abort control for DMA channel 0. Bit n corresponds to DMA channel n. 0 = no effect. 1 = aborts DMA operations on channel n.	NA
31:25	-	Reserved.	-

17.6.16 Channel configuration registers

The CFGn register contains various configuration options for DMA channel n.

See Table 260 for a summary of trigger options.

Table 259. Configuration registers for channel 0 to 15(CFG[0:15], addresses 0x5000 8400 (CFG0) to address 0x5000 84F0 (CFG15)) bit description

Bit	Symbol	Value	Description	Reset value
0	PERIPHREQEN		Peripheral request Enable. If a DMA channel is used to perform a memory-to-memory move, any peripheral DMA request associated with that channel can be disabled to prevent any interaction between the peripheral and the DMA controller.	0
		0	Disabled. Peripheral DMA requests are disabled.	
		1	Enabled. Peripheral DMA requests are enabled.	
1	HWTRIGEN		Hardware Triggering Enable for this channel.	0
		0	Disabled. Hardware triggering is not used.	
		1	Enabled. Use hardware triggering.	
3:2	-		Reserved. Read value is undefined, only zero should be written.	NA
4	TRIGPOL		Trigger Polarity. Selects the polarity of a hardware trigger for this channel.	0
		0	Active low - falling edge. Hardware trigger is active low or falling edge triggered, based on TRIGTYPE.	
		1	Active high - rising edge. Hardware trigger is active high or rising edge triggered, based on TRIGTYPE.	
5	TRIGTYPE		Trigger Type. Selects hardware trigger as edge triggered or level triggered.	0
		0	Edge. Hardware trigger is edge triggered. Transfers will be initiated and completed, as specified for a single trigger.	
		1	Level. Hardware trigger is level triggered. Note that when level triggering without burst (BURSTPOWER = 0) is selected, only hardware triggers should be used on that channel. Transfers continue as long as the trigger level is asserted. Once the trigger is de-asserted, the transfer will be paused until the trigger is, again, asserted. However, the transfer will not be paused until any remaining transfers within the current BURSTPOWER length are completed.	
6	TRIGBURST		Trigger Burst. Selects whether hardware triggers cause a single or burst transfer.	0
		0	Single transfer. Hardware trigger causes a single transfer.	
		1	Burst transfer. When the trigger for this channel is set to edge triggered, a hardware trigger causes a burst transfer, as defined by BURSTPOWER. When the trigger for this channel is set to level triggered, a hardware trigger causes transfers to continue as long as the trigger is asserted, unless the transfer is complete.	
7	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 259. Configuration registers for channel 0 to 15(CFG[0:15], addresses 0x5000 8400 (CFG0) to address 0x5000 84F0 (CFG15)) bit description

Bit	Symbol	Value	Description	Reset value
11:8	BURSTPOWER		Burst Power is used in two ways. It always selects the address wrap size when SRCBURSTWRAP and/or DSTBURSTWRAP modes are selected (see descriptions elsewhere in this register). When the TRIGBURST field elsewhere in this register = 1, Burst Power selects how many transfers are performed for each DMA trigger. This can be used, for example, with peripherals that contain a FIFO that can initiate a DMA operation when the FIFO reaches a certain level. 0000: Burst size = 1 (2^0). 0001: Burst size = 2 (2^1). 0010: Burst size = 4 (2^2). ... 1010: Burst size = 1024 (2^{10}). This corresponds to the maximum supported transfer count. others: not supported. The total transfer length as defined in the XFERCOUNT bits in the XFERCFG register must be an even multiple of the burst size.	0
13:12	-		Reserved. Read value is undefined, only zero should be written.	NA
14	SRCBURSTWRAP		Source Burst Wrap. When enabled, the source data address for the DMA is “wrapped”, meaning that the source address range for each burst will be the same. As an example, this could be used to read several sequential registers from a peripheral for each DMA burst, reading the same registers again for each burst.	0
		0	Disabled. Source burst wrapping is not enabled for this DMA channel.	
		1	Enabled. Source burst wrapping is enabled for this DMA channel.	
15	DSTBURSTWRAP		Destination Burst Wrap. When enabled, the destination data address for the DMA is “wrapped”, meaning that the destination address range for each burst will be the same. As an example, this could be used to write several sequential registers to a peripheral for each DMA burst, writing the same registers again for each burst.	0
		0	Disabled. Destination burst wrapping is not enabled for this DMA channel.	
		1	Enabled. Destination burst wrapping is enabled for this DMA channel.	
17:16	CHPRIORITY		Priority of this channel when multiple DMA requests are pending. Four priority levels are supported. 0x0 = highest priority. 0x3 = lowest priority.	0
31:18	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 260. Trigger setting summary

TrigBurst	TrigType	TrigPol	Description
0	0	0	Hardware DMA trigger is falling edge sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap.
0	0	1	Hardware DMA trigger is rising edge sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap.
0	1	0	Hardware DMA trigger is low level sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap.
0	1	1	Hardware DMA trigger is high level sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap.
1	0	0	Hardware DMA trigger is falling edge sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap, and also determines how much data is transferred for each trigger.
1	0	1	Hardware DMA trigger is rising edge sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap, and also determines how much data is transferred for each trigger.
1	1	0	Hardware DMA trigger is low level sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap, and also determines how much data is transferred for each trigger.
1	1	1	Hardware DMA trigger is high level sensitive. The BURSTPOWER field controls address wrapping if enabled via SrcBurstWrap and/or DstBurstWrap, and also determines how much data is transferred for each trigger.

17.6.17 Channel control and status registers

The CTLSTATn register provides status flags specific to DMA channel n.

Table 261. Control and Status registers for channel 0 to 15 (CTLSTAT[0:15], 0x5000 8404 (CTLSTAT0) to address 0x5000 84F4 (CTLSTAT15)) bit description

Bit	Symbol	Value	Description	Reset value
0	VALIDPENDING		Valid pending flag for this channel. This bit is set when a 1 is written to the corresponding bit in the related SETVALID register when CFGVALID = 1 for the same channel.	0
		0	No effect on DMA operation.	
		1	Valid pending.	
1	-		Reserved. Read value is undefined, only zero should be written.	NA
2	TRIG		Trigger flag. Indicates that the trigger for this channel is currently set. This bit is cleared at the end of an entire transfer or upon reload when CLRTRIG = 1.	0
		0	Not triggered. The trigger for this DMA channel is not set. DMA operations will not be carried out.	
		1	Triggered. The trigger for this DMA channel is set. DMA operations will be carried out.	
31:3	-		Reserved. Read value is undefined, only zero should be written.	NA

17.6.18 Channel transfer configuration registers

The XFERCFGn register contains transfer related configuration information for DMA channel n. Using the Reload bit, this register can optionally be automatically reloaded when the current settings are exhausted (the full transfer count has been completed), allowing linked transfers with more than one descriptor to be performed.

See "Trigger operation" for details on trigger operation.

Table 262. Transfer Configuration registers for channel 0 to 15 (XFERCFG[0:15], addresses 0x5000 8408 (XFERCFG0) to 0x5000 84F8 (XFERCFG24)) bit description

Bit	Symbol	Value	Description	Reset Value
0	CFGVALID		Configuration Valid flag. This bit indicates whether the current channel descriptor is valid and can potentially be acted upon, if all other activation criteria are fulfilled.	0
		0	Not valid. The channel descriptor is not considered valid until validated by an associated SETVALID0 setting.	
		1	Valid. The current channel descriptor is considered valid.	
1	RELOAD		Indicates whether the channel's control structure will be reloaded when the current descriptor is exhausted. Reloading allows ping-pong and linked transfers.	0
		0	Disabled. Do not reload the channels' control structure when the current descriptor is exhausted.	
		1	Enabled. Reload the channels' control structure when the current descriptor is exhausted.	
2	SWTRIG		Software Trigger.	0
		0	When written by software, the trigger for this channel is not set. A new trigger, as defined by the HWTRIGEN, TRIGPOL, and TRIGTYPE will be needed to start the channel.	
		1	When written by software, the trigger for this channel is set immediately. This feature should not be used with level triggering when TRIGBURST = 0.	
3	CLRTRIG		Clear Trigger.	0
		0	Not cleared. The trigger is not cleared when this descriptor is exhausted. If there is a reload, the next descriptor will be started.	
		1	Cleared. The trigger is cleared when this descriptor is exhausted.	
4	SETINTA		Set Interrupt flag A for this channel. There is no hardware distinction between interrupt A and B. They can be used by software to assist with more complex descriptor usage. By convention, interrupt A may be used when only one interrupt flag is needed.	0
		0	No effect.	
		1	Set. The INTA flag for this channel will be set when the current descriptor is exhausted.	
5	SETINTB		Set Interrupt flag B for this channel. There is no hardware distinction between interrupt A and B. They can be used by software to assist with more complex descriptor usage. By convention, interrupt A may be used when only one interrupt flag is needed.	0
		0	No effect.	
		1	Set. The INTB flag for this channel will be set when the current descriptor is exhausted.	
7:6	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 262. Transfer Configuration registers for channel 0 to 15 (XFERCFG[0:15], addresses 0x5000 8408 (XFERCFG0) to 0x5000 84F8 (XFERCFG24)) bit description

Bit	Symbol	Value	Description	Reset Value
9:8	WIDTH		Transfer width used for this DMA channel.	0
		0x0	8-bit transfers are performed (8-bit source reads and destination writes).	
		0x1	16-bit transfers are performed (16-bit source reads and destination writes).	
		0x2	32-bit transfers are performed (32-bit source reads and destination writes).	
		0x3	Reserved setting, do not use.	
11:10	-		Reserved. Read value is undefined, only zero should be written.	NA
13:12	SRCINC		Determines whether the source address is incremented for each DMA transfer.	0
		0x0	No increment. The source address is not incremented for each transfer. This is the usual case when the source is a peripheral device.	
		0x1	1 x width. The source address is incremented by the amount specified by Width for each transfer. This is the usual case when the source is memory.	
		0x2	2 x width. The source address is incremented by 2 times the amount specified by Width for each transfer.	
		0x3	4 x width. The source address is incremented by 4 times the amount specified by Width for each transfer.	
15:14	DSTINC		Determines whether the destination address is incremented for each DMA transfer.	0
		0x0	No increment. The destination address is not incremented for each transfer. This is the usual case when the destination is a peripheral device.	
		0x1	1 x width. The destination address is incremented by the amount specified by Width for each transfer. This is the usual case when the destination is memory.	
		0x2	2 x width. The destination address is incremented by 2 times the amount specified by Width for each transfer.	
		0x3	4 x width. The destination address is incremented by 4 times the amount specified by Width for each transfer.	
25:16	XFERCOUNT		Total number of transfers to be performed, minus 1 encoded. The number of bytes transferred is: (XFERCOUNT + 1) x data width (as defined by the WIDTH field). **Remark:** The DMA controller uses this bit field during transfer to count down. Hence, it cannot be used by software to read back the size of the transfer, for instance, in an interrupt handler. 0x0 = a total of 1 transfer will be performed. 0x1 = a total of 2 transfers will be performed. ... 0x3FF = a total of 1,024 transfers will be performed.	0
31:26	-		Reserved. Read value is undefined, only zero should be written.	NA

17.7 Functional description

17.7.1 Trigger operation

A trigger of some kind is always needed to start a transfer on a DMA channel. This can be a hardware or software trigger, and can be used in several ways.

If a channel is configured with the SWTRIG bit equal to 0, the channel can be later triggered either by hardware or software. Software triggering is accomplished by writing a 1 to the appropriate bit in the SETTRIG register. Hardware triggering requires setup of the HWTRIGEN, TRIGPOL, TRIGTYPE, and TRIGBURST fields in the CFG register for the related channel. When a channel is initially set up, the SWTRIG bit in the XFERCFG register can be set, causing the transfer to begin immediately.

Once triggered, transfer on a channel will be paced by DMA requests if the PERIPHREQEN bit in the related CFG register is set. Otherwise, the transfer will proceed at full speed.

The TRIG bit in the CTLSTAT register can be cleared at the end of a transfer, determined by the value CLRTRIG (bit 0) in the XFERCFG register. When a 1 is found in CLRTRIG, the trigger is cleared when the descriptor is exhausted.

UM11607

Chapter 18: USART0/1/2

Rev. 3 — April 2023 User manual

18.1 How to read this chapter

Three USARTs are available on all parts depending on the switch matrix configuration.

18.2 Features

- 7, 8, or 9 data bits and 1 or 2 stop bits
- Synchronous mode with master or slave operation. Includes data phase selection and continuous clock option.
- Multiprocessor/multidrop (9-bit) mode with software address compare.
- RS-485 transceiver output enable.
- Parity generation and checking: odd, even, or none.
- Software selectable oversampling from 5 to 16 clocks in asynchronous mode.
- One transmit and one receive data buffer.
- RTS/CTS for hardware signaling for automatic flow control. Software flow control can be performed using Delta CTS detect, Transmit Disable control, and any GPIO as an RTS output.
- Received data and status can optionally be read from a single register
- Break generation and detection.
- Receive data is 2 of 3 sample "voting". Status flag set when one sample differs.
- Built-in Baud Rate Generator with autobaud function.
- A fractional rate divider is shared among all USARTs.
- Interrupts available for Receiver Ready, Transmitter Ready, Receiver Idle, change in receiver break detect, Framing error, Parity error, Overrun, Underrun, Delta CTS detect, and receiver sample noise detected.
- Loopback mode for testing of data and flow control.
- USARTn transmit and receive functions can operated with the system DMA controller.

18.3 Basic configuration

Configure the USARTs for receiving and transmitting data:

- In the SYSAHBCLKCTRL register, set bit 14 to 16, bit 30, and 31 (Table 94) to enable the clock to the register interface.
- Clear the USART peripheral resets using the PRESETCTRL register (Table 95).
- Enable or disable the USART interrupts in slots #3, #4, and #5,in the NVIC. See Table 46.
- Configure the USART pin functions through the switch matrix.
- Configure the USART clock and baud rate. See Section 18.3.1.

- Send and receive lines are connected to DMA request lines. See Table 238 “Channel descriptor”.

For wake-up from deep-sleep and power-down modes the USART must be configured in synchronous mode. See Section 18.3.2 for details.

18.3.1 Configure the USART clock and baud rate

All three USARTs have a separate clock selection that include two shared fractional dividers. The peripheral clock and the fractional divider for the baud rate calculation are set up in the SYSCON block as follows (see Figure 22):

1. If a fractional value is needed to obtain a particular baud rate, program the fractional divider. The fractional divider value is the fraction of MULT/DIV. The MULT and DIV values are programmed in the FRGCTRL register. The DIV value must be programmed with the fixed value of 256.

   FCLK = (FRGINPUTCLK)/(1+(MULT/DIV))

   The following rules apply for MULT and DIV:

   – Always set DIV to 256 by programming the FRGCTRL register with the value of 0xFF.

   – Set the MULT to any value between 0 and 255.

2. In asynchronous mode: Configure the baud rate divider BRGVAL in the USARTn BRG register. The baud rate divider divides the FCLK clock to create the clock needed to produce the required baud rate.

   baud rate = FCLK/((OSRVAL+1) x (BRGVAL+1)).

   BRGVAL = FCLK/((OSRVAL + 1) x baud rate) – 1

   (assumes FCLK $\geq$ oversample rate x baud rate)

   Section 18.6.9 “USART Baud Rate Generator register”

3. In synchronous mode: The serial clock is Un_SCLK = FCLK/(BRGVAL+1).

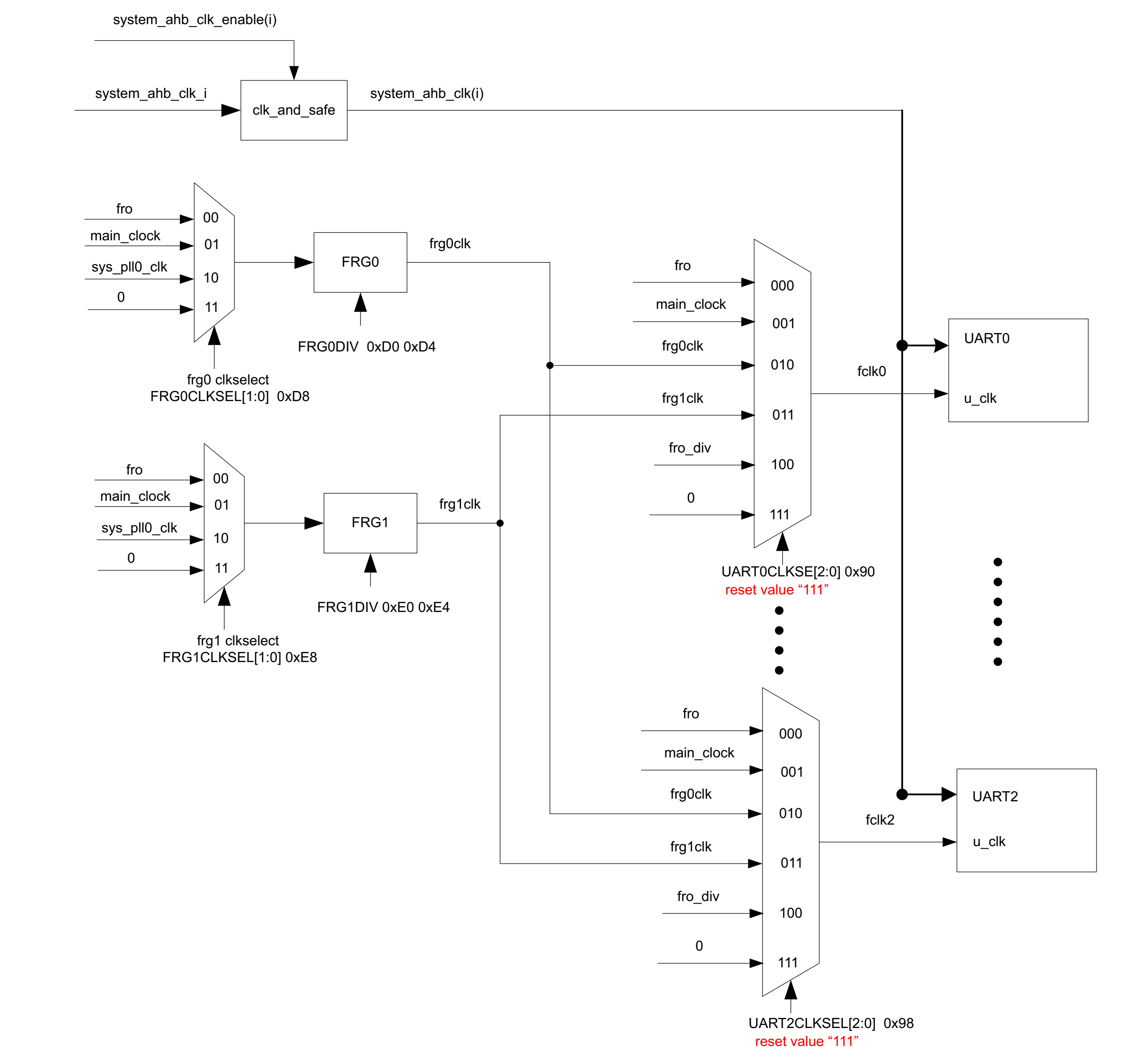


Fig 22. USART clocking

For details on the clock configuration see:

Section 18.7.1 “Clocking and baud rates”

18.3.2 Configure the USART for wake-up

The USART can wake up the system from sleep mode in asynchronous or synchronous mode on any enabled USART interrupt.

In deep-sleep or power-down mode, you can configure the USART as a wake-up source if the USART is configured in synchronous slave mode. The USART block can create an interrupt on a received signal even when the USART block receives no clocks from the

ARM core, that is, in deep-sleep or power-down mode. As long as the USART receives a clock signal from the master, it can receive up to one byte in the RXDAT register while in deep-sleep or power-down mode. Any interrupt raised as part of the receive data process can then wake up the part.

18.3.2.1 Wake-up from sleep mode

- Configure the USART in either asynchronous mode or synchronous mode. See Table 265.
- Enable the USART interrupt in the NVIC.
- Any USART interrupt wakes up the part from sleep mode. Enable the USART interrupt in the INTENSET register (Table 268).

18.3.2.2 Wake-up from deep-sleep or power-down mode

- Configure the USART in synchronous slave mode. See Table 265. You must connect the SCLK function to a pin and connect the pin to the master.
- Enable the USART wake-up in the STARTERP1 register. See Table 115 “Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description”.
- Enable the USART interrupt in the NVIC.
- In the PDAWAKE register, configure all peripherals that need to be running when the part wakes up.
- The USART wakes up the part from deep-sleep or power-down mode on all events that cause an interrupt and are also enabled in the INTENSET register. Typical wake-up events are:
  - A start bit has been received.
  - The RXDAT buffer has received a byte.
  - Data is ready to be transmitted in the TXDAT buffer and a serial clock from the master has been received.
  - A change in the state of the CTS pin if the CTS function is connected and the DELTACTS interrupt is enabled. This event wakes up the part without the synchronous UART clock running.

  Remark: By enabling or disabling the interrupt in the INTENSET register (Table 268), you can customize when the wake-up occurs in the USART receive/transmit protocol.

18.4 Pin description

Table 263. USART pin description

Function	I/O	Type	Connect to	Use register	Reference	Description
U0_TXD	O	external to pin	any pin	PINASSIGN0	Table 128	Transmitter output for USART0. Serial transmit data.
U0_RXD	I	external to pin	any pin	PINASSIGN0	Table 128	Receiver input for USART0.
$\overline{\text{U0\_RTS}}$	O	external to pin	any pin	PINASSIGN0	Table 128	Request To Send output for USART0. This signal supports inter-processor communication through the use of hardware flow control. This feature is active when the USART RTS signal is configured to appear on a device pin.
$\overline{\text{U0\_CTS}}$	I	external to pin	any pin	PINASSIGN0	Table 128	Clear To Send input for USART0. Active low signal indicates that the external device that is in communication with the USART is ready to accept data. This feature is active when enabled by the CTSEn bit in CFG register and when configured to appear on a device pin. When de-asserted (high) by the external device, the USART will complete transmitting any character already in progress, then stop until CTS is again asserted (low).
U0_SCLK	I/O	external to pin	any pin	PINASSIGN1	Table 129	Serial clock input/output for USART0 n synchronous mode.
U1_TXD	O	external to pin	any pin	PINASSIGN1	Table 129	Transmitter output for USART1. Serial transmit data.
U1_RXD	I	external to pin	any pin	PINASSIGN1	Table 129	Receiver input for USART1.
$\overline{\text{U1\_RTS}}$	O	external to pin	any pin	PINASSIGN1	Table 129	Request To Send output for USART1.
$\overline{\text{U1\_CTS}}$	I	external to pin	any pin	PINASSIGN2	Table 130	Clear To Send input for USART1.
U1_SCLK	I/O	external to pin	any pin	PINASSIGN2	Table 130	Serial clock input/output for USART1 in synchronous mode.
U2_TXD	O	external to pin	any pin	PINASSIGN2	Table 130	Transmitter output for USART2. Serial transmit data.
U2_RXD	I	external to pin	any pin	PINASSIGN2	Table 130	Receiver input for USART2.
$\overline{\text{U2\_RTS}}$	O	external to pin	any pin	PINASSIGN3	Table 131	Request To Send output for USART2.
$\overline{\text{U2\_CTS}}$	I	external to pin	any pin	PINASSIGN3	Table 131	Clear To Send input for USART2.
U2_SCLK	I/O	external to pin	any pin	PINASSIGN3	Table 131	Serial clock input/output for USART2 in synchronous mode.

18.5 General description

The USART receiver block monitors the serial input line, Un_RXD, for valid input. The receiver shift register assembles characters as they are received, after which they are passed to the receiver buffer register to await access by the CPU or the DMA controller. When RTS signal is configured as an RS-485 output enable, it is asserted at the beginning of an transmitted character, and de-asserted either at the end of the character, or after a one character delay (selected by software).

The USART transmitter block accepts data written by the CPU or DMA controllers and buffers the data in the transmit holding register. When the transmitter is available, the transmit shift register takes that data, formats it, and serializes it to the serial output, Un_TXD.

The Baud Rate Generator block divides the incoming clock to create a 16x baud rate clock in the standard asynchronous operating mode. The BRG clock input source is the shared Fractional Rate Generator that runs from the common USART peripheral clock U_PCLK).

In synchronous slave mode, data is transmitted and received using the serial clock directly. In synchronous master mode, data is transmitted and received using the baud rate clock without division.

Status information from the transmitter and receiver is saved and provided via the Stat register. Many of the status flags are able to generate interrupts, as selected by software.

Remark: The fractional value and the USART peripheral clock are shared between all USARTs.

After a period of activity, when UART receiver is idle for a certain period of time, STAT.RXIDLETO flag is set and interrupt will be asserted if enabled. Writing 1 to STAT.RXIDLETO clears the flag. Once cleared, it can't be set again until after the receiver receives a new character (non-idle).

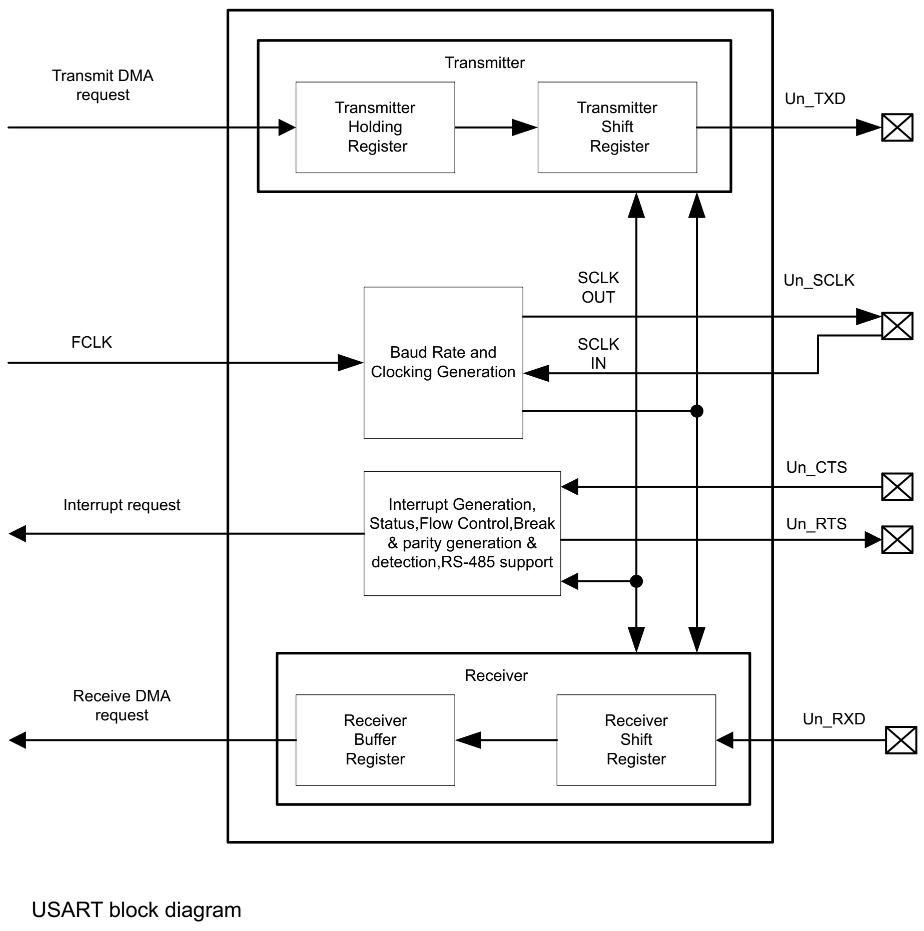


Fig 23. USART block diagram

Remark: The existing STAT.RXIDLE flag is not affected.

18.6 Register description

The reset value reflects the data stored in used bits only. It does not include the content of reserved bits.

Table 264. Register overview: USART (base address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2))

Name	Access	Offset	Description	Reset value	Reference
CFG	R/W	0x000	USART Configuration register. Basic USART configuration settings that typically are not changed during operation.	0	Table 265
CTL	R/W	0x004	USART Control register. USART control settings that are more likely to change during operation.	0	Table 266
STAT	R/W	0x008	USART Status register. The complete status value can be read here. Writing ones clears some bits in the register. Some bits can be cleared by writing a 1 to them.	0x000E	Table 267
INTENSET	R/W	0x00C	Interrupt Enable read and Set register. Contains an individual interrupt enable bit for each potential USART interrupt. A complete value may be read from this register. Writing a 1 to any implemented bit position causes that bit to be set.	0	Table 268
INTENCLR	W	0x010	Interrupt Enable Clear register. Allows clearing any combination of bits in the INTENSET register. Writing a 1 to any implemented bit position causes the corresponding bit to be cleared.	-	Table 269
RXDAT	R	0x014	Receiver Data register. Contains the last character received.	-	Table 270
RXDATSTAT	R	0x018	Receiver Data with Status register. Combines the last character received with the current USART receive status. Allows DMA or software to recover incoming data and status together.	-	Table 271
TXDAT	R/W	0x01C	Transmit Data register. Data to be transmitted is written here.	0	Table 272
BRG	R/W	0x020	Baud Rate Generator register. 16-bit integer baud rate divisor value.	0	Table 273
INTSTAT	R	0x024	Interrupt status register. Reflects interrupts that are currently enabled.	0x0005	Table 274
OSR	R/W	0x028	Oversample selection register for asynchronous communication.	0xF	Table 275
ADDR	R/W	0x02C	Address register for automatic address matching.	0	Table 276

18.6.1 USART Configuration register

The CFG register contains communication and mode settings for aspects of the USART that would normally be configured once in an application.

Remark: If software needs to change configuration values, the following sequence should be used: 1) Make sure the USART is not currently sending or receiving data. 2) Disable the USART by writing a 0 to the Enable bit (0 may be written to the entire register). 3) Write the new configuration value, with the ENABLE bit set to 1.

Table 265. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2)) bit description

Bit	Symbol	Value	Description	Reset Value
0	ENABLE		USART Enable.	0
		0	Disabled. The USART is disabled and the internal state machine and counters are reset. While Enable = 0, all USART interrupts and DMA transfers are disabled. When Enable is set again, CFG and most other control bits remain unchanged. For instance, when re-enabled, the USART will immediately generate a TXRDY interrupt (if enabled in the INTENSET register) or a DMA transfer request because the transmitter has been reset and is therefore available.	
		1	Enabled. The USART is enabled for operation.	
1	-		Reserved. Read value is undefined, only zero should be written.	NA
3:2	DATALEN		Selects the data size for the USART.	00
		0x0	7 bit Data length.	
		0x1	8 bit Data length.	
		0x2	9 bit data length. The 9th bit is commonly used for addressing in multidrop mode. See the ADDRDET bit in the CTL register.	
		0x3	Reserved.	
5:4	PARITYSEL		Selects what type of parity is used by the USART.	00
		0x0	No parity.	
		0x1	Reserved.	
		0x2	Even parity. Adds a bit to each character such that the number of 1s in a transmitted character is even, and the number of 1s in a received character is expected to be even.	
		0x3	Odd parity. Adds a bit to each character such that the number of 1s in a transmitted character is odd, and the number of 1s in a received character is expected to be odd.	
6	STOPLEN		Number of stop bits appended to transmitted data. Only a single stop bit is required for received data.	0
		0	1 stop bit.	
		1	2 stop bits. This setting should only be used for asynchronous communication.	
8:7	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 265. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2)) bit description *…continued*

Bit	Symbol	Value	Description	Reset Value
9	CTSEN		CTS Enable. Determines whether CTS is used for flow control. CTS can be from the input pin, or from the USART's own RTS if loopback mode is enabled.	0
		0	No flow control. The transmitter does not receive any automatic flow control signal.	
		1	Flow control enabled. The transmitter uses the CTS input (or RTS output in loopback mode) for flow control purposes.	
10	-		Reserved. Read value is undefined, only zero should be written.	NA
11	SYNCEN		Selects synchronous or asynchronous operation.	0
		0	Asynchronous mode is selected.	
		1	Synchronous mode is selected.	
12	CLKPOL		Selects the clock polarity and sampling edge of received data in synchronous mode.	0
		0	Falling edge. Un_RXD is sampled on the falling edge of SCLK.	
		1	Rising edge. Un_RXD is sampled on the rising edge of SCLK.	
13	-		Reserved. Read value is undefined, only zero should be written.	NA
14	SYNCMST		Synchronous mode Master select.	0
		0	Slave. When synchronous mode is enabled, the USART is a slave.	
		1	Master. When synchronous mode is enabled, the USART is a master.	
15	LOOP		Selects data loopback mode.	0
		0	Normal operation.	
		1	Loopback mode. This provides a mechanism to perform diagnostic loopback testing for USART data. Serial data from the transmitter (Un_TXD) is connected internally to serial input of the receive (Un_RXD). Un_TXD and Un_RTS activity will also appear on external pins if these functions are configured to appear on device pins. The receiver RTS signal is also looped back to CTS and performs flow control if enabled by CTSEN.	
17:16	-		Reserved. Read value is undefined, only zero should be written.	NA
18	OETA		Output Enable Turnaround time enable for RS-485 operation.	0
		0	De-asserted. If selected by OESEL, the Output Enable signal de-asserted at the end of the last stop bit of a transmission.	
		1	Asserted. If selected by OESEL, the Output Enable signal remains asserted for 1 character time after then end the last stop bit of a transmission. OE will also remain asserted if another transmit begins before it is de-asserted.	

Table 265. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2)) bit description *…continued*

Bit	Symbol	Value	Description	Reset Value
19	AUTOADDR		Automatic Address matching enable.	0
		0	Disabled. When addressing is enabled by ADDRDET, address matching is done by software. This provides the possibility of versatile addressing (e.g. respond to more than one address).	
		1	Enabled. When addressing is enabled by ADDRDET, address matching is done by hardware, using the value in the ADDR register as the address to match.	
20	OESEL		Output Enable Select.	0
		0	Flow control. The RTS signal is used as the standard flow control function.	
		1	Output enable. The RTS signal is taken over in order to provide an output enable signal to control an RS-485 transceiver.	
21	OEPOL		Output Enable Polarity.	0
		0	Low. If selected by OESEL, the output enable is active low.	
		1	High. If selected by OESEL, the output enable is active high.	
22	RXPOL		Receive data polarity.	0
		0	Not changed. The RX signal is used as it arrives from the pin. This means that the RX rest value is 1, start bit is 0, data is not inverted, and the stop bit is 1.	
		1	Inverted. The RX signal is inverted before being used by the UART. This means that the RX rest value is 0, start bit is 1, data is inverted, and the stop bit is 0.	
23	TXPOL		Transmit data polarity.	0
		0	Not changed. The TX signal is sent out without change. This means that the TX rest value is 1, start bit is 0, data is not inverted, and the stop bit is 1.	
		1	Inverted. The TX signal is inverted by the UART before being sent out. This means that the TX rest value is 0, start bit is 1, data is inverted, and the stop bit is 0.	
31:24	-		Reserved. Read value is undefined, only zero should be written.	NA

18.6.2 USART Control register

The CTL register controls aspects of USART operation that are more likely to change during operation.

Table 266. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1), 0x4006 C004 (USART2)) bit description

Bit	Symbol	Value	Description	Reset value
0	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 266. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1), 0x4006 C004 (USART2)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
1	TXBRKEN		Break Enable.	0
		0	Normal operation.	
		1	Continuous break is sent immediately when this bit is set, and remains until this bit is cleared. A break may be sent without danger of corrupting any currently transmitting character if the transmitter is first disabled (TXDIS in CTL is set) and then waiting for the transmitter to be disabled (TXDISINT in STAT = 1) before writing 1 to TXBRKEN.	
2	ADDRDET		Enable address detect mode.	0
		0	Disabled. The USART presents all incoming data.	
		1	Enabled. The USART receiver ignores incoming data that does not have the most significant bit of the data (typically the 9th bit) = 1. When the data MSB bit = 1, the receiver treats the incoming data normally, generating a received data interrupt. Software can then check the data to see if this is an address that should be handled. If it is, the ADDRDET bit is cleared by software and further incoming data is handled normally.	
5:3	-		Reserved. Read value is undefined, only zero should be written.	NA
6	TXDIS		Transmit Disable.	0
		0	Not disabled. USART transmitter is not disabled.	
		1	Disabled. USART transmitter is disabled after any character currently being transmitted is complete. This feature can be used to facilitate software flow control.	
7	-		Reserved. Read value is undefined, only zero should be written.	NA
8	CC		Continuous Clock generation. By default, SCLK is only output while data is being transmitted in synchronous mode.	0
		0	Clock on character. In synchronous mode, SCLK cycles only when characters are being sent on Un_TXD or to complete a character that is being received.	
		1	Continuous clock. SCLK runs continuously in synchronous mode, allowing characters to be received on Un_RxD independently from transmission on Un_TXD).	
9	CLRCCONRX		Clear Continuous Clock.	0
		0	No effect on the CC bit.	
		1	Auto-clear. The CC bit is automatically cleared when a complete character has been received. This bit is cleared at the same time.	
15:10	-		Reserved. Read value is undefined, only zero should be written.	NA
16	AUTOBAUD		Autobaud enable.	0
		0	Disabled. UART is in normal operating mode.	
		1	Enabled. UART is in autobaud mode. This bit should only be set when the UART is enabled in the CFG register and the UART receiver is idle. The first start bit of RX is measured and used the update the BRG register to match the received data rate. AUTOBAUD is cleared once this process is complete, or if there is an ABERR. This bit can be cleared by software when set, but only when the UART receiver is idle. Disabling the UART in the CFG register also clears the AUTOBAUD bit.	

Table 266. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1), 0x4006 C004 (USART2)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
17	-		Reserved. Read value is undefined, only zero should be written.	
20:18	RXIDLETOCFG		RX IDLE time out configuration Configures the number of idle characters that must be received before the RXIDLETO flag is set. An idle character is a character whose entire frame are all 1's, including start bit, data bits and stop bits (i.e. no activity on receiver line) 000b - 1 idle character 001b - 2 idle characters 010b - 4 idle characters 011b - 8 idle characters 100b - 16 idle characters 101b - 32 idle characters 110b - 64 idle characters 111b - 128 idle characters	0
31:21	-		Reserved. Read value is undefined, only zero should be written.	NA

18.6.3 USART Status register

The STAT register primarily provides a complete set of USART status flags for software to read. Flags other than read-only flags may be cleared by writing ones to corresponding bits of STAT. Interrupt status flags that are read-only and cannot be cleared by software, can be masked using the INTENCLR register (see Table 269).

The error flags for received noise, parity error, framing error, and overrun are set immediately upon detection and remain set until cleared by software action in STAT.

Table 267. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1), 0x4006 C008 (USART2)) bit description

Bit	Symbol	Description	Reset value	Access [1]
0	RXRDY	Receiver Ready flag. When 1, indicates that data is available to be read from the receiver buffer. Cleared after a read of the RXDAT or RXDATSTAT registers.	0	RO
1	RXIDLE	Receiver Idle. When 0, indicates that the receiver is currently in the process of receiving data. When 1, indicates that the receiver is not currently in the process of receiving data.	1	RO
2	TXRDY	Transmitter Ready flag. When 1, this bit indicates that data may be written to the transmit buffer. Previous data may still be in the process of being transmitted. Cleared when data is written to TXDAT. Set when the data is moved from the transmit buffer to the transmit shift register.	1	RO
3	TXIDLE	Transmitter Idle. When 0, indicates that the transmitter is currently in the process of sending data.When 1, indicate that the transmitter is not currently in the process of sending data.	1	RO
4	CTS	This bit reflects the current state of the CTS signal, regardless of the setting of the CTSEN bit in the CFG register. This will be the value of the CTS input pin unless loopback mode is enabled.	NA	RO
5	DELTACTS	This bit is set when a change in the state is detected for the CTS flag above. This bit is cleared by software.	0	W1

Table 267. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1), 0x4006 C008 (USART2)) bit description

Bit	Symbol	Description	Reset value	Access [1]
6	TXDISSTAT	Transmitter Disabled Interrupt flag. When 1, this bit indicates that the USART transmitter is fully idle after being disabled via the TXDIS in the CTL register (TXDIS = 1).	0	RO
7	-	Reserved. Read value is undefined, only zero should be written.	NA	NA
8	OVERRUNINT	Overrun Error interrupt flag. This flag is set when a new character is received while the receiver buffer is still in use. If this occurs, the newly received character in the shift register is lost.	0	W1
9	-	Reserved. Read value is undefined, only zero should be written.	NA	NA
10	RXBRK	Received Break. This bit reflects the current state of the receiver break detection logic. It is set when the Un_RXD pin remains low for 16 bit times. Note that FRAMERRINT will also be set when this condition occurs because the stop bit(s) for the character would be missing. RXBRK is cleared when the Un_RXD pin goes high.	0	RO
11	DELTARXBRK	This bit is set when a change in the state of receiver break detection occurs. Cleared by software.	0	W1
12	START	This bit is set when a start is detected on the receiver input. Its purpose is primarily to allow wake-up from deep-sleep or power-down mode immediately when a start is detected. Cleared by software.	0	W1
13	FRAMERRINT	Framing Error interrupt flag. This flag is set when a character is received with a missing stop bit at the expected location. This could be an indication of a baud rate or configuration mismatch with the transmitting source.	0	W1
14	PARITYERRINT	Parity Error interrupt flag. This flag is set when a parity error is detected in a received character..	0	W1
15	RXNOISEINT	Received Noise interrupt flag. Three samples of received data are taken in order to determine the value of each received data bit, except in synchronous mode. This acts as a noise filter if one sample disagrees. This flag is set when a received data bit contains one disagreeing sample. This could indicate line noise, a baud rate or character format mismatch, or loss of synchronization during data reception.	0	W1
16	ABERR	Autobaud Error. An autobaud error can occur if the BRG counts to its limit before the end of the start bit that is being measured, essentially an autobaud time-out.	0	W1
17	RXIDLETO	RX IDLE Timeout flag. Set when the receiver has been idle for a certain period of time as specified by CTRL.RXIDLETOCFG. Writing 1 clears this bit and lowers the corresponding interrupt. Once cleared, it cannot be set again until after the receiver receives a new character (non-idle).	0	R/W1C
31:18	-	Reserved. Read value is undefined, only zero should be written.	NA	NA

[1] RO = Read-only, W1 = write 1 to clear.

18.6.4 USART Interrupt Enable read and set register

The INTENSET register is used to enable various USART interrupt sources. Enable bits in INTENSET are mapped in locations that correspond to the flags in the STAT register. The complete set of interrupt enables may be read from this register. Writing ones to implemented bits in this register causes those bits to be set. The INTENCLR register is used to clear bits in this register.

Table 268. USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1), 0x4006C00C (USART2)) bit description

Bit	Symbol	Description	Reset Value
0	RXRDYEN	When 1, enables an interrupt when there is a received character available to be read from the RXDAT register.	0
1	-	Reserved. Read value is undefined, only zero should be written.	NA
2	TXRDYEN	When 1, enables an interrupt when the TXDAT register is available to take another character to transmit.	0
3	TXIDLEEN	When 1, enables an interrupt when the transmitter becomes idle (TXIDLE = 1).	0
4	-	Reserved. Read value is undefined, only zero should be written.	NA
5	DELTACTSEN	When 1, enables an interrupt when there is a change in the state of the CTS input.	0
6	TXDISEN	When 1, enables an interrupt when the transmitter is fully disabled as indicated by the TXDISINT flag in STAT. See description of the TXDISINT bit for details.	0
7	-	Reserved. Read value is undefined, only zero should be written.	NA
8	OVERRUNEN	When 1, enables an interrupt when an overrun error occurred.	0
10:9	-	Reserved. Read value is undefined, only zero should be written.	NA
11	DELTARXBRKEN	When 1, enables an interrupt when a change of state has occurred in the detection of a received break condition (break condition asserted or deasserted).	0
12	STARTEN	When 1, enables an interrupt when a received start bit has been detected.	0
13	FRAMERREN	When 1, enables an interrupt when a framing error has been detected.	0
14	PARITYERREN	When 1, enables an interrupt when a parity error has been detected.	0
15	RXNOISEEN	When 1, enables an interrupt when noise is detected.	0
16	ABERREN	When 1, enables an interrupt when an autobaud error occurs.	0
17	RXIDLETOEN	RX IDLE timeout interrupt enable. Write 1 to set the bit. Writing 0 has no effect. 0: RX IDLE time out interrupt is disabled. 1: When STAT.RXIDLETO is set, interrupt is asserted.	
31:18	-	Reserved. Read value is undefined, only zero should be written.	NA

18.6.5 USART Interrupt Enable Clear register

The INTENCLR register is used to clear bits in the INTENSET register.

Table 269. USART Interrupt Enable clear register (INTENCLR, address 0x4006 4010 (USART0), 0x4006 8010 (USART1), 0x4006 C010 (USART2)) bit description

Bit	Symbol	Description	Reset Value
0	RXRDYCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
1	-	Reserved. Read value is undefined, only zero should be written.	NA
2	TXRDYCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
3	TXIDLECLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
4	-	Reserved. Read value is undefined, only zero should be written.	NA
5	DELTACTSCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
6	TXDISINTCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
7	-	Reserved. Read value is undefined, only zero should be written.	NA
8	OVERRUNCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
10:9	-	Reserved. Read value is undefined, only zero should be written.	NA
11	DELTARXBRKCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
12	STARTCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
13	FRAMERRCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
14	PARITYERRCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
15	RXNOISECLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
16	ABERRCLR	Writing 1 clears the corresponding bit in the INTENSET register.	0
17	RXIDLETOEN	Writing 1 to clears INTENSET[RXIDLETOEN]. Writing 0 has no effect. The access for this bit is write-only.	0
31:18	-	Reserved. Read value is undefined, only zero should be written.	NA

18.6.6 USART Receiver Data register

The RXDAT register contains the last character received before any overrun.

Remark: Reading this register changes the status flags in the RXDATSTAT register.

Table 270. USART Receiver Data register (RXDAT, address 0x4006 4014 (USART0), 0x4006 8014 (USART1), 0x4006 C014 (USART2)) bit description

Bit	Symbol	Description	Reset Value
8:0	RXDAT	The USART Receiver Data register contains the next received character. The number of bits that are relevant depends on the USART configuration settings.	0
31:9	-	Reserved, the value read from a reserved bit is not defined.	NA

18.6.7 USART Receiver Data with Status register

The RXDATSTAT register contains the next complete character to be read and its relevant status flags. This allows getting all information related to a received character with one 16-bit read, which may be especially useful when the DMA is used with the USART receiver.

Remark: Reading this register changes the status flags.

Table 271. USART Receiver Data with Status register (RXDATSTAT, address 0x4006 4018 (USART0), 0x4006 8018 (USART1), 0x4006 C018 (USART2)) bit description

Bit	Symbol	Description	Reset Value
8:0	RXDAT	The USART Receiver Data register contains the next received character. The number of bits that are relevant depends on the USART configuration settings.	0
12:9	-	Reserved, the value read from a reserved bit is not defined.	NA
13	FRAMERR	Framing Error status flag. This bit is valid when there is a character to be read in the RXDAT register and reflects the status of that character. This bit will set when the character in RXDAT was received with a missing stop bit at the expected location. This could be an indication of a baud rate or configuration mismatch with the transmitting source.	0
14	PARITYERR	Parity Error status flag. This bit is valid when there is a character to be read in the RXDAT register and reflects the status of that character. This bit will be set when a parity error is detected in a received character.	0
15	RXNOISE	Received Noise flag. See description of the RXNOISEINT bit in Table 267.	0
31:16	-	Reserved, the value read from a reserved bit is not defined.	NA

18.6.8 USART Transmitter Data Register

The TXDAT register is written in order to send data via the USART transmitter. That data will be transferred to the transmit shift register when it is available, and another character may then be written to TXDAT.

Table 272. USART Transmitter Data Register (TXDAT, address 0x4006 401C (USART0), 0x4006 801C (USART1), 0x4006 C01C (USART2)) bit description

Bit	Symbol	Description	Reset Value
8:0	TXDAT	Writing to the USART Transmit Data Register causes the data to be transmitted as soon as the transmit shift register is available and any conditions for transmitting data are met: CTS low (if CTSEN bit = 1), TXDIS bit = 0.	0
31:9	-	Reserved. Only zero should be written.	NA

18.6.9 USART Baud Rate Generator register

The Baud Rate Generator is a simple 16-bit integer divider controlled by the BRG register. The BRG register contains the value used to divide the base clock in order to produce the clock used for USART internal operations.

A 16-bit value allows producing standard baud rates from 300 baud and lower at the highest frequency of the device, up to 921,600 baud from a base clock as low as 14.7456 MHz.

Typically, the baud rate clock is 16 times the actual baud rate. This overclocking allows for centering the data sampling time within a bit cell, and for noise reduction and detection by taking three samples of incoming data.

Details on how to select the right values for BRG can be found in Section 18.7.1.

Remark: If software needs to change the baud rate, the following sequence should be used: 1) Make sure the USART is not currently sending or receiving data. 2) Disable the USART by writing a 0 to the Enable bit (0 may be written to the entire registers). 3) Write the new BRGVAL. 4) Write to the CFG register to set the Enable bit to 1.

Table 273. USART Baud Rate Generator register (BRG, address 0x4006 4020 (USART0), 0x4006 8020 (USART1), 0x4006 8020 (USART2)) bit description

Bit	Symbol	Description	Reset Value
15:0	BRGVAL	This value is used to divide the USART input clock to determine the baud rate, based on the input clock from the FRG. 0 = The FRG clock is used directly by the USART function. 1 = The FRG clock is divided by 2 before use by the USART function. 2 = The FRG clock is divided by 3 before use by the USART function. ... 0xFFFF = The FRG clock is divided by 65,536 before use by the USART function.	0
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

18.6.10 USART Interrupt Status register

The read-only INTSTAT register provides a view of those interrupt flags that are currently enabled. This can simplify software handling of interrupts. See Table 267 for detailed descriptions of the interrupt flags.

Table 274. USART Interrupt Status register (INTSTAT, address 0x4006 4024 (USART0), 0x4006 8024 (USART1), 0x4006 8024 (USART2)) bit description

Bit	Symbol	Description	Reset Value
0	RXRDY	Receiver Ready flag.	0
1	-	Reserved. Read value is undefined, only zero should be written.	NA
2	TXRDY	Transmitter Ready flag.	1
3	TXIDLE	Transmitter idle status.	1
4	-	Reserved. Read value is undefined, only zero should be written.	NA
5	DELTACTS	This bit is set when a change in the state of the CTS input is detected.	0
6	TXDISINT	Transmitter Disabled Interrupt flag.	0
7	-	Reserved. Read value is undefined, only zero should be written.	NA
8	OVERRUNINT	Overrun Error interrupt flag.	0
10:9	-	Reserved. Read value is undefined, only zero should be written.	NA
11	DELTARXBRK	This bit is set when a change in the state of receiver break detection occurs.	0
12	START	This bit is set when a start is detected on the receiver input.	0
13	FRAMERRINT	Framing Error interrupt flag.	0
14	PARITYERRINT	Parity Error interrupt flag.	0
15	RXNOISEINT	Received Noise interrupt flag.	0
16	ABERR	Autobaud Error flag.	0
17	RXIDLETOINT	RX IDLE timeout interrupt flag. The access of this bit is read-only.	0
31:18	-	Reserved. Read value is undefined, only zero should be written.	NA

18.6.11 USART Oversample selection register

The OSR register allows selection of oversampling in asynchronous modes. The oversample value is the number of BRG clocks used to receive one data bit. The default is industry standard 16x oversampling.

Changing the oversampling can sometimes allow better matching of baud rates in cases where the peripheral clock rate is not a multiple of 16 times the expected maximum baud rate. For all modes where the OSR setting is used, the UART receiver takes three consecutive samples of input data in the approximate middle of the bit time. Smaller values of OSR can make the sampling position within a data bit less accurate and may potentially cause more noise errors or incorrect data.

Table 275. USART Oversample selection register (OSR, address 0x4006 4028 (USART0), 0x4006 4028 (USART1), 0x4006 8028 (USART2)) bit description

Bit	Symbol	Description	Reset value
3:0	OSRVAL	Oversample Selection Value. 0 to 3 = not supported 0x4 = 5 peripheral clocks are used to transmit and receive each data bit. 0x5 = 6 peripheral clocks are used to transmit and receive each data bit. ... 0xF= 16 peripheral clocks are used to transmit and receive each data bit.	0xF
31:4	-	Reserved, the value read from a reserved bit is not defined.	NA

18.6.12 USART Address register

The ADDR register holds the address for hardware address matching in address detect mode with automatic address matching enabled.

Table 276. USART Address register (ADDR, address 0x4006 402C (USART0), 0x4006 802C (USART1), 0x4006 C02C (USART2)) bit description

Bit	Symbol	Description	Reset value
7:0	ADDRESS	8-bit address used with automatic address matching. Used when address detection is enabled (ADDRDET in CTL = 1) and automatic address matching is enabled (AUTOADDR in CFG = 1).	0
31:8	-	Reserved, the value read from a reserved bit is not defined.	NA

18.7 Functional description

18.7.1 Clocking and baud rates

In order to use the USART, clocking details must be defined such as setting up the BRG, and typically also setting up the FRG. See Figure 22.

18.7.1.1 Fractional Rate Generator (FRG)

The Fractional Rate Generator can be used to obtain more precise baud rates when the peripheral clock is not a good multiple of standard (or otherwise desirable) baud rates.

The FRG is typically set up to produce an integer multiple of the highest required baud rate, or a very close approximation. The BRG is then used to obtain the actual baud rate needed.

The FRG register controls the USART Fractional Rate Generator, which provides the base clock for the USART. The Fractional Rate Generator creates a lower rate output clock by suppressing selected input clocks. When not needed, the value of 0 can be set for the FRG, which will then not divide the input clock.

The FRG output clock is defined as the inputs clock divided by 1 + (MULT / 256), where MULT is in the range of 1 to 255. This allows producing an output clock that ranges from the input clock divided by 1+1/256 to 1+255/256 (just more than 1 to just less than 2). Any further division can be done specific to each USART block by the integer BRG divider contained in each USART.

The base clock produced by the FRG cannot be perfectly symmetrical, so the FRG distributes the output clocks as evenly as is practical. Since the USART normally uses 16x overclocking, the jitter in the fractional rate clock in these cases tends to disappear in the ultimate USART output.

Any of the USARTs on the LPC86x device can use either FRG0 or FRG1. See Table 100 “Fractional generator 0 divider value register (FRG0DIV, address 0x4004 80D0) bit description” and Table 101 “Fractional generator 0 multiplier value register (FRG0MULT, address 0x4004 80D4) bit description”.

For details see Section 18.3.1 “Configure the USART clock and baud rate”.

18.7.1.2 Baud Rate Generator (BRG)

The Baud Rate Generator (see Section 18.6.9) is used to divide the base clock to produce a rate 16 times the desired baud rate. Typically, standard baud rates can be generated by integer divides of higher baud rates.

18.7.1.3 Baud rate calculations

Base clock rates are 16x for asynchronous mode and 1x for synchronous mode.

18.7.2 DMA

A DMA request is provided for each USART direction, and can be used in lieu of interrupts for transferring data by configuring the DMA controller appropriately. The DMA controller provides an acknowledgement signal that clears the related request when it completes handling a that request. The transmitter DMA request is asserted when the transmitter can accept more data. The receiver DMA request is asserted when received data is available to be read.

When DMA is used to perform USART data transfers, other mechanisms can be used to generate interrupts when needed. For instance, completion of the configured DMA transfer can generate an interrupt from the DMA controller. Also, interrupts for special conditions, such as a received break, can still generate useful interrupts.

18.7.3 Synchronous mode

Remark: Synchronous mode transmit and receive operate at the incoming clock rate in slave mode and the BRG selected rate (not divided by 16) in master mode.

18.7.4 Flow control

The USART supports both hardware and software flow control.

18.7.4.1 Hardware flow control

The USART supports hardware flow control using RTS and/or CTS signalling. If RTS is configured to appear on a device pin so that it can be sent to an external device, it indicates to an external device the ability of the receiver to receive more data.

If connected to a pin, and if enabled to do so, the CTS input can allow an external device to throttle the USART transmitter.

Figure 24 shows an overview of RTS and CTS within the USART.

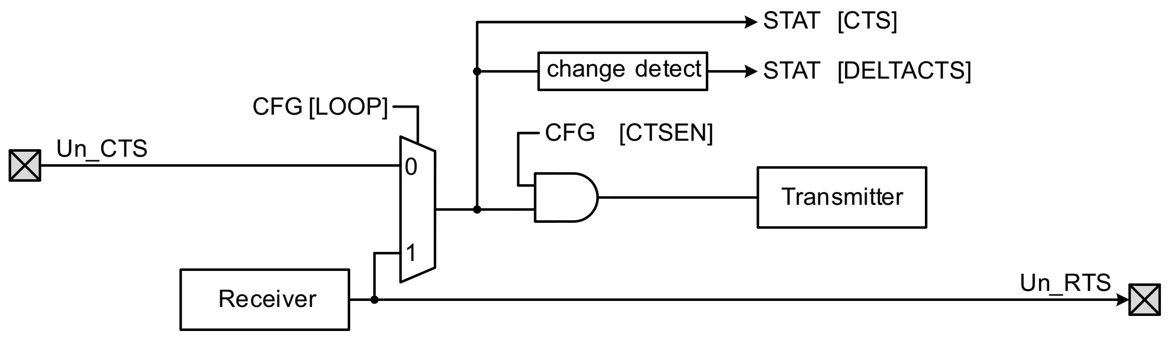


Fig 24. Hardware flow control using RTS and CTS

18.7.4.2 Software flow control

Software flow control could include XON / XOFF flow control, or other mechanisms. these are supported by the ability to check the current state of the CTS input, and/or have an interrupt when CTS changes state (via the CTS and DELTACTS bits, respectively, in the STAT register), and by the ability of software to gracefully turn off the transmitter (via the TXDIS bit in the CTL register).

18.7.5 Autobaud function

The autobaud functions attempts to measure the start bit time of the next received character. For this to work, the measured character must have a 1 in the least significant bit position, so that the start bit is bounded by a falling and rising edge. The measurement is made using the current clocking settings, including the oversampling configuration. The result is that a value is stored in the BRG register that is as close as possible to the correct setting for the sampled character and the current clocking settings. The sampled character is provided in the RXDAT and RXDATSTAT registers, allowing software to double-check for the expected character.

Autobaud includes a time-out that is flagged by ABERR if no character is received at the expected time. It is recommended that autobaud only be enabled when the USART receiver is idle. Once enabled, either RXRDY or ABERR will be asserted at some point. The assertion of RXRDY clears the AUTOBAUD bit automatically. The assertion of ABERR clears the AUTOBAUD bit once the receive line goes inactive.

Autobaud has no meaning, and should not be enabled, if the USART is in synchronous mode.

Remark: Before using autobaud, set the BRG register to 0x0 (this is the default). This setting allows the autobaud function to handle all baud rates.

18.7.6 RS-485 support

RS-485 support requires some form of address recognition and data direction control.

This USART has provisions for hardware address recognition (see the AUTOADDR bit in the CFG register in Section 18.6.1 and the ADDR register in Section 18.6.12), as well as software address recognition (see the ADDRDET bit in the CTL register in Section 18.6.2).

Automatic data direction control with the RTS pin can be set up using the OESEL, OEPOL, and OETA bits in the CFG register (Section 18.6.1). Data direction control can also be implemented in software using a GPIO pin.

18.7.7 Oversampling

Typical industry standard UARTs use a 16x oversample clock to transmit and receive asynchronous data. This is the number of BRG clocks used for one data bit. The Oversample Select Register (OSR) allows this UART to use a 16x down to a 5x oversample clock. There is no oversampling in synchronous modes.

Reducing the oversampling can sometimes help in getting better baud rate matching when the baud rate is very high, or the peripheral clock is very low. For example, the closest actual rate near 115,200 baud with a 12 MHz peripheral clock and 16x oversampling is 107,143 baud, giving a rate error of 7%. Changing the oversampling to 15x gets the actual rate to 114,286 baud, a rate error of 0.8%. Reducing the oversampling to 13x gets the actual rate to 115,385 baud, a rate error of only 0.16%.

There is a cost for altering the oversampling. In asynchronous modes, the UART takes three samples of incoming data on consecutive oversample clocks, as close to the center of a bit time as can be done. When the oversample rate is reduced, the three samples spread out and occupy a larger proportion of a bit time. For example, with 5x oversampling, there is one oversample clock, then three data samples taken, then one more oversample clock before the end of the bit time. Since the oversample clock is running asynchronously from the input data, skew of the input data relative to the expected timing has little room for error. At 16x oversampling, there are several oversample clocks before actual data sampling is done, making the sampling more robust. Generally speaking, it is recommended to use the highest oversampling where the rate error is acceptable in the system.

Chapter 19: SPI0/1

19.1 How to read this chapter

The SPI interfaces are available on all parts depending on the switch matrix configuration.

19.2 Features

- Data transmits of 1 to 16 bits supported directly. Larger frames supported by software.
- Master and slave operation.
- Data can be transmitted to a slave without the need to read incoming data. This can be useful while setting up an SPI memory.
- Control information can optionally be written along with data. This allows very versatile operation, including frames of arbitrary length.
- Up to four Slave Select input/outputs with selectable polarity and flexible usage.
- Supports DMA transfers: SPIn transmit and receive functions can operated with the system DMA controller.

Remark: Texas Instruments SSI and National Microwire modes are not supported.

19.3 Basic configuration

Configure SPI0/1 using the following registers:

- In the SYSAHBCLKCTRL register, set bit 11 and 12 (Table 94) to enable the clock to the register interface.
- Clear the SPI0/1 peripheral resets using the PRESETCTRL register (Table 95).
- Enable/disable the SPI0/1 interrupts in interrupt slots #0 and 1 in the NVIC.
- Configure the SPI0/1 pin functions through the switch matrix. See Section 19.4.
- Using the SPI0CLKSEL register, the peripheral clock sources for the SPI can be FRO, main_clk, frg0clk, frg1clk, and fro_div. (see Figure 7 “Clock generation”).

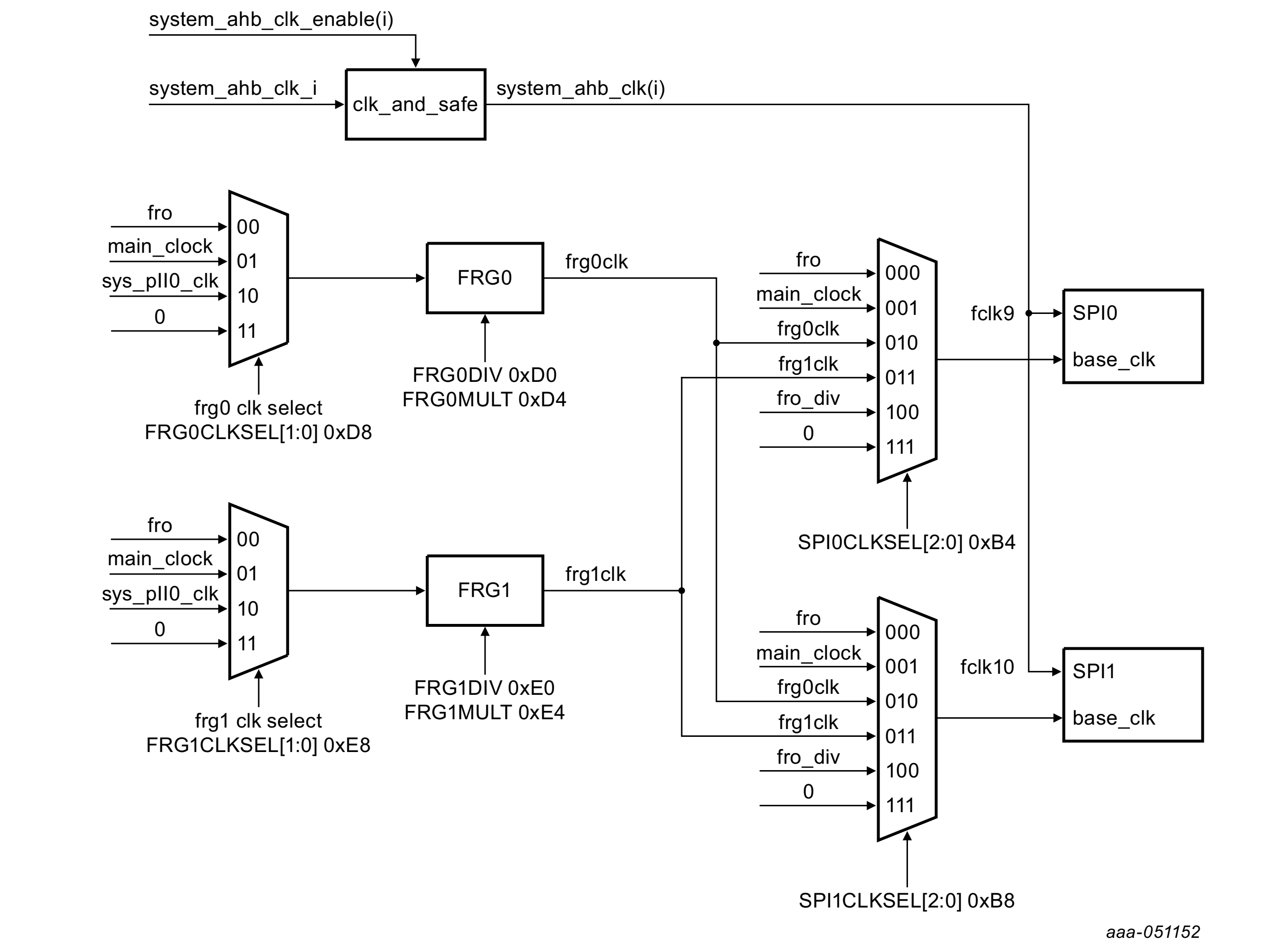


Fig 25. SPI clocking

19.3.1 Configure the SPI for wake-up

In sleep mode, any signal that triggers an SPI interrupt can wake up the part, provided that the interrupt is enabled in the INTENSET register and the NVIC. As long as the SPI clock SPI_PCLK remains active in sleep mode, the SPI can wake up the part independently of whether the SPI block is configured in master or slave mode.

In Deep-sleep or Power-down mode, the SPI clock is turned off as are all peripheral clocks. However, if the SPI is configured in slave mode and an external master on the provides the clock signal, the SPI can create an interrupt asynchronously. This interrupt, if enabled in the NVIC and in the SPI's INTENSET register, can then wake up the core.

19.3.1.1 Wake-up from Sleep mode

- Configure the SPI in either master or slave mode. See Table 279.
- Enable the SPI interrupt in the NVIC.
- Any SPI interrupt wakes up the part from sleep mode. Enable the SPI interrupt in the INTENSET register (Table 282).

19.3.1.2 Wake-up from Deep-sleep or Power-down mode

- Configure the SPI in slave mode. See Table 279. You must connect the SCK function to a pin and connect the pin to the master.
- Enable the SPI interrupt in the STARTERP1 register. See Table 115 “Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description”.
- In the PDAWAKE register, configure all peripherals that need to be running when the part wakes up.
- Enable the SPI interrupt in the NVIC.
- Enable the interrupt in the INTENSET register which configures the interrupt as wake-up event (Table 282). Examples are the following wake-up events:
  – A change in the state of the SSEL pins.
  – Data available to be received.
  – Receiver overrun.

19.4 Pin description

The SPI signals are movable functions and are assigned to external pins through the switch matrix.

See Section 11.3.1 “Connect an internal signal to a package pin” to assign the SPI functions to pins on the part.

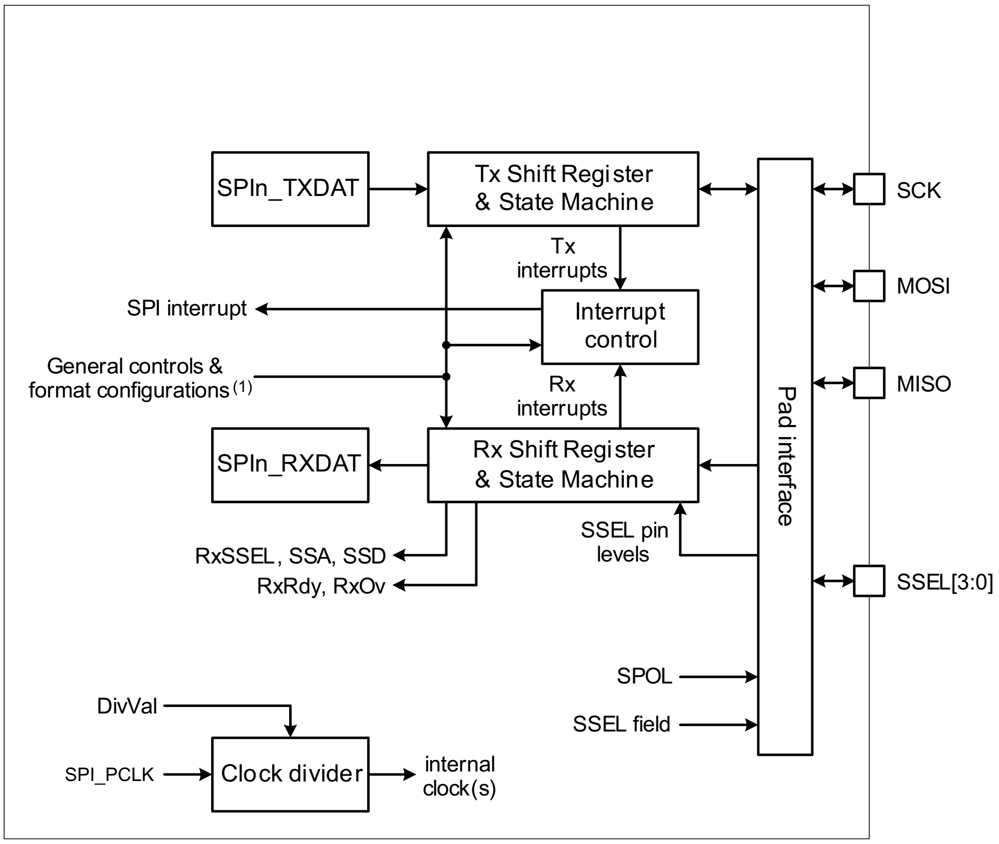
Table 277. SPI Pin Description

Function	I/O	Type	Connect to	Use register	Reference	Description
SPI0_SCK	I/O	external to pin	any pin	PINASSIGN3	Table 131	Serial Clock. SCK is a clock signal used to synchronize the transfer of data. It is driven by the master and received by the slave. When the SPI interface is used, the clock is programmable to be active-high or active-low. SCK only switches during a data transfer. It is driven whenever the Master bit in CFG equals 1, regardless of the state of the Enable bit.
SPI0_MOSI	I/O	external to pin	any pin	PINASSIGN4	Table 132	Master Out Slave In. The MOSI signal transfers serial data from the master to the slave. When the SPI is a master, it outputs serial data on this signal. When the SPI is a slave, it clocks in serial data from this signal. MOSI is driven whenever the Master bit in SPInCfg equals 1, regardless of the state of the Enable bit.
SPI0_MISO	I/O	external to pin	any pin	PINASSIGN4	Table 132	Master In Slave Out. The MISO signal transfers serial data from the slave to the master. When the SPI is a master, serial data is input from this signal. When the SPI is a slave, serial data is output to this signal. MISO is driven when the SPI block is enabled, the Master bit in CFG equals 0, and when the slave is selected by one or more SSEL signals.

Table 277. SPI Pin Description

Function	I/O	Type	Connect to	Use register	Reference	Description
SPI0_SSEL0	I/O	external to pin	any pin	PINASSIGN4	Table 132	Slave Select 0. When the SPI interface is a master, it will drive the SSEL signals to an active state before the start of serial data and then release them to an inactive state after the serial data has been sent. By default, this signal is active low but can be selected to operate as active high. When the SPI is a slave, any SSEL in an active state indicates that this slave is being addressed. The SSEL pin is driven whenever the Master bit in the CFG register equals 1, regardless of the state of the Enable bit.
SPI0_SSEL1	I/O	external to pin	any pin	PINASSIGN4	Table 132	Slave Select 1.
SPI0_SSEL2	I/O	external to pin	any pin	PINASSIGN5	Table 133	Slave Select 2.
SPI0_SSEL3	I/O	external to pin	any pin	PINASSIGN5	Table 133	Slave Select 3.
SPI1_SCK	I/O	external to pin	any pin	PINASSIGN5	Table 133	Serial Clock.
SPI1_MOSI	I/O	external to pin	any pin	PINASSIGN5	Table 133	Master Out Slave In.
SPI1_MISO	I/O	external to pin	any pin	PINASSIGN6	Table 134	Master In Slave Out.
SPI1_SSEL0	I/O	external to pin	any pin	PINASSIGN6	Table 134	Slave Select 0.
SPI1_SSEL1	I/O	external to pin	any pin	PINASSIGN6	Table 134	Slave Select 1.

19.5 General description



(1) Includes CPOL, CPHA, LSBF, FLEN, master, enable, transfer_delay, frame_delay, pre_delay, post_delay, SOT, EOT, EOF, RXIGNORE, individual interrupt enables.

Fig 26. SPI block diagram

19.6 Register description

The Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

Table 278. Register overview: SPI (base address 0x4005 8000 (SPI0) and 0x4005 C000 (SPI1))

Name	Access	Offset	Description	Reset value	Reference
CFG	R/W	0x000	SPI Configuration register	0	Table 279
DLY	R/W	0x004	SPI Delay register	0	Table 280
STAT	R/W	0x008	SPI Status. Some status flags can be cleared by writing a 1 to that bit position	0x0102	Table 281

Table 278. Register overview: SPI (base address 0x4005 8000 (SPI0) and 0x4005 C000 (SPI1)) *…continued*

Name	Access	Offset	Description	Reset value	Reference
INTENSET	R/W	0x00C	SPI Interrupt Enable read and Set. A complete value may be read from this register. Writing a 1 to any implemented bit position causes that bit to be set.	0	Table 282
INTENCLR	W	0x010	SPI Interrupt Enable Clear. Writing a 1 to any implemented bit position causes the corresponding bit in INTENSET to be cleared.	NA	Table 283
RXDAT	R	0x014	SPI Receive Data	NA	Table 284
TXDATCTL	R/W	0x018	SPI Transmit Data with Control	0	Table 285
TXDAT	R/W	0x01C	SPI Transmit Data	0	Table 286
TXCTL	R/W	0x020	SPI Transmit Control	0	Table 287
DIV	R/W	0x024	SPI clock Divider	0	Table 288
INTSTAT	R	0x028	SPI Interrupt Status	0x02	Table 289

19.6.1 SPI Configuration register

The CFG register contains information for the general configuration of the SPI. Typically, this information is not changed during operation. Some configurations, such as CPOL, CPHA, and LSBF should not be made while the SPI is not fully idle. See the description of the master idle status (MSTIDLE in Table 281) for more information.

Remark: If the interface is re-configured from Master mode to Slave mode or the reverse (an unusual case), the SPI should be disabled and re-enabled with the new configuration.

Table 279. SPI Configuration register (CFG, addresses 0x4005 8000 (SPI0), 0x4005 C000 (SPI1)) bit description

Bit	Symbol	Value	Description	Reset value
0	ENABLE		SPI enable.	0
		0	Disabled. The SPI is disabled and the internal state machine and counters are reset.	
		1	Enabled. The SPI is enabled for operation.	
1	-		Reserved. Read value is undefined, only zero should be written.	NA
2	MASTER		Master mode select.	0
		0	Slave mode. The SPI will operate in slave mode. SCK, MOSI, and the SSEL signals are inputs, MISO is an output.	
		1	Master mode. The SPI will operate in master mode. SCK, MOSI, and the SSEL signals are outputs, MISO is an input.	
3	LSBF		LSB First mode enable.	0
		0	Standard. Data is transmitted and received in standard MSB first order.	
		1	Reverse. Data is transmitted and received in reverse order (LSB first).	

Table 279. SPI Configuration register (CFG, addresses 0x4005 8000 (SPI0), 0x4005 C000 (SPI1)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
4	CPHA		Clock Phase select.	0
		0	Change. The SPI captures serial data on the first clock transition of the transfer (when the clock changes away from the rest state). Data is changed on the following edge.	
		1	Capture. The SPI changes serial data on the first clock transition of the transfer (when the clock changes away from the rest state). Data is captured on the following edge.	
5	CPOL		Clock Polarity select.	0
		0	Low. The rest state of the clock (between transfers) is low.	
		1	High. The rest state of the clock (between transfers) is high.	
6	-		Reserved. Read value is undefined, only zero should be written.	NA
7	LOOP		Loopback mode enable. Loopback mode applies only to Master mode, and connects transmit and receive data connected together to allow simple software testing.	0
		0	Disabled.	
		1	Enabled.	
8	SPOL0		SSEL0 Polarity select.	0
		0	Low. The SSEL0 pin is active low. The value in the SSEL0 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL0 is not inverted relative to the pins.	
		1	High. The SSEL0 pin is active high. The value in the SSEL0 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL0 is inverted relative to the pins.	
9	SPOL1		SSEL1 Polarity select.	0
		0	Low. The SSEL1 pin is active low. The value in the SSEL1 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL1 is not inverted relative to the pins.	
		1	High. The SSEL1 pin is active high. The value in the SSEL1 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL1 is inverted relative to the pins.	
10	SPOL2		SSEL2 Polarity select.	0
		0	Low. The SSEL2 pin is active low. The value in the SSEL2 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL2 is not inverted relative to the pins.	
		1	High. The SSEL2 pin is active high. The value in the SSEL2 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL2 is inverted relative to the pins.	
11	SPOL3		SSEL3 Polarity select.	0
		0	Low. The SSEL3 pin is active low. The value in the SSEL3 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL3 is not inverted relative to the pins.	
		1	High. The SSEL3 pin is active high. The value in the SSEL3 fields of the RXDAT, TXDATCTL, and TXCTL registers related to SSEL3 is inverted relative to the pins.	
31:12	-		Reserved. Read value is undefined, only zero should be written.	NA

19.6.2 SPI Delay register

The DLY register controls several programmable delays related to SPI signalling. These delays apply only to master mode, and are all stated in SPI clocks.

Timing details are shown in:

Section 19.7.2.1 “Pre_delay and Post_delay”

Section 19.7.2.2 “Frame_delay”

Section 19.7.2.3 “Transfer_delay”

Table 280. SPI Delay register (DLY, addresses 0x4005 8004 (SPI0), 0x4005 C004 (SPI1)) bit description

Bit	Symbol	Description	Reset value
3:0	PRE_DELAY	Controls the amount of time between SSEL assertion and the beginning of a data transfer. There is always one SPI clock time between SSEL assertion and the first clock edge. This is not considered part of the pre-delay. 0x0 = No additional time is inserted. 0x1 = 1 SPI clock time is inserted. 0x2 = 2 SPI clock times are inserted. ... 0xF = 15 SPI clock times are inserted.	0
7:4	POST_DELAY	Controls the amount of time between the end of a data transfer and SSEL deassertion. 0x0 = No additional time is inserted. 0x1 = 1 SPI clock time is inserted. 0x2 = 2 SPI clock times are inserted. ... 0xF = 15 SPI clock times are inserted.	0
11:8	FRAME_DELAY	If the EOF flag is set, controls the minimum amount of time between the current frame and the next frame (or SSEL deassertion if EOT). 0x0 = No additional time is inserted. 0x1 = 1 SPI clock time is inserted. 0x2 = 2 SPI clock times are inserted. ... 0xF = 15 SPI clock times are inserted.	0
15:12	TRANSFER_DELAY	Controls the minimum amount of time that the SSEL is deasserted between transfers. 0x0 = The minimum time that SSEL is deasserted is 1 SPI clock time. (Zero added time.) 0x1 = The minimum time that SSEL is deasserted is 2 SPI clock times. 0x2 = The minimum time that SSEL is deasserted is 3 SPI clock times. ... 0xF = The minimum time that SSEL is deasserted is 16 SPI clock times.	0
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

19.6.3 SPI Status register

The STAT register provides SPI status flags for software to read, and a control bit for forcing an end of transfer. Flags other than read-only flags may be cleared by writing ones to corresponding bits of STAT.

STAT contains 2 error flags (in slave mode only): RXOV and TXUR. These are receiver overrun and transmit underrun, respectively. If either of these errors occur during operation, the SPI should be disabled, then re-enabled in order to make sure all internal states are cleared before attempting to resume operation.

In this register, the following notation is used: RO = Read-only, W1 = write 1 to clear.

Table 281. SPI Status register (STAT, addresses 0x4005 8008 (SPI0), 0x4005 C008 (SPI1)) bit description

Bit	Symbol	Description	Reset value	Access [1]
0	RXRDY	Receiver Ready flag. When 1, indicates that data is available to be read from the receiver buffer. Cleared after a read of the RXDAT register.	0	RO
1	TXRDY	Transmitter Ready flag. When 1, this bit indicates that data may be written to the transmit buffer. Previous data may still be in the process of being transmitted. Cleared when data is written to TXDAT or TXDATCTL until the data is moved to the transmit shift register.	1	RO
2	RXOV	Receiver Overrun interrupt flag. This flag applies only to slave mode (Master = 0). This flag is set when the beginning of a received character is detected while the receiver buffer is still in use. If this occurs, the receiver buffer contents are preserved, and the incoming data is lost. Data received by the SPI should be considered undefined if RxOv is set.	0	W1
3	TXUR	Transmitter Underrun interrupt flag. This flag applies only to slave mode (Master = 0). In this case, the transmitter must begin sending new data on the next input clock if the transmitter is idle. If that data is not available in the transmitter holding register at that point, there is no data to transmit and the TXUR flag is set. Data transmitted by the SPI should be considered undefined if TXUR is set.	0	W1
4	SSA	Slave Select Assert. This flag is set whenever any slave select transitions from deasserted to asserted, in both master and slave modes. This allows determining when the SPI transmit/receive functions become busy, and allows waking up the device from reduced power modes when a slave mode access begins. This flag is cleared by software.	0	W1
5	SSD	Slave Select Deassert. This flag is set whenever any asserted slave selects transition to deasserted, in both master and slave modes. This allows determining when the SPI transmit/receive functions become idle. This flag is cleared by software.	0	W1
6	STALLED	Stalled status flag. This indicates whether the SPI is currently in a stall condition.	0	RO

Table 281. SPI Status register (STAT, addresses 0x4005 8008 (SPI0), 0x4005 C008 (SPI1)) bit description

Bit	Symbol	Description	Reset value	Access [1]
7	ENDTRANSFER	End Transfer control bit. Software can set this bit to force an end to the current transfer when the transmitter finishes any activity already in progress, as if the EOT flag had been set prior to the last transmission. This capability is included to support cases where it is not known when transmit data is written that it will be the end of a transfer. The bit is cleared when the transmitter becomes idle as the transfer comes to an end. Forcing an end of transfer in this manner causes any specified FRAME_DELAY and TRANSFER_DELAY to be inserted.	0	RO/W1
8	MSTIDLE	Master idle status flag. This bit is 1 whenever the SPI master function is fully idle. This means that the transmit holding register is empty and the transmitter is not in the process of sending data.	1	RO
31:9	-	Reserved. Read value is undefined, only zero should be written.	NA	NA

[1] RO = Read-only, W1 = write 1 to clear.

19.6.4 SPI Interrupt Enable read and Set register

The INTENSET register is used to enable various SPI interrupt sources. Enable bits in INTENSET are mapped in locations that correspond to the flags in the STAT register. The complete set of interrupt enables may be read from this register. Writing ones to implemented bits in this register causes those bits to be set. The INTENCLR register is used to clear bits in this register. See Table 281 for details of the interrupts.

Table 282. SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI0), 0x4005 C00C (SPI1)) bit description

Bit	Symbol	Value	Description	Reset value
0	RXRDYEN		Determines whether an interrupt occurs when receiver data is available.	0
		0	No interrupt will be generated when receiver data is available.	
		1	An interrupt will be generated when receiver data is available in the RXDAT register.	
1	TXRDYEN		Determines whether an interrupt occurs when the transmitter holding register is available.	0
		0	No interrupt will be generated when the transmitter holding register is available.	
		1	An interrupt will be generated when data may be written to TXDAT.	
2	RXOVEN		Determines whether an interrupt occurs when a receiver overrun occurs. This happens in slave mode when there is a need for the receiver to move newly received data to the RXDAT register when it is already in use. The interface prevents receiver overrun in Master mode by not allowing a new transmission to begin when a receiver overrun would otherwise occur.	0
		0	No interrupt will be generated when a receiver overrun occurs.	
		1	An interrupt will be generated if a receiver overrun occurs.	
3	TXUREN		Determines whether an interrupt occurs when a transmitter underrun occurs. This happens in slave mode when there is a need to transmit data when none is available.	0
		0	No interrupt will be generated when the transmitter underruns.	
		1	An interrupt will be generated if the transmitter underruns.	

Table 282. SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI0), 0x4005 C00C (SPI1)) bit description

Bit	Symbol	Value	Description	Reset value
4	SSAEN		Determines whether an interrupt occurs when the Slave Select is asserted.	0
		0	No interrupt will be generated when any Slave Select transitions from deasserted to asserted.	
		1	An interrupt will be generated when any Slave Select transitions from deasserted to asserted.	
5	SSDEN		Determines whether an interrupt occurs when the Slave Select is deasserted.	0
		0	No interrupt will be generated when all asserted Slave Selects transition to deasserted.	
		1	An interrupt will be generated when all asserted Slave Selects transition to deasserted.	
31:6	-		Reserved. Read value is undefined, only zero should be written.	NA

19.6.5 SPI Interrupt Enable Clear register

The INTENCLR register is used to clear interrupt enable bits in the INTENSET register.

Table 283. SPI Interrupt Enable clear register (INTENCLR, addresses 0x4005 8010 (SPI0), 0x4005 C010 (SPI1)) bit description

Bit	Symbol	Description	Reset value
0	RXRDYEN	Writing 1 clears the corresponding bits in the INTENSET register.	0
1	TXRDYEN	Writing 1 clears the corresponding bits in the INTENSET register.	0
2	RXOVEN	Writing 1 clears the corresponding bits in the INTENSET register.	0
3	TXUREN	Writing 1 clears the corresponding bits in the INTENSET register.	0
4	SSAEN	Writing 1 clears the corresponding bits in the INTENSET register.	0
5	SSDEN	Writing 1 clears the corresponding bits in the INTENSET register.	0
31:6	-	Reserved. Read value is undefined, only zero should be written.	NA

19.6.6 SPI Receiver Data register

The read-only RXDAT register provides the means to read the most recently received data. The value of SSEL can be read along with the data.

For details on the slave select process, see Section 19.7.4.

Table 284. SPI Receiver Data register (RXDAT, addresses 0x4005 8014 (SPI0), 0x4005 C014 (SPI1)) bit description

Bit	Symbol	Description	Reset value
15:0	RXDAT	Receiver Data. This contains the next piece of received data. The number of bits that are used depends on the LEN setting in TXCTL / TXDATCTL.	undefined
16	RXSSEL0_N	Slave Select for receive. This field allows the state of the SSEL0 pin to be saved along with received data. The value will reflect the SSEL0 pin for both master and slave operation. A zero indicates that a slave select is active. The actual polarity of each slave select pin is configured by the related SPOL bit in CFG.	undefined
17	RXSSEL1_N	Slave Select for receive. This field allows the state of the SSEL1 pin to be saved along with received data. The value will reflect the SSEL1 pin for both master and slave operation. A zero indicates that a slave select is active. The actual polarity of each slave select pin is configured by the related SPOL bit in CFG.	undefined
18	RXSSEL2_N	Slave Select for receive. This field allows the state of the SSEL2 pin to be saved along with received data. The value will reflect the SSEL2 pin for both master and slave operation. A zero indicates that a slave select is active. The actual polarity of each slave select pin is configured by the related SPOL bit in CFG.	undefined
19	TXSSEL3_N	Slave Select for receive. This field allows the state of the SSEL3 pin to be saved along with received data. The value will reflect the SSEL3 pin for both master and slave operation. A zero indicates that a slave select is active. The actual polarity of each slave select pin is configured by the related SPOL bit in CFG.	undefined
20	SOT	Start of Transfer flag. This flag will be 1 if this is the first data after the SSELs went from deasserted to asserted (i.e., any previous transfer has ended). This information can be used to identify the first piece of data in cases where the transfer length is greater than 16 bit.	
31:21	-	Reserved, the value read from a reserved bit is not defined.	NA

19.6.7 SPI Transmitter Data and Control register

The TXDATCTL register provides a location where both transmit data and control information can be written simultaneously. This allows detailed control of the SPI without a separate write of control information for each piece of data, which can be especially useful when the SPI is used with DMA.

Remark: The SPI has no receiver control registers. Hence software needs to set the data length in the transmitter control or transmitter data and control register first in order to handle reception with correct data length. The programmed data length becomes active only when data is actually transmitted. Therefore, this must be done before any data can be received.

When control information remains static during transmit, the TXDAT register should be used (see Section 19.6.8) instead of the TXDATCTL register. Control information can then be written separately via the TXCTL register (see Section 19.6.9). The upper part of TXDATCTL (bits 27 to 16) are the same bits contained in the TXCTL register. The two registers simply provide two ways to access them.

For details on the slave select process, see Section 19.7.4.

For details on using multiple consecutive data transmits for transfer lengths larger than 16 bit, see Section 19.7.6 “Data lengths greater than 16 bits”.

Table 285. SPI Transmitter Data and Control register (TXDATCTL, addresses 0x4005 8018 (SPI0), 0x4005 C018 (SPI1)) bit description

Bit	Symbol	Value	Description	Reset value
15:0	TXDAT		Transmit Data. This field provides from 1 to 16 bits of data to be transmitted.	0
16	TXSSEL0_N		Transmit Slave Select. This field asserts SSEL0 in master mode. The output on the pin is active LOW by default. **Remark:** The active state of the SSEL0 pin is configured by bits in the CFG register.	0
		0	SSEL0 asserted.	
		1	SSEL0 not asserted.	
17	TXSSEL1_N		Transmit Slave Select. This field asserts SSEL1 in master mode. The output on the pin is active LOW by default. **Remark:** The active state of the SSEL1 pin is configured by bits in the CFG register.	0
		0	SSEL1 asserted.	
		1	SSEL1 not asserted.	
18	TXSSEL2_N		Transmit Slave Select. This field asserts SSEL2 in master mode. The output on the pin is active LOW by default. **Remark:** The active state of the SSEL2 pin is configured by bits in the CFG register.	0
		0	SSEL2 asserted.	
		1	SSEL2 not asserted.	
19	TXSSEL3_N		Transmit Slave Select. This field asserts SSEL3 in master mode. The output on the pin is active LOW by default. **Remark:** The active state of the SSEL3 pin is configured by bits in the CFG register.	0
		0	SSEL3 asserted.	
		1	SSEL3 not asserted.	

Table 285. SPI Transmitter Data and Control register (TXDATCTL, addresses 0x4005 8018 (SPI0), 0x4005 C018 (SPI1)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
20	EOT		End of Transfer. The asserted SSEL will be deasserted at the end of a transfer, and remain so for at least the time specified by the Transfer_delay value in the DLY register.	0
		0	SSEL not deasserted. This piece of data is not treated as the end of a transfer. SSEL will not be deasserted at the end of this data.	
		1	SSEL deasserted. This piece of data is treated as the end of a transfer. SSEL will be deasserted at the end of this piece of data.	
21	EOF		End of Frame. Between frames, a delay may be inserted, as defined by the FRAME_DELAY value in the DLY register. The end of a frame may not be particularly meaningful if the FRAME_DELAY value = 0. This control can be used as part of the support for frame lengths greater than 16 bits.	0
		0	Data not EOF. This piece of data transmitted is not treated as the end of a frame.	
		1	Data EOF. This piece of data is treated as the end of a frame, causing the FRAME_DELAY time to be inserted before subsequent data is transmitted.	
22	RXIGNORE		Receive Ignore. This allows data to be transmitted using the SPI without the need to read unneeded data from the receiver.Setting this bit simplifies the transmit process and can be used with the DMA.	0
		0	Read received data. Received data must be read in order to allow transmission to progress. In slave mode, an overrun error will occur if received data is not read before new data is received.	
		1	Ignore received data. Received data is ignored, allowing transmission without reading unneeded received data. No receiver flags are generated.	
23	-		Reserved. Read value is undefined, only zero should be written.	NA
27:24	LEN		Data Length. Specifies the data length from 1 to 16 bits. Note that transfer lengths greater than 16 bits are supported by implementing multiple sequential transmits. 0x0 = Data transfer is 1 bit in length. 0x1 = Data transfer is 2 bits in length. 0x2 = Data transfer is 3 bits in length. ... 0xF = Data transfer is 16 bits in length.	0x0
31:28	-		Reserved. Read value is undefined, only zero should be written.	NA

19.6.8 SPI Transmitter Data Register

The TXDAT register is written in order to send data via the SPI transmitter when control information is not changing during the transfer (see Section 19.6.7). That data will be sent to the transmit shift register when it is available, and another character may then be written to TXDAT.

Table 286. SPI Transmitter Data Register (TXDAT, addresses 0x4005 801C (SPI0), 0x4005 C01C (SPI1)) bit description

Bit	Symbol	Description	Reset value
15:0	DATA	Transmit Data. This field provides from 4 to 16 bits of data to be transmitted.	0
31:16	-	Reserved. Only zero should be written.	NA

19.6.9 SPI Transmitter Control register

The TXCTL register provides a way to separately access control information for the SPI. These bits are another view of the same-named bits in the TXDATCTL register (see Section 19.6.7). Changing bits in TXCTL has no effect unless data is later written to the TXDAT register. Data written to TXDATCTL overwrites the TXCTL register.

When control information needs to be changed during transmission, the TXDATCTL register should be used (see Section 19.6.7) instead of TXDAT. Control information can then be written along with data.

Table 287. SPI Transmitter Control register (TXCTL, addresses 0x4005 8020 (SPI0), 0x4005 C020 (SPI1)) bit description

Bit	Symbol	Description	Reset value
15:0	-	Reserved. Read value is undefined, only zero should be written.	NA
16	TXSSEL0_N	Transmit Slave Select 0.	0x0
17	TXSSEL1_N	Transmit Slave Select 1.	0x0
18	TXSSEL2_N	Transmit Slave Select 2.	0x0
19	TXSSEL3_n	Transmit Slave Select 3.	0x0
20	EOT	End of Transfer.	0
21	EOF	End of Frame.	0
22	RXIGNORE	Receive Ignore.	0
23	-	Reserved. Read value is undefined, only zero should be written.	NA
27:24	LEN	Data transfer Length.	0x0
31:28	-	Reserved. Read value is undefined, only zero should be written.	NA

19.6.10 SPI Divider register

The DIV register determines the clock used by the SPI in master mode.

For details on clocking, see Section 19.7.3 “Clocking and data rates”.

Table 288. SPI Divider register (DIV, addresses 0x4005 8024 (SPI0), 0x4005 C024 (SPI1)) bit description

Bit	Symbol	Description	Reset Value
15:0	DIVVAL	Rate divider value. Specifies how the PCLK for the SPI is divided to produce the SPI clock rate in master mode. DIVVAL is -1 encoded such that the value 0 results in PCLK/1, the value 1 results in PCLK/2, up to the maximum possible divide value of 0xFFFF, which results in PCLK/65536.	0
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

19.6.11 SPI Interrupt Status register

The read-only INTSTAT register provides a view of those interrupt flags that are currently enabled. This can simplify software handling of interrupts. See Table 281 for detailed descriptions of the interrupt flags.

Table 289. SPI Interrupt Status register (INTSTAT, addresses 0x4005 8028 (SPI0), 0x4005 C028 (SPI1)) bit description

Bit	Symbol	Description	Reset value
0	RXRDY	Receiver Ready flag.	0
1	TXRDY	Transmitter Ready flag.	1
2	RXOV	Receiver Overrun interrupt flag.	0
3	TXUR	Transmitter Underrun interrupt flag.	0
4	SSA	Slave Select Assert.	0
5	SSD	Slave Select Deassert.	0
31:6	-	Reserved. Read value is undefined, only zero should be written.	NA

19.7 Functional description

19.7.1 Operating modes: clock and phase selection

SPI interfaces typically allow configuration of clock phase and polarity. These are sometimes referred to as numbered SPI modes, as described in Table 290 and shown in Figure 27. CPOL and CPHA are configured by bits in the CFG register (Section 19.6.1).

Table 290. SPI mode summary

CPOL	CPHA	SPI Mode	Description	SCK rest state	SCK data change edge	SCK data sample edge
0	0	0	The SPI captures serial data on the first clock transition of the transfer (when the clock changes away from the rest state). Data is changed on the following edge.	low	falling	rising
0	1	1	The SPI changes serial data on the first clock transition of the transfer (when the clock changes away from the rest state). Data is captured on the following edge.	low	rising	falling
1	0	2	Same as mode 0 with SCK inverted.	high	rising	falling
1	1	3	Same as mode 1 with SCK inverted.	high	falling	rising

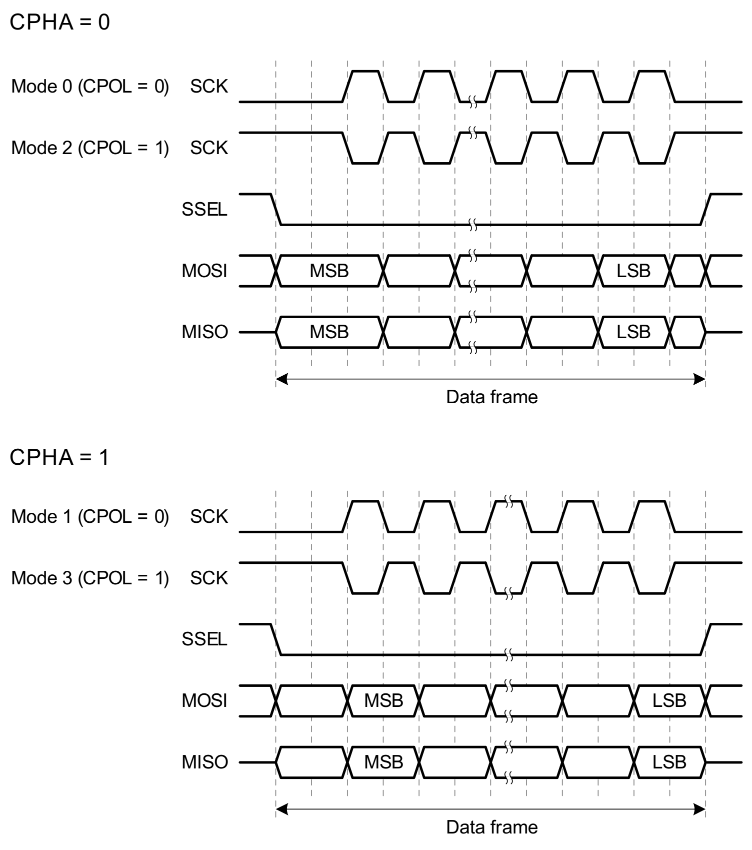


Fig 27. Basic SPI operating modes

19.7.2 Frame delays

Several delays can be specified for SPI frames. These include:

- Pre_delay: delay after SSEL is asserted before data clocking begins
- Post_delay: delay at the end of a data frame before SSEL is de-asserted
- Frame_delay: delay between data frames when SSEL is not de-asserted
- Transfer_delay: minimum duration of SSEL in the de-asserted state between transfers

19.7.2.1 Pre_delay and Post_delay

Pre_delay and Post_delay are illustrated by the examples in Figure 28. The Pre_delay value controls the amount of time between SSEL being asserted and the beginning of the subsequent data frame. The Post_delay value controls the amount of time between the end of a data frame and the de-assertion of SSEL.

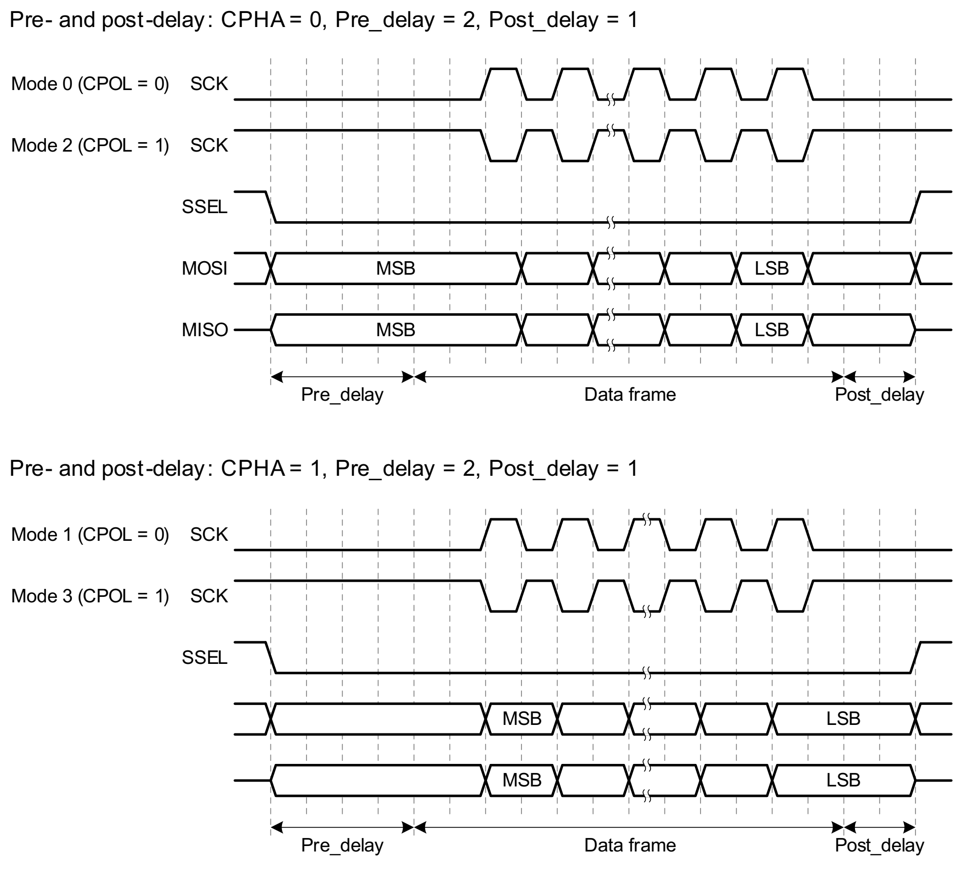


Fig 28. Pre_delay and Post_delay

19.7.2.2 Frame_delay

The Frame_delay value controls the amount of time at the end of each frame. This delay is inserted when the EOF bit = 1. Frame_delay is illustrated by the examples in Figure 29. Note that frame boundaries occur only where specified. This is because frame lengths can be any size, involving multiple data writes. See Section 19.7.6 for more information.

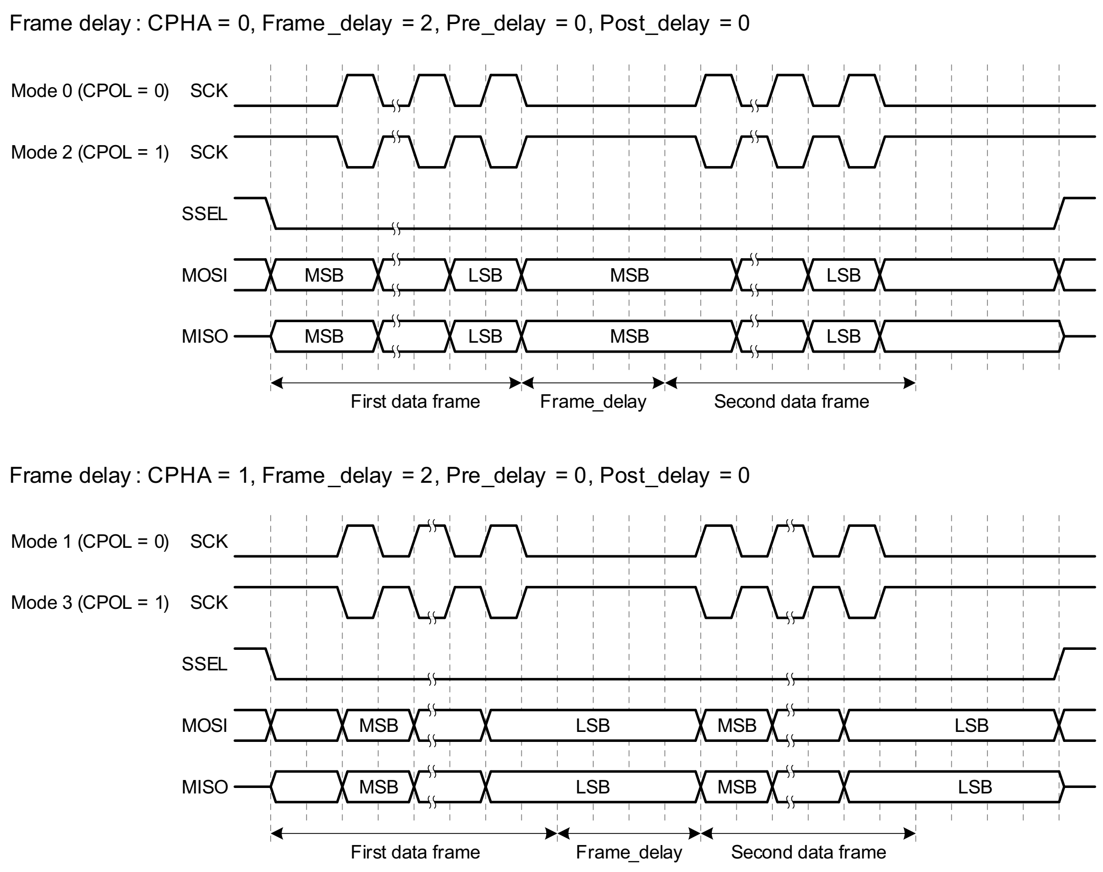


Fig 29. Frame_delay

19.7.2.3 Transfer_delay

The Transfer_delay value controls the minimum amount of time that SSEL is de-asserted between transfers, because the EOT bit = 1. When Transfer_delay = 0, SSEL may be de-asserted for a minimum of one SPI clock time. Transfer_delay is illustrated by the examples in Figure 30.

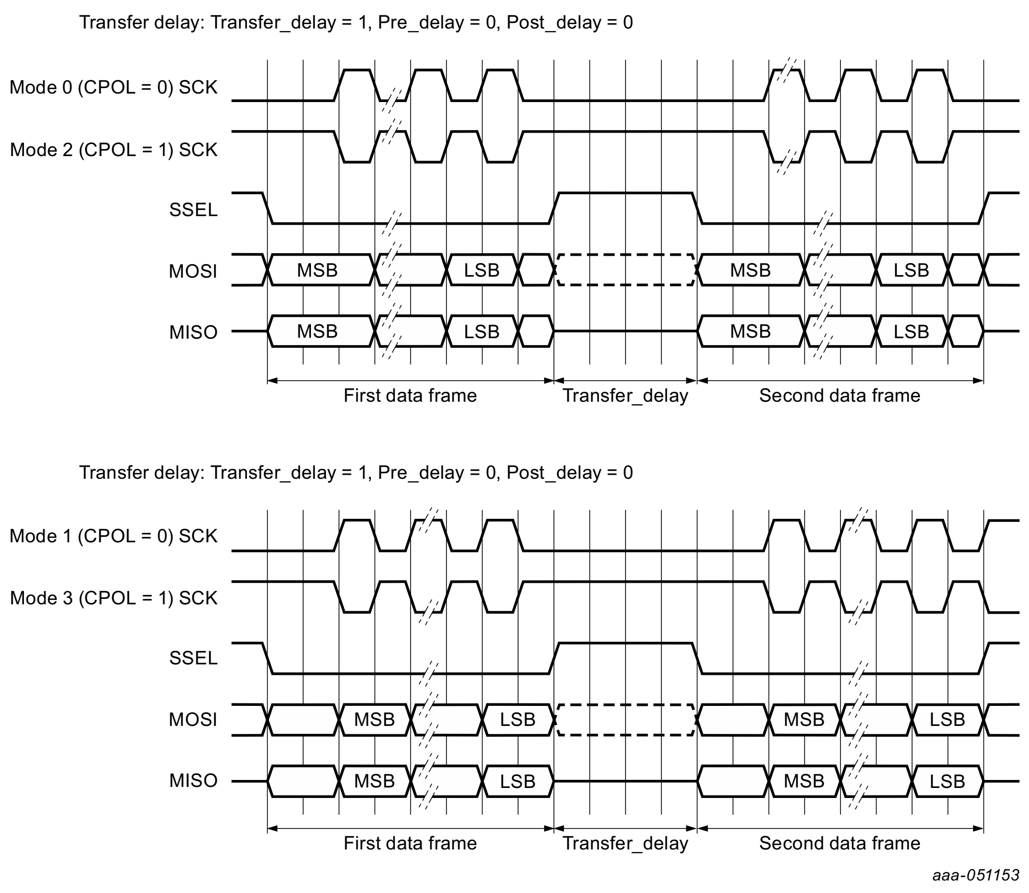


Fig 30. Transfer_delay

19.7.3 Clocking and data rates

In order to use the SPI, clocking details must be defined. This includes configuring the system clock and selection of the clock divider value in DIV. See Figure 25.

19.7.3.1 Data rate calculations

The SPI interface is designed to operate asynchronously from any on-chip clocks, and without the need for overclocking.

In slave mode, this means that the SCK from the external master is used directly to run the transmit and receive shift registers and other logic.

In master mode, the SPI rate clock produced by the SPI clock divider is used directly as the outgoing SCK.

The SPI clock divider is an integer divider. The SPI in master mode can be set to run at the same speed as the selected PCLK, or at lower integer divide rates. The SPI rate will be = PCLK_SPIn / DIVVAL.

In slave mode, the clock is taken from the SCK input and the SPI clock divider is not used.

19.7.4 Slave select

The SPI block provides for four Slave Select inputs in slave mode or outputs in master mode. Each SSEL can be set for normal polarity (active low), or can be inverted (active high). Representation of the 4 SSELs in a register is always active low. If an SSEL is inverted, this is done as the signal leaves/enters the SPI block.

In slave mode, **any** asserted SSEL that is connected to a pin will activate the SPI. In master mode, all SSELs that are connected to a pin will be output as defined in the SPI registers. In the latter case, the SSELs could potentially be decoded externally in order to address more than four slave devices. Note that at least one SSEL is asserted when data is transferred in master mode.

In master mode, Slave Selects come from the SSELN field, which appears in both the CTL and DATCTL registers. In slave mode, the state of all four SSELs is saved along with received data in the RXSSEL_N field of the RXDAT register.

19.7.5 DMA operation

A DMA request is provided for each SPI direction, and can be used in lieu of interrupts for transferring data by configuring the DMA controller appropriately, and enabling the Rx and/or Tx DMA via the CFG register. The DMA controller provides an acknowledgement signal that clears the related request when it completes handling that request.

The transmitter DMA request is asserted when Tx DMA is enabled and the transmitter can accept more data.

The receiver DMA request is asserted when Rx DMA is enabled and received data is available to be read.

19.7.6 Data lengths greater than 16 bits

The SPI interface handles data frame sizes from 1 to 16 bits directly. Larger sizes can be handled by splitting data up into groups of 16 bits or less. For example, 24 bits can be supported as 2 groups of 16 bits and 8 bits or 2 groups of 12 bits, among others. Frames of any size, including greater than 32 bits, can supported in the same way.

Details of how to handle larger data widths depend somewhat on other SPI configuration options. For instance, if it is intended for Slave Selects to be de-asserted between frames, then this must be suppressed when a larger frame is split into more than one part. Sending 2 groups of 12 bits with SSEL de-asserted between 24-bit increments, for instance, would require changing the value of the EOF bit on alternate 12-bit frames.

19.7.7 Data stalls

A stall for Master transmit data can happen in modes 0 and 2 when SCK cannot be returned to the rest state until the MSB of the next data frame can be driven on MOSI. In this case, the stall happens just before the final clock edge of data if the next piece of data is not yet available.

A stall for Master receive can happen when a receiver overrun would otherwise occur if the transmitter was not stalled. In modes 0 and 2, this occurs if the previously received data is not read before the end of the next piece of is received. This stall happens one clock edge earlier than the transmitter stall.

In modes 1 and 3, the same kind of receiver stall can occur, but just before the final clock edge of the received data. Also, a transmitter stall will not happen in modes 1 and 3 because the transmitted data is complete at the point where a stall would otherwise occur, so it is not needed.

Stalls are reflected in the STAT register by the Stalled status flag, which indicates the current SPI status.

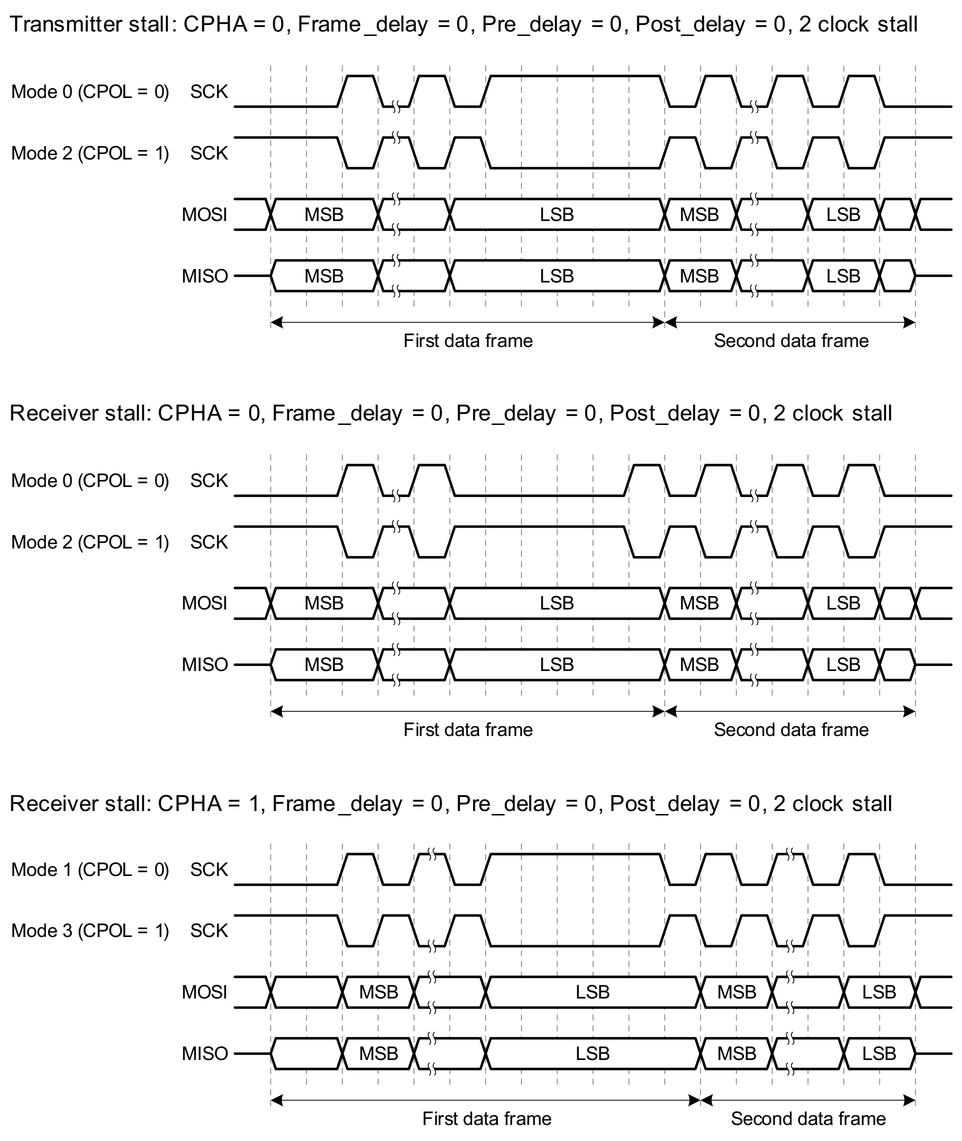

Transmitter stall: CPHA = 0, Frame_delay = 0, Pre_delay = 0, Post_delay = 0, 2 clock stall
Mode 0 (CPOL = 0) SCK
Mode 2 (CPOL = 1) SCK
MOSI
MISO
MSB
LSB
First data frame
Second data frame
Receiver stall: CPHA = 0, Frame_delay = 0, Pre_delay = 0, Post_delay = 0, 2 clock stall
Receiver stall: CPHA = 1, Frame_delay = 0, Pre_delay = 0, Post_delay = 0, 2 clock stall
Mode 1 (CPOL = 0) SCK
Mode 3 (CPOL = 1) SCK

Fig 31. Examples of data stalls

UM11607

Chapter 20: I2C0

Rev. 3 — April 2023 User manual

20.1 How to read this chapter

One I2C interface is available on all parts.

Read this chapter if you want to understand the I2C operation and the software interface and want to learn how to use the I2C for wake-up from reduced power modes.

20.2 Features

- Independent Master, Slave, and Monitor functions.
- Supports both DMA_ITRIG_PINMUX Multi-master and Multi-master with Slave functions.
- Multiple I^2C slave addresses supported in hardware.
- One slave address can be selectively qualified with a bit mask or an address range in order to respond to multiple I^2C bus addresses.
- 10-bit addressing supported with software assist.
- Supports System Management (SMBus).
- Separate DMA requests for Master and Slave.
- Supports the I^2C-bus specification up to Fast-mode Plus (up to 1 MHz).

20.3 Basic configuration

Configure the I2C interfaces using the following registers:

- In the SYSAHBCLKCTRL register, set the corresponding bits to enable the clocks to the register interfaces. See Table 94.
- Clear the I2C peripheral resets using the PRESETCTRL register (Table 95).
- Enable/disable the I2C interrupt in interrupt slots #8 in the NVIC. See Table 46.
- Configure the I2C pin functions through the switch matrix. See Table 291.
- The peripheral clock for the I2C is the system clock (see Figure 32).

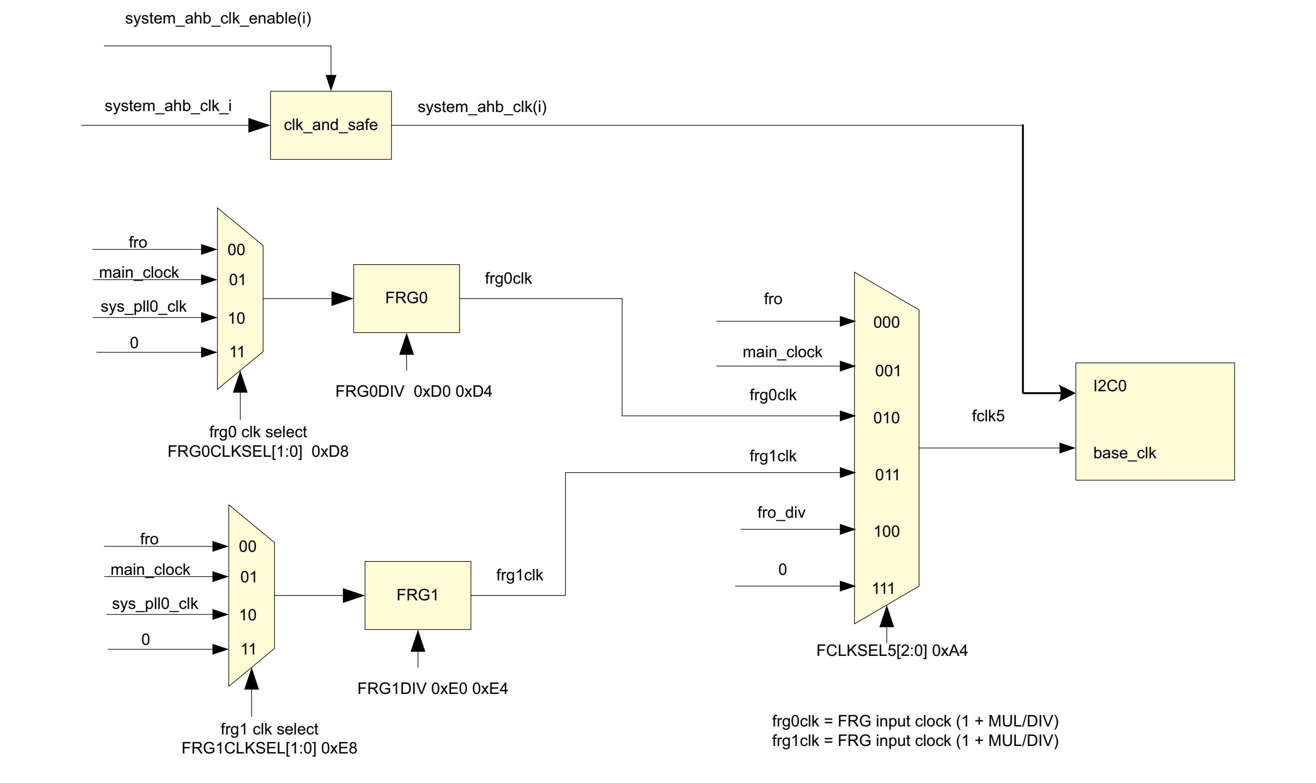


Fig 32. I2C clocking

20.3.1 I2C transmit/receive in master mode

In this example, the I2C is configured as the master. The master sends 8 bits to the slave and then receives 8 bits from the slave. The system clock is set to 60 MHz and the bit rate is approximately 400 kHz. You must enable the I2C0_SCL and I2C0_SDA functions on pins PIO0_11 and PIO0_10 or assign the SCL and SDA functions for any of the other I2C blocks to pins through the switch matrix. See Table 291.

For a 400 kHz bit rate, the I2C0 pins can be configured in standard mode in the IOCON block. See Table 150 “PIO0_11 register (PIO0_11, address 0x4004 401C) bit description” and Table 151 “PIO0_10 register (PIO0_10, address 0x4004 4020) bit description”.

The transmission of the address and data bits is controlled by the state of the MSTPENDING status bit. Whenever the status is Master pending, the master can read or write to the MSTDAT register and go to the next step of the transmission protocol by writing to the MSTCTL register.

Configure the I2C bit rate:

- Divide the system clock (I2C_PCLK) by a factor of 2. See Table 300 “I^{2}C Clock Divider register (CLKDIV, address 0x4005 0014 (I2C0)) bit description”.
- Set the SCL high and low times to 2 clock cycles each. This is the default. See Table 303 “Master Time register (MSTTIME, address 0x4005 0024 (I2C0)) bit description”. The result is an SCL clock of 375 kHz.

20.3.1.1 Master write to slave

Configure the I2C as master: Set the MSTEN bit to 1 in the CFG register. See Table 293.

Write data to the slave:

1. Write the slave address with the $\overline{\text{RW}}$ bit set to 0 to the Master data register MSTDAT. See Table 304.
2. Start the transmission by setting the MSTSTART bit to 1 in the Master control register. See Table 302. The following happens:
   – The pending status is cleared and the I2C-bus is busy.
   – The I2C master sends the start bit and address with the $\overline{\text{RW}}$ bit to the slave.
3. Wait for the pending status to be set (MSTPENDING = 1) by polling the STAT register.
4. Write 8 bits of data to the MSTDAT register.
5. Continue with the transmission of data by setting the MSTCONT bit to 1 in the Master control register. See Table 302. The following happens:
   – The pending status is cleared and the I2C-bus is busy.
   – The I2C master sends the data bits to the slave address.
6. Wait for the pending status to be set (MSTPENDING = 1) by polling the STAT register.
7. Stop the transmission by setting the MSTSTOP bit to 1 in the Master control register. See Table 302.

20.3.1.2 Master read from slave

Configure the I2C as master: Set the MSTEN bit to 1 in the CFG register. See Table 293.

Read data from the slave:

1. Write the slave address with the $\overline{\text{RW}}$ bit set to 1 to the Master data register MSTDAT. See Table 304.
2. Start the transmission by setting the MSTSTART bit to 1 in the Master control register. See Table 302. The following happens:
   – The pending status is cleared and the I2C-bus is busy.
   – The I2C master sends the start bit and address with the $\overline{\text{RW}}$ bit to the slave.
   – The slave sends 8 bit of data.
3. Wait for the pending status to be set (MSTPENDING = 1) by polling the STAT register.
4. Read 8 bits of data from the MSTDAT register.
5. Stop the transmission by setting the MSTSTOP bit to 1 in the Master control register. See Table 302.

20.3.2 I2C receive/transmit in slave mode

In this example, the I2C is configured as the slave. The slave receives 8 bits from the master and sends 8 bits to the slave. You must enable the I2C0_SCL and I2C0_SDA functions on pins PIO0_11 and PIO0_10 or assign the SCL and SDA functions for any of the other I2C blocks to pins through the switch matrix. See Table 291.

For a 400 kHz bit rate, the pins can be configured in standard mode in the IOCON block. See Table 150 "PIO0_11 register (PIO0_11, address 0x4004 401C) bit description" and Table 151 "PIO0_10 register (PIO0_10, address 0x4004 4020) bit description".

The transmission of the address and data bits is controlled by the state of the SLVPENDING status bit. Whenever the status is Slave pending, the slave can acknowledge ("ack") or send or receive an address and data. The received data or the data to be sent to the master are available in the SLVDAT register. After sending and receiving data, continue to the next step of the transmission protocol by writing to the SLVCTL register.

20.3.2.1 Slave read from master

Configure the I2C as slave with address x:

- Set the SLVEN bit to 1 in the CFG register. See Table 293.
- Write the slave address x to the address 0 match register. See Table 307.

Read data from the master:

1. Wait for the pending status to be set (SLVPENDING = 1) by polling the STAT register.
2. Acknowledge ("ack") the address by setting SLVCONTINUE = 1 in the slave control register. See Table 305.
3. Wait for the pending status to be set (SLVPENDING = 1) by polling the STAT register.
4. Read 8 bits of data from the SLVDAT register. See Table 306.
5. Acknowledge ("ack") the data by setting SLVCONTINUE = 1 in the slave control register. See Table 305.

20.3.2.2 Slave write to master

- Set the SLVEN bit to 1 in the CFG register. See Table 293.
- Write the slave address x to the address 0 match register. See Table 307.

Write data to the master:

1. Wait for the pending status to be set (SLVPENDING = 1) by polling the STAT register.
2. ACK the address by setting SLVCONTINUE = 1 in the slave control register. See Table 305.
3. Wait for the pending status to be set (SLVPENDING = 1) by polling the STAT register.
4. Write 8 bits of data to SLVDAT register. See Table 306.
5. Continue the transaction by setting SLVCONTINUE = 1 in the slave control register. See Table 305.

20.3.3 Configure the I2C for wake-up

In sleep mode, any activity on the I2C-bus that triggers an I2C interrupt can wake up the part, provided that the interrupt is enabled in the INTENSET register and the NVIC. As long as the I2C clock I2C_PCLK remains active in sleep mode, the I2C can wake up the part independently of whether the I2C block is configured in master or slave mode.

In Deep-sleep or Power-down mode, the I2C clock is turned off as are all peripheral clocks. However, if the I2C is configured in slave mode and an external master on the I2C-bus provides the clock signal, the I2C block can create an interrupt asynchronously. This interrupt, if enabled in the NVIC and in the I2C block's INTENCLR register, can then wake up the core.

20.3.3.1 Wake-up from Sleep mode

- Enable the I2C interrupt in the NVIC.
- Enable the I2C wake-up event in the I2C INTENSET register. Wake-up on any enabled interrupts is supported (see the INTENSET register). Examples are the following events:
  - Master pending
  - Change to idle state
  - Start/stop error
  - Slave pending
  - Address match (in slave mode)
  - Data available/ready

20.3.3.2 Wake-up from Deep-sleep and Power-down modes

- Enable the I2C interrupt in the NVIC.
- Enable the I2C interrupt in the STARTERP1 register in the SYSCON block to create the interrupt signal asynchronously while the core and the peripheral are not clocked. See Table 115 "Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description".

- In the PDAWAKE register, configure all peripherals that need to be running when the part wakes up.
- Configure the I2C in slave mode.
- Enable the I2C the interrupt in the I2C INTENCLR register which configures the interrupt as wake-up event. Examples are the following events:
  - Slave deselect
  - Slave pending (wait for read, write, or ACK)
  - Address match
  - Data available/ready for the monitor

20.4 Pin description

The I2C0 pins are fixed-pin functions and enabled through the switch matrix.

If the I2C0-bus interface is used in Fast-mode Plus mode, configure the I2C-pins for this mode in the IOCON block: Table 150 "PIO0_11 register (PIO0_11, address 0x4004 401C) bit description" and Table 151 "PIO0_10 register (PIO0_10, address 0x4004 4020) bit description".

Table 291. I2C-bus pin description

Function	Direction	Type	Connect to	Use register	Reference	Description
I2C0_SDA	I/O	external to pin	Any pin	PINASSIGN6	Table 134	I2C0 serial data.
I2C0_SCL	I/O	external to pin	Any pin	PINASSIGN7	Table 135	I2C0 serial clock.

20.5 General description

The architecture of the I2C-bus interface is shown in Figure 33.

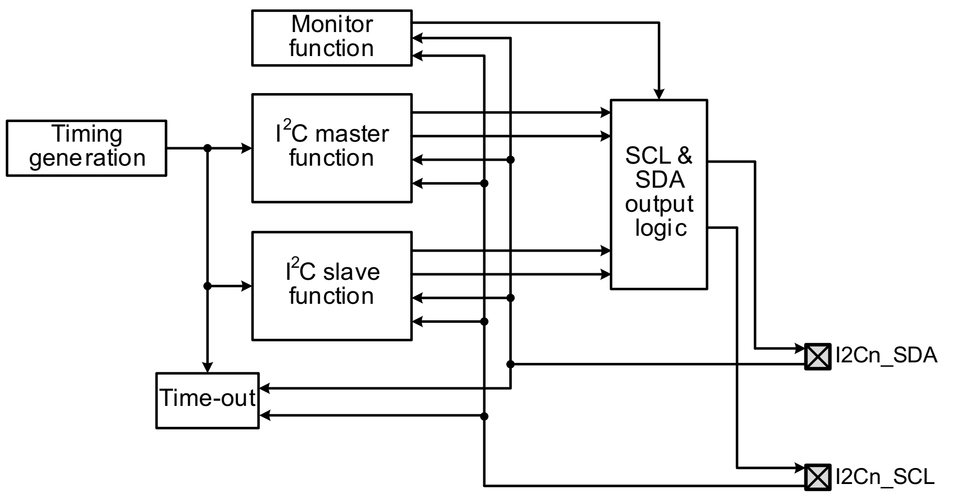


Fig 33. I2C block diagram

20.6 Register description

Table 292. Register overview: I2C (base address 0x4005 0000 (I2C0))

Name	Access	Offset	Description	Reset value	Reference
CFG	R/W	0x00	Configuration for shared functions.	0	Table 293
STAT	R/W	0x04	Status register for Master, Slave, and Monitor functions.	0x000801	Table 294
INTENSET	R/W	0x08	Interrupt Enable Set and read register.	0	Table 297
INTENCLR	W	0x0C	Interrupt Enable Clear register.	NA	Table 298
TIMEOUT	R/W	0x10	Time-out value register.	0xFFFF	Table 299
CLKDIV	R/W	0x14	Clock pre-divider for the entire I^{2}C block. This determines what time increments are used for the MSTTIME and SLVTIME registers.	0	Table 300
INTSTAT	R	0x18	Interrupt Status register for Master, Slave, and Monitor functions.	0	Table 301
MSTCTL	R/W	0x20	Master control register.	0	Table 302
MSTTIME	R/W	0x24	Master timing configuration.	0x77	Table 303
MSTDAT	R/W	0x28	Combined Master receiver and transmitter data register.	NA	Table 304
SLVCTL	R/W	0x40	Slave control register.	0	Table 305
SLVDAT	R/W	0x44	Combined Slave receiver and transmitter data register.	NA	Table 306
SLVADR0	R/W	0x48	Slave address 0.	0x01	Table 307
SLVADR1	R/W	0x4C	Slave address 1.	0x01	Table 307
SLVADR2	R/W	0x50	Slave address 2.	0x01	Table 307
SLVADR3	R/W	0x54	Slave address 3.	0x01	Table 307
SLVQUAL0	R/W	0x58	Slave Qualification for address 0.	0	Table 308
MONRXDAT	RO	0x80	Monitor receiver data register.	0	Table 309

20.6.1 I2C Configuration register

The CFG register contains mode settings that apply to Master, Slave, and Monitor functions.

Table 293. I2C Configuration register (CFG, address 0x4005 0000 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset Value
0	MSTEN		Master Enable. When disabled, configurations settings for the Master function are not changed, but the Master function is internally reset.	0
		0	Disabled. The I^{2}C Master function is disabled.	
		1	Enabled. The I^{2}C Master function is enabled.	
1	SLVEN		Slave Enable. When disabled, configurations settings for the Slave function are not changed, but the Slave function is internally reset.	0
		0	Disabled. The I^{2}C slave function is disabled.	
		1	Enabled. The I^{2}C slave function is enabled.	

Table 293. I2C Configuration register (CFG, address 0x4005 0000 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset Value
2	MONEN		Monitor Enable. When disabled, configurations settings for the Monitor function are not changed, but the Monitor function is internally reset.	0
		0	Disabled. The I^{2}C monitor function is disabled.	
		1	Enabled. The I^{2}C monitor function is enabled.	
3	TIMEOUTEN		I^{2}C bus Time-out Enable. When disabled, the time-out function is internally reset.	0
		0	Disabled. Time-out function is disabled.	
		1	Enabled. Time-out function is enabled. Both types of time-out flags will be generated and will cause interrupts if they are enabled. Typically, only one time-out will be used in a system.	
4	MONCLKSTR		Monitor function Clock Stretching.	0
		0	Disabled. The monitor function will not perform clock stretching. Software or DMA may not always be able to read data provided by the monitor function before it is overwritten. This mode may be used when non-invasive monitoring is critical.	
		1	Enabled. The monitor function will perform clock stretching in order to ensure that software or DMA can read all incoming data supplied by the monitor function.	
31:5	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.2 I2C Status register

The STAT register provides status flags and state information about all of the functions of the I^{2}C block. Some information in this register is read-only and some flags can be cleared by writing a 1 to them.

Access to bits in this register varies. RO = Read-only, W1 = write 1 to clear.

Details on the master and slave states described in the MSTSTATE and SLVSTATE bits in this register are listed in Table 295 and Table 296.

Table 294. I^{2}C Status register (STAT, address 0x4005 0004 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value	Access
0	MSTPENDING		Master Pending. Indicates that the Master is waiting to continue communication on the I2C-bus (pending) or is idle. When the master is pending, the MSTSTATE bits indicate what type of software service if any the master expects. This flag will cause an interrupt when set if, enabled via the INTENSET register. The MSTPENDING flag is not set when the DMA is handling an event (if the MSTDMA bit in the MSTCTL register is set). If the master is in the idle state, and no communication is needed, mask this interrupt.	1	RO
		0	In progress. Communication is in progress and the Master function is busy and cannot currently accept a command.		
		1	Pending. The Master function needs software service or is in the idle state. If the master is not in the idle state, it is waiting to receive or transmit data or the NACK bit.		
3:1	MSTSTATE		Master State code. The master state code reflects the master state when the MSTPENDING bit is set, that is the master is pending or in the idle state. Each value of this field indicates a specific required service for the Master function. All other values are reserved.	0	RO
		0x0	Idle. The Master function is available to be used for a new transaction.		
		0x1	Receive ready. Received data available (Master Receiver mode). Address plus Read was previously sent and Acknowledged by slave.		
		0x2	Transmit ready. Data can be transmitted (Master Transmitter mode). Address plus Write was previously sent and Acknowledged by slave.		
		0x3	NACK Address. Slave NACKed address.		
		0x4	NACK Data. Slave NACKed transmitted data.		
4	MSTARBLOSS		Master Arbitration Loss flag. This flag can be cleared by software writing a 1 to this bit. It is also cleared automatically a 1 is written to MSTCONTINUE.	0	W1
		0	No loss. No Arbitration Loss has occurred.		
		1	Arbitration loss. The Master function has experienced an Arbitration Loss. At this point, the Master function has already stopped driving the bus and gone to an idle state. Software can respond by doing nothing, or by sending a Start in order to attempt to gain control of the bus when it next becomes idle.		
5	-		Reserved. Read value is undefined, only zero should be written.	NA	NA

Table 294. I^{2}C Status register (STAT, address 0x4005 0004 (I2C0)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
6	MSTSTSTPERR		Master Start/Stop Error flag. This flag can be cleared by software writing a 1 to this bit. It is also cleared automatically a 1 is written to MSTCONTINUE.	0	W1
		0	No Start/Stop Error has occurred.		
		1	Start/stop error has occurred. The Master function has experienced a Start/Stop Error. A Start or Stop was detected at a time when it is not allowed by the I^{2}C specification. The Master interface has stopped driving the bus and gone to an idle state, no action is required. A request for a Start could be made, or software could attempt to insure that the bus has not stalled.		
7	-		Reserved. Read value is undefined, only zero should be written.	NA	NA
8	SLVPENDING		Slave Pending. Indicates that the Slave function is waiting to continue communication on the I2C-bus and needs software service. This flag will cause an interrupt when set if enabled via INTENSET. The SLVPENDING flag is not set when the DMA is handling an event (if the SLVDMA bit in the SLVCTL register is set). The SLVPENDING flag is read-only and is automatically cleared when a 1 is written to the SLVCONTINUE bit in the MSTCTL register.	0	RO
		0	In progress. The Slave function does not currently need service.		
		1	Pending. The Slave function needs service. Information on what is needed can be found in the adjacent SLVSTATE field.		
10:9	SLVSTATE		Slave State code. Each value of this field indicates a specific required service for the Slave function. All other values are reserved.	0	RO
		0x0	Slave address. Address plus R/W received. At least one of the four slave addresses has been matched by hardware.		
		0x1	Slave receive. Received data is available (Slave Receiver mode).		
		0x2	Slave transmit. Data can be transmitted (Slave Transmitter mode).		
		0x3	Reserved.		
11	SLVNOTSTR		Slave Not Stretching. Indicates when the slave function is stretching the I^{2}C clock. This is needed in order to gracefully invoke Deep Sleep or Power-down modes during slave operation. This read-only flag reflects the slave function status in real time.	1	RO
		0	Stretching. The slave function is currently stretching the I^{2}C bus clock. Deep-Sleep or Power-down mode cannot be entered at this time.		
		1	Not stretching. The slave function is not currently stretching the I^{2}C bus clock. Deep-sleep or Power-down mode could be entered at this time.		

Table 294. I^{2}C Status register (STAT, address 0x4005 0004 (I2C0)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
13:12	SLVIDX		Slave address match Index. This field is valid when the I^{2}C slave function has been selected by receiving an address that matches one of the slave addresses defined by any enabled slave address registers, and provides an identification of the address that was matched. It is possible that more than one address could be matched, but only one match can be reported here.	0	RO
		0x0	Slave address 0 was matched.		
		0x1	Slave address 1 was matched.		
		0x2	Slave address 2 was matched.		
		0x3	Slave address 3 was matched.		
14	SLVSEL		Slave selected flag. SLVSEL is set after an address match when software tells the Slave function to acknowledge the address. It is cleared when another address cycle presents an address that does not match an enabled address on the Slave function, when slave software decides to NACK a matched address, or when there is a Stop detected on the bus. SLVSEL is not cleared if software Nacks data.	0	RO
		0	Not selected. The Slave function is not currently selected.		
		1	Selected. The Slave function is currently selected.		
15	SLVDESEL		Slave Deselected flag. This flag will cause an interrupt when set if enabled via INTENSET. This flag can be cleared by writing a 1 to this bit.	0	W1
		0	Not deselected. The Slave function has not become deselected. This does not mean that it is currently selected. That information can be found in the SLVSEL flag.		
		1	Deselected. The Slave function has become deselected. This is specifically caused by the SLVSEL flag changing from 1 to 0. See the description of SLVSEL for details on when that event occurs.		
16	MONRDY		Monitor Ready. This flag is cleared when the MONRXDAT register is read.	0	RO
		0	No data. The Monitor function does not currently have data available.		
		1	Data waiting. The Monitor function has data waiting to be read.		
17	MONOV		Monitor Overflow flag.	0	W1
		0	No overrun. Monitor data has not overrun.		
		1	Overrun. A Monitor data overrun has occurred. This can only happen when Monitor clock stretching not enabled via the MONCLKSTR bit in the CFG register. Writing 1 to this bit clears the flag.		
18	MONACTIVE		Monitor Active flag. This flag indicates when the Monitor function considers the I^{2}C bus to be active. Active is defined here as when some Master is on the bus: a bus Start has occurred more recently than a bus Stop.	0	RO
		0	Inactive. The Monitor function considers the I^{2}C bus to be inactive.		
		1	Active. The Monitor function considers the I^{2}C bus to be active.		

Table 294. I^{2}C Status register (STAT, address 0x4005 0004 (I2C0)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
19	MONIDLE		Monitor Idle flag. This flag is set when the Monitor function sees the I^{2}C bus change from active to inactive. This can be used by software to decide when to process data accumulated by the Monitor function. This flag will cause an interrupt when set if enabled via the INTENSET register . The flag can be cleared by writing a 1 to this bit.	0	W1
		0	Not idle. The I^{2}C bus is not idle, or this flag has been cleared by software.		
		1	Idle. The I^{2}C bus has gone idle at least once since the last time this flag was cleared by software.		
23:20	-		Reserved. Read value is undefined, only zero should be written.	NA	NA
24	EVENTTIMEOUT		Event Time-out Interrupt flag. Indicates when the time between events has been longer than the time specified by the TIMEOUT register. Events include Start, Stop, and clock edges. The flag is cleared by writing a 1 to this bit. No time-out is created when the I2C-bus is idle.	0	W1
		0	No time-out. I^{2}C bus events have not caused a time-out.		
		1	Event time-out. The time between I^{2}C bus events has been longer than the time specified by the I2C TIMEOUT register.		
25	SCLTIMEOUT		SCL Time-out Interrupt flag. Indicates when SCL has remained low longer than the time specific by the TIMEOUT register. The flag is cleared by writing a 1 to this bit.	0	W1
		0	No time-out. SCL low time has not caused a time-out.		
		1	Time-out. SCL low time has caused a time-out.		
31:26	-		Reserved. Read value is undefined, only zero should be written.	NA	NA

Table 295. Master function state codes (MSTSTATE)

MSTSTATE	Description	Actions	DMA access allowed
0x0	**Idle.** The Master function is available to be used for a new transaction.	Send a Start or disable MSTPENDING interrupt if the Master function is not needed currently.	No
0x1	**Received data is available (Master Receiver mode).** Address plus Read was previously sent and Acknowledged by slave.	Read data and either continue, send a Stop, or send a Repeated Start.	Yes
0x2	**Data can be transmitted (Master Transmitter mode).** Address plus Write was previously sent and Acknowledged by slave.	Send data and continue, or send a Stop or Repeated Start.	Yes
0x3	**Slave NACKed address.**	Send a Stop or Repeated Start.	No
0x4	**Slave NACKed transmitted data.**	Send a Stop or Repeated Start.	No

Table 296. Slave function state codes (SLVSTATE)

SLVSTATE		Description	Actions	DMA access allowed
0	SLVST_ADDR	**Address plus R/W received.** At least one of the 4 slave addresses has been matched by hardware.	Software can further check the address if needed, for instance if a subset of addresses qualified by SLVQUAL0 is to be used. Software can ACK or NACK the address by writing 1 to either SLVCONTINUE or SLVNACK. Also see Section 20.7.3 regarding 10-bit addressing.	No
1	SLVST_RX	**Received data is available (Slave Receiver mode).**	Read data reply with an ACK or a NACK.	Yes
2	SLVST_TX	**Data can be transmitted (Slave Transmitter mode).**	Send data.	Yes
3	-	Reserved.	-	-

20.6.3 Interrupt Enable Set and read register

The INTENSET register controls which I^{2}C status flags generate interrupts. Writing a 1 to a bit position in this register enables an interrupt in the corresponding position in the STAT register, if an interrupt is supported there. Reading INTENSET indicates which interrupts are currently enabled.

Table 297. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
0	MSTPENDINGEN		Master Pending interrupt Enable.	0
		0	The MstPending interrupt is disabled.	
		1	The MstPending interrupt is enabled.	
3:1	-		Reserved. Read value is undefined, only zero should be written.	NA
4	MSTARBLOSSEN		Master Arbitration Loss interrupt Enable.	0
		0	The MstArbLoss interrupt is disabled.	
		1	The MstArbLoss interrupt is enabled.	
5	-		Reserved. Read value is undefined, only zero should be written.	NA
6	MSTSTSTPERREN		Master Start/Stop Error interrupt Enable.	0
		0	The MstStStpErr interrupt is disabled.	
		1	The MstStStpErr interrupt is enabled.	
7	-		Reserved. Read value is undefined, only zero should be written.	NA
8	SLVPENDINGEN		Slave Pending interrupt Enable.	0
		0	The SlvPending interrupt is disabled.	
		1	The SlvPending interrupt is enabled.	
10:9	-		Reserved. Read value is undefined, only zero should be written.	NA

Table 297. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
11	SLVNOTSTREN		Slave Not Stretching interrupt Enable.	0
		0	The SlvNotStr interrupt is disabled.	
		1	The SlvNotStr interrupt is enabled.	
14:12	-		Reserved. Read value is undefined, only zero should be written.	NA
15	SLVDESELEN		Slave Deselect interrupt Enable.	0
		0	The SlvDeSel interrupt is disabled.	
		1	The SlvDeSel interrupt is enabled.	
16	MONRDYEN		Monitor data Ready interrupt Enable.	0
		0	The MonRdy interrupt is disabled.	
		1	The MonRdy interrupt is enabled.	
17	MONOVEN		Monitor Overrun interrupt Enable.	0
		0	The MonOv interrupt is disabled.	
		1	The MonOv interrupt is enabled.	
18	-		Reserved. Read value is undefined, only zero should be written.	NA
19	MONIDLEEN		Monitor Idle interrupt Enable.	0
		0	The MonIdle interrupt is disabled.	
		1	The MonIdle interrupt is enabled.	
23:20	-		Reserved. Read value is undefined, only zero should be written.	NA
24	EVENTTIMEOUTEN		Event time-out interrupt Enable.	0
		0	The Event time-out interrupt is disabled.	
		1	The Event time-out interrupt is enabled.	
25	SCLTIMEOUTEN		SCL time-out interrupt Enable.	0
		0	The SCL time-out interrupt is disabled.	
		1	The SCL time-out interrupt is enabled.	
31:26	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.4 Interrupt Enable Clear register

Writing a 1 to a bit position in INTENCLR clears the corresponding position in the INTENSET register, disabling that interrupt. INTENCLR is a write-only register.

Bits that do not correspond to defined bits in INTENSET are reserved and only zeroes should be written to them.

Table 298. Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0)) bit description

Bit	Symbol	Description	Reset value
0	MSTPENDINGCLR	Master Pending interrupt clear. Writing 1 to this bit clears the corresponding bit in the INTENSET register if implemented.	0
3:1	-	Reserved. Read value is undefined, only zero should be written.	NA
4	MSTARBLOSSCLR	Master Arbitration Loss interrupt clear.	0
5	-	Reserved. Read value is undefined, only zero should be written.	NA
6	MSTSTSTPERRCLR	Master Start/Stop Error interrupt clear.	0
7	-	Reserved. Read value is undefined, only zero should be written.	NA
8	SLVPENDINGCLR	Slave Pending interrupt clear.	0
10:9	-	Reserved. Read value is undefined, only zero should be written.	NA
11	SLVNOTSTRCLR	Slave Not Stretching interrupt clear.	0
14:12	-	Reserved. Read value is undefined, only zero should be written.	NA
15	SLVDESELCLR	Slave Deselect interrupt clear.	0
16	MONRDYCLR	Monitor data Ready interrupt clear.	0
17	MONOVCLR	Monitor Overrun interrupt clear.	0
18	-	Reserved. Read value is undefined, only zero should be written.	NA
19	MONIDLECLR	Monitor Idle interrupt clear.	0
23:20	-	Reserved. Read value is undefined, only zero should be written.	NA
24	EVENTTIMEOUTCLR	Event time-out interrupt clear.	0
25	SCLTIMEOUTCLR	SCL time-out interrupt clear.	0
31:26	-	Reserved. Read value is undefined, only zero should be written.	NA

20.6.5 Time-out value register

The TIMEOUT register allows setting an upper limit to certain I^{2}C bus times, informing by status flag and/or interrupt when those times are exceeded.

Two time-outs are generated, and software can elect to use either of them.

1. EVENTTIMEOUT checks the time between bus events while the bus is not idle: Start, SCL rising, SCL falling, and Stop. The EVENTTIMEOUT status flag in the STAT register is set if the time between any two events becomes longer than the time configured in the TIMEOUT register. The EVENTTIMEOUT status flag can cause an interrupt if enabled to do so by the EVENTTIMEOUTEN bit in the INTENSET register.

2. SCLTIMEOUT checks only the time that the SCL signal remains low while the bus is not idle. The SCLTIMEOUT status flag in the STAT register is set if SCL remains low longer than the time configured in the TIMEOUT register. The SCLTIMEOUT status flag can cause an interrupt if enabled to do so by the SCLTIMEOUTEN bit in the INTENSET register. The SCLTIMEOUT can be used with the SMBus.

Also see Section 20.7.2 “Time-out”.

Table 299. Time-out value register (TIMEOUT, address 0x4005 0010 (I2C0)) bit description

Bit	Symbol	Description	Reset value
3:0	TOMIN	Time-out time value, bottom four bits. These are hard-wired to 0xF. This gives a minimum time-out of 16 I^2C function clocks and also a time-out resolution of 16 I^2C function clocks.	0xF
15:4	TO	Time-out time value. Specifies the time-out interval value in increments of 16 I^2C function clocks, as defined by the CLKDIV register. To change this value while I^2C is in operation, disable all time-outs, write a new value to TIMEOUT, then re-enable time-outs. 0x000 = A time-out will occur after 16 counts of the I^2C function clock. 0x001 = A time-out will occur after 32 counts of the I^2C function clock. ... 0xFFF = A time-out will occur after 65,536 counts of the I^2C function clock.	0xFFF
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

20.6.6 Clock Divider register

The CLKDIV register divides down the Peripheral Clock (PCLK) to produce the I^2C function clock that is used to time various aspects of the I^2C interface. The I^2C function clock is used for some internal operations in the I^2C block and to generate the timing required by the I^2C bus specification, some of which are user configured in the MSTTIME register for Master operation and the SLVTIME register for Slave operation.

See Section 20.7.1.1 “Rate calculations” for details on bus rate setup.

Table 300. I^2C Clock Divider register (CLKDIV, address 0x4005 0014 (I2C0)) bit description

Bit	Symbol	Description	Reset value
15:0	DIVVAL	This field controls how the clock (PCLK) is used by the I^2C functions that need an internal clock in order to operate. 0x0000 = PCLK is used directly by the I^2C function. 0x0001 = PCLK is divided by 2 before use by the I^2C function. 0x0002 = PCLK is divided by 3 before use by the I^2C function. ... 0xFFFF = PCLK is divided by 65,536 before use by the I^2C function.	0
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

20.6.7 Interrupt Status register

The INTSTAT register provides register provides a view of those interrupt flags that are currently enabled. This can simplify software handling of interrupts. See Table 294 for detailed descriptions of the interrupt flags.

Table 301. I^{2}C Interrupt Status register (INTSTAT, address 0x4005 0018 (I2C0)) bit description

Bit	Symbol	Description	Reset value
0	MSTPENDING	Master Pending.	1
3:1	-	Reserved.	
4	MSTARBLOSS	Master Arbitration Loss flag.	0
5	-	Reserved. Read value is undefined, only zero should be written.	NA
6	MSTSTSTPERR	Master Start/Stop Error flag.	0
7	-	Reserved. Read value is undefined, only zero should be written.	NA
8	SLVPENDING	Slave Pending.	0
10:9	-	Reserved. Read value is undefined, only zero should be written.	NA
11	SLVNOTSTR	Slave Not Stretching status.	1
14:12	-	Reserved. Read value is undefined, only zero should be written.	NA
15	SLVDESEL	Slave Deselected flag.	0
16	MONRDY	Monitor Ready.	0
17	MONOV	Monitor Overflow flag.	0
18	-	Reserved. Read value is undefined, only zero should be written.	NA
19	MONIDLE	Monitor Idle flag.	0
23:20	-	Reserved. Read value is undefined, only zero should be written.	NA
24	EVENTTIMEOUT	Event time-out Interrupt flag.	0
25	SCLTIMEOUT	SCL time-out Interrupt flag.	0
31:26	-	Reserved. Read value is undefined, only zero should be written.	NA

20.6.8 Master Control register

The MSTCTL register contains bits that control various functions of the I^{2}C Master interface. Only write to this register when the master is pending (MSTPENDING = 1 in the STAT register, Table 294).

Software should always write a complete value to MSTCTL, and not OR new control bits into the register as is possible in other registers such as CFG. This is due to the fact that MSTSTART and MSTSTOP are not self-clearing flags. ORing in new data following a Start or Stop may cause undesirable side effects.

After an initial I2C Start, MSTCTL should generally only be written when the MSTPENDING flag in the STAT register is set, after the last bus operation has completed. An exception is when DMA is being used and a transfer completes. In this case there is no

MSTPENDING flag, and the MSTDMA control bit would be cleared by software potentially at the same time as setting either the MSTSTOP or MSTSTART control bit.

Remark: When in the idle or slave NACKed states (see Table 295), set the MSTDMA bit either with or after the MSTCONTINUE bit. MSTDMA can be cleared at any time.

Table 302. Master Control register (MSTCTL, address 0x4005 0020 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
0	MSTCONTINUE		Master Continue. This bit is write-only.	0
		0	No effect.	
		1	Continue. Informs the Master function to continue to the next operation. This must done after writing transmit data, reading received data, or any other housekeeping related to the next bus operation.	
1	MSTSTART		Master Start control. This bit is write-only.	0
		0	No effect.	
		1	Start. A Start will be generated on the I^{2}C bus at the next allowed time.	
2	MSTSTOP		Master Stop control. This bit is write-only.	0
		0	No effect.	
		1	Stop. A Stop will be generated on the I^{2}C bus at the next allowed time, preceded by a NACK to the slave if the master is receiving data from the slave (Master Receiver mode).	
3	MSTDMA		Master DMA enable. Data operations of the I^{2}C can be performed with DMA. Protocol type operations such as Start, address, Stop, and address match must always be done with software, typically via an interrupt. When a DMA data transfer is complete, MSTDMA must be cleared prior to beginning the next operation, typically a Start or Stop.This bit is read/write.	0
		0	Disable. No DMA requests are generated for master operation.	
		1	Enable. A DMA request is generated for I^{2}C master data operations. When this I^{2}C master is generating Acknowledge bits in Master Receiver mode, the acknowledge is generated automatically.	
31:4	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.9 Master Time

The MSTTIME register allows programming of certain times that may be controlled by the Master function. These include the clock (SCL) high and low times, repeated Start setup time, and transmitted data setup time.

The I2C clock pre-divider is described in Table 300.

Table 303. Master Time register (MSTTIME, address 0x4005 0024 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
2:0	MSTSCLLOW		Master SCL Low time. Specifies the minimum low time that will be asserted by this master on SCL. Other devices on the bus (masters or slaves) could lengthen this time. This corresponds to the parameter t_{LOW} in the I^2C bus specification. I^2C bus specification parameters t_{BUF} and $t_{SU;STA}$ have the same values and are also controlled by MSTSCLLOW.	0x7
		0x0	2 clocks. Minimum SCL low time is 2 clocks of the I^2C clock pre-divider.	
		0x1	3 clocks. Minimum SCL low time is 3 clocks of the I^2C clock pre-divider.	
		0x2	4 clocks. Minimum SCL low time is 4 clocks of the I^2C clock pre-divider.	
		0x3	5 clocks. Minimum SCL low time is 5 clocks of the I^2C clock pre-divider.	
		0x4	6 clocks. Minimum SCL low time is 6 clocks of the I^2C clock pre-divider.	
		0x5	7 clocks. Minimum SCL low time is 7 clocks of the I^2C clock pre-divider.	
		0x6	8 clocks. Minimum SCL low time is 8 clocks of the I^2C clock pre-divider.	
		0x7	9 clocks. Minimum SCL low time is 9 clocks of the I^2C clock pre-divider.	
3	-		Reserved.	0

Table 303. Master Time register (MSTTIME, address 0x4005 0024 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
6:4	MSTSCLHIGH		Master SCL High time. Specifies the minimum high time that will be asserted by this master on SCL. Other masters in a multi-master system could shorten this time. This corresponds to the parameter t_{HIGH} in the I^{2}C bus specification. I^{2}C bus specification parameters $t_{SU;STO}$ and $t_{HD;STA}$ have the same values and are also controlled by MSTSCLHIGH.	0x7
		0x0	2 clocks. Minimum SCL high time is 2 clock of the I^{2}C clock pre-divider.	
		0x1	3 clocks. Minimum SCL high time is 3 clocks of the I^{2}C clock pre-divider .	
		0x2	4 clocks. Minimum SCL high time is 4 clock of the I^{2}C clock pre-divider.	
		0x3	5 clocks. Minimum SCL high time is 5 clock of the I^{2}C clock pre-divider.	
		0x4	6 clocks. Minimum SCL high time is 6 clock of the I^{2}C clock pre-divider.	
		0x5	7 clocks. Minimum SCL high time is 7 clock of the I^{2}C clock pre-divider.	
		0x6	8 clocks. Minimum SCL high time is 8 clock of the I^{2}C clock pre-divider.	
		0x7	9 clocks. Minimum SCL high time is 9 clocks of the I^{2}C clock pre-divider.	
31:7	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.10 Master Data register

The MSTDAT register provides the means to read the most recently received data for the Master function, and to transmit data using the Master function.

Table 304. Master Data register (MSTDAT, address 0x4005 0028 (I2C0)) bit description

Bit	Symbol	Description	Reset value
7:0	DATA	Master function data register. Read: read the most recently received data for the Master function. Write: transmit data using the Master function.	0
31:8	-	Reserved. Read value is undefined, only zero should be written.	NA

20.6.11 Slave Control register

The SLVCTL register contains bits that control various functions of the I^{2}C Slave interface. Only write to this register when the slave is pending (SLVPENDING = 1 in the STAT register, Table 294).

Remark: When in the slave address state (slave state 0, see Table 296), set the SLVDMA bit either with or after the SLVCONTINUE bit. SLVDMA can be cleared at any time.

Table 305. Slave Control register (SLVCTL, address 0x4005 0040 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset Value
0	SLVCONTINUE		Slave Continue.	0
		0	No effect.	
		1	Continue. Informs the Slave function to continue to the next operation. This must done after writing transmit data, reading received data, or any other housekeeping related to the next bus operation.	
1	SLVNACK		Slave NACK.	0
		0	No effect.	
		1	NACK. Causes the Slave function to NACK the master when the slave is receiving data from the master (Slave Receiver mode).	
2	-		Reserved. Read value is undefined, only zero should be written.	NA
3	SLVDMA		Slave DMA enable.	0
		0	Disabled. No DMA requests are issued for Slave mode operation.	
		1	Enabled. DMA requests are issued for I^2C slave data transmission and reception.	
31:4	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.12 Slave Data register

The SLVDAT register provides the means to read the most recently received data for the Slave function and to transmit data using the Slave function.

Table 306. Slave Data register (SLVDAT, address 0x4005 0044 (I2C0)) bit description

Bit	Symbol	Description	Reset Value
7:0	DATA	Slave function data register. Read: read the most recently received data for the Slave function. Write: transmit data using the Slave function.	0
31:8	-	Reserved. Read value is undefined, only zero should be written.	NA

20.6.13 Slave Address registers

The SLVADR[0:3] registers allow enabling and defining one of the addresses that can be automatically recognized by the I^2C slave hardware. The value in the SLVADR0 register is qualified by the setting of the SLVQUAL0 register.

When the slave address is compared to the receive address, the compare can be affected by the setting of the SLVQUAL0 register (see Section 20.6.14).

The I^2C slave function has 4 address comparators. The additional 3 address comparators do not include the address qualifier feature. For handling of the general call address, one of the 4 address registers can be programmed to respond to address 0.

Table 307. Slave Address registers (SLVADR[0:3], address 0x4005 0048 (SLVADR0) to 0x4005 0054 (SLVADR3) (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
0	SADISABLE		Slave Address n Disable.	1
		0	Enabled. Slave Address n is enabled and will be recognized with any changes specified by the SLVQUAL0 register.	
		1	Ignored Slave Address n is ignored.	
7:1	SLVADR		Seven bit slave address that is compared to received addresses if enabled.	0
31:8	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.14 Slave address Qualifier 0 register

The SLVQUAL0 register can alter how Slave Address 0 is interpreted.

Table 308. Slave address Qualifier 0 register (SLVQUAL0, address 0x4005 0058 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset Value
0	QUALMODE0		Reserved. Read value is undefined, only zero should be written.	0
		0	The SLVQUAL0 field is used as a logical mask for matching address 0.	
		1	The SLVQUAL0 field is used to extend address 0 matching in a range of addresses.	
7:1	SLVQUAL0		Slave address Qualifier for address 0. A value of 0 causes the address in SLVADR0 to be used as-is, assuming that it is enabled. If QUALMODE0 = 0, any bit in this field which is set to 1 will cause an automatic match of the corresponding bit of the received address when it is compared to the SLVADR0 register. If QUALMODE0 = 1, an address range is matched for address 0. This range extends from the value defined by SLVADR0 to the address defined by SLVQUAL0 (address matches when SLVADR0[7:1] <= received address <= SLVQUAL0[7:1]).	0
31:8	-		Reserved. Read value is undefined, only zero should be written.	NA

20.6.15 Monitor data register

The read-only MONRXDAT register provides information about events on the I^{2}C bus, primarily to facilitate debugging of the I^{2}C during application development. All data addresses and data passing on the bus and whether these were acknowledged, as well as Start and Stop events, are reported.

The Monitor function must be enabled by the MONEN bit in the CFG register. Monitor mode can be configured to stretch the I^{2}C clock if data is not read from the MONRXDAT register in time to prevent it, via the MONCLKSTR bit in the CFG register. This can help ensure that nothing is missed but can cause the monitor function to be somewhat intrusive (by potentially adding clock delays, depending on software or DMA response time). In order to improve the chance of collecting all Monitor information if clock stretching is not enabled, Monitor data is buffered such that it is available until the end of the next piece of information from the I^{2}C bus.

Table 309. Monitor data register (MONRXDAT, address 0x4005 0080 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset value
7:0	MONRXDAT		Monitor function Receiver Data. This reflects every data byte that passes on the I^{2}C pins, and adds indication of Start, Repeated Start, and data NACK.	0
8	MONSTART		Monitor Received Start.	0
		0	No detect. The monitor function has not detected a Start event on the I^{2}C bus.	
		1	Start detect. The monitor function has detected a Start event on the I^{2}C bus.	
9	MONRESTART		Monitor Received Repeated Start.	0
		0	No start detect. The monitor function has not detected a Repeated Start event on the I^{2}C bus.	
		1	Repeated start detect. The monitor function has detected a Repeated Start event on the I^{2}C bus.	
10	MONNACK		Monitor Received NACK.	0
		0	Acknowledged. The data currently being provided by the monitor function was acknowledged by at least one master or slave receiver.	
		1	Not acknowledged. The data currently being provided by the monitor function was not acknowledged by any receiver.	
31:11	-		Reserved. Read value is undefined, only zero should be written.	NA

20.7 Functional description

20.7.1 Bus rates and timing considerations

Due to the nature of the I^{2}C bus, it is generally not possible to guarantee a specific clock rate on the SCL pin. On the I2C-bus, the The clock can be stretched by any slave device, extended by software overhead time, etc. In a multi-master system, the master that provides the shortest SCL high time will cause that time to appear on SCL as long as that master is participating in I2C traffic (i.e. when it is the only master on the bus, or during arbitration between masters).

Rate calculations give a base frequency that represents the fastest that the I^{2}C bus could operate if nothing slows it down.

20.7.1.1 Rate calculations

SCL high time (in I^{2}C function clocks) = (CLKDIV + 1) * (MSTSCLHIGH + 2)

SCL low time (in I^{2}C function clocks) = (CLKDIV + 1) * (MSTSCLLOW + 2)

Nominal SCL rate = I^{2}C function clock rate / (SCL high time + SCL low time)

Remark: DIVVAL must be ≥ 1.

Remark: For 400 KHz clock rate, the clock frequency after the I^{2}C divider (divval) must be ≤ 2 MHz. Refer to Table 310.

Table 310. Settings for 400 KHz clock rate

Input clock to I2C	DIVVAL for CLKDIV register	MSTSCLHIGH for MSTTIME register	MSTSCLLOW for MSTTIME register
60 MHz	14	2	2
48 MHz	14	0	1
24 MHz	14	0	0
24 MHz	11	0	1
12 MHz	5	0	1

20.7.2 Time-out

A time-out feature on an I^{2}C interface can be used to detect a “stuck” bus and potentially do something to alleviate the condition. Two different types of time-out are supported. Both types apply whenever the I^{2}C block and the time-out function are both enabled, Master, Slave, or Monitor functions do not need to be enabled.

In the first type of time-out, reflected by the EVENTTIMEOUT flag in the STAT register, the time between bus events governs the time-out check. These events include Start, Stop, and all changes on the I^{2}C clock (SCL). This time-out is asserted when the time between any of these events is longer than the time configured in the TIMEOUT register. This time-out could be useful in monitoring an I^{2}C bus within a system as part of a method to keep the bus running of problems occur.

The second type of I^{2}C time-out is reflected by the SCLTIMEOUT flag in the STAT register. This time-out is asserted when the SCL signal remains low longer than the time configured in the TIMEOUT register. This corresponds to SMBus time-out parameter $T_{TIMEOUT}$. In this situation, a slave could reset its own I^{2}C interface in case it is the offending device. If all listening slaves (including masters that can be addressed as slaves) do this, then the bus will be released unless it is a current master causing the problem. Refer to the SMBus specification for more details.

Both types of time-out are generated when the I^{2}C bus is considered busy.

20.7.3 Ten-bit addressing

Ten-bit addressing is accomplished by the I^{2}C master sending a second address byte to extend a particular range of standard 7-bit addresses. In the case of the master writing to the slave, the I^{2}C frame simply continues with data after the 2 address bytes. For the master to read from a slave, it needs to reverse the data direction after the second address byte. This is done by sending a Repeated Start, followed by a repeat of the same standard 7-bit address, with a Read bit. The slave must remember that it had been addressed by the previous write operation and stay selected for the subsequent read with the correct partial I^{2}C address.

For the Master function, the I2C is simply instructed to perform the 2-byte addressing as a normal write operation, followed either by more write data, or by a Repeated Start with a repeat of the first part of the 10-bit slave address and then reading in the normal fashion.

For the Slave function, the first part of the address is automatically matched in the same fashion as 7-bit addressing. The Slave address qualifier feature (see Section 20.6.14) can be used to intercept all potential 10-bit addresses (first address byte values F0 through F6), or just one. In the case of Slave Receiver mode, data is received in the normal fashion after software matches the first data byte to the remaining portion of the 10-bit address. The Slave function should record the fact that it has been addressed, in case there is a follow-up read operation.

For Slave Transmitter mode, the slave function responds to the initial address in the same fashion as for Slave Receiver mode, and checks that it has previously been addressed with a full 10-bit address. If the address matched is address 0, and address qualification is enabled, software must check that the first part of the 10-bit address is a complete match to the previous address before acknowledging the address.

20.7.4 Clocking and power considerations

The Master function of the I^{2}C always requires a peripheral clock to be running in order to operate. The Slave function can operate without any internal clocking when the slave is not currently addressed. This means that reduced power modes up to Power-down mode can be entered, and the device will wake up when the I^{2}C Slave function recognizes an address. Monitor mode can similarly wake up the device from a reduced power mode when information becomes available.

20.7.5 Interrupt handling

The I2C provides a single interrupt output that handles all interrupts for Master, Slave, and Monitor functions.

20.7.6 DMA

Generally, data transfers can be handled by DMA for Master mode after an address is sent and acknowledged by a slave, and for Slave mode after software has acknowledged an address. In either mode, software is always involved in the address portion of a message. In master and slave modes, data receive and transmit data can be transferred by the DMA. The DMA supports three DMA requests: data transfer in master mode, slave mode, and monitor mode.

UM11607

Chapter 21: I3C bus interface

Rev. 3 — April 2023 User manual

21.1 How to read this chapter

The MIPI Alliance Improved Inter-Integrated Circuit (MIPI I3C) improves upon the use and power of I2C, and provides an alternative to SPI for mid-speed applications.

The I3C bus protocol supports:

- In-band interrupts (IBI). These interrupts go from target to controller without extra wires, and the controller knows which target sent the interrupt.
- Common Command Codes (CCC)
- Dynamic addressing
- Multi-master / multi-drop
- Hot-Join
- I2C compatibility

The I3C peripheral supports all required and most optional features of the MIPI Alliance Specification for I3C, v1.0 and v1.1, except for ternary data rates (HDR-TSP and HDR-TSL).

Note: Terminology in this chapter has been updated to align with MIPI I3C Specification v1.1.1.

Updated term	Deprecated term
Controller	Master
Target	Slave
Controller Request (CR)	Master Request (MR)

21.2 Block diagram

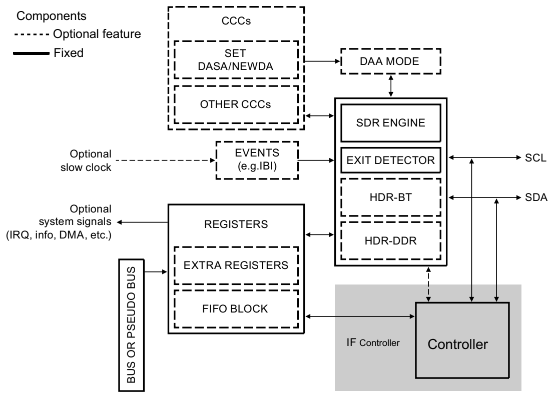


Fig 34. I3C block diagram

21.3 Features

- Two-wire multi-drop bus capable of 12.5 MHz clock speeds, with up to 11 devices.
  - Uses standard pads with 4 mA drive..
  - Dynamically assigns target addresses, and targets do not require static addresses. However, targets may have an I2C static address assigned at start-up, so the target can operate on an I2C bus. By default, I3C supports seven-bit I2C-style addresses.
  - Allows targets to use the inbound SCL clock as the peripheral clock (instead of the clock from the controller) so devices can have slow or inaccurate clocks internally.
  - Allows simple targets, such as temperature sensors, to have no internal clock.
  - I3C controller supports handoff from Open Drain to Push-Pull mode for ACK to data transfer.
  - Normally the controller terminates the read, but for I3C, the target can also end the read.
- In-Band Interrupts (IBI) allow targets to send notifications to a controller.
  - Can be prioritized. When multiple targets send interrupts to a controller at the same time, the order is resolved.

    Dynamic addresses establish the priority of the targets, so the controller controls the priority of the targets. Targets with lower-value dynamic addresses are higher priority level IBIs.
  - Can start interrupts even when the controller is not active on the bus. No free-running clock is needed, but starting an interrupt requires a Bus Available condition.
  - Can resolve an initial event via a time-stamping option, not requiring an interrupt.
- Built-in commands are in a separate space. These commands do not collide with normal controller-to-target messages.
  - Controls bus behavior, modes and states, low power state, inquiries, and more.
  - Has additional room for new built-in commands to be used by other groups.
- Organized forms of multi-controller modes:
  - Secondary controllers, which use clean handoffs between different controllers.
- Hot-join onto I3C bus allows devices to connect to the bus later than when the bus starts.
  - Enables a device or module to get onto the I3C bus when it woke up after power-up or was physically inserted onto the I3C bus.
  - Provides a clean method for notification when new devices or modules get onto the I3C bus.
- Can use both I2C and I3C buses.
  - I3C supports specific legacy I2C devices on the bus.
  - I3C target devices can operate on I2C buses.
  - Supports bridging to I2C, SPI, UART, and other busses.
- Higher data rate modes are available.

  - Has a High Data Rate - Double Data Rate (HDR-DDR) mode, which is double the data rate of SDR.
  - Only the controller and the specific target must support the higher data rate. The other targets can ignore it.

The I3C peripheral supports the full I3C feature set, except for the ternary data rates (HDR-TSP and HDR-TSL) and peer-to-peer messaging, which are not supported.

21.4 Functional description

The following sections describe functional details of the I3C module.

21.4.1 Operating modes

This section describes all functional operation modes of the I3C module.

21.4.1.1 Controller and target roles for I3C

The I3C protocol defines these roles for devices on the I3C bus:

- Main Controller. Controls the I3C bus.
- Secondary Controller. Controls I3C with permission from the Main Controller, and returns control of the bus to the Main Controller when the Secondary Controller has finished its tasks.
- Target. Responds to commands from any I3C controller.
- I2C target. Responds to commands from any I2C target.

  The I3C peripheral contains both controller and target components and can be parameterized to be either controller or target, or both controller and target. However, if the I3C is chosen to be a controller, then the I3C peripheral supports Target mode to facilitate handoffs to multiple controllers.

In general, any I3C controller can be a controller or a target, because a controller becomes a target when giving over control to another controller. The only exceptions to this rule occur when using a point-to-point controller or when using a controller that never shares controllership.

21.4.1.2 Master requirements

Controller mode uses software that supports the requirements of the controller (including as a Secondary Controller):

- Managing the Enter Dynamic Address Assignments (ENTDAA) assigning dynamic addresses to each target. This process is supported by the I3C peripheral but requires the software to make choices.
- Driving the serial data (SDA) signal in both Open Drain and Push-Pull mode. After Start, SDA is in Open Drain mode and after Repeated Start it is in Push-Pull mode.
  - SDA is subject to arbitration, because both controller and target can drive the SDA line. Arbitration is also useful for handoff from controller to target.
  - The ninth bit of SDA controller write data is an odd parity bit.
- Building a table of targets and their capabilities to control which actions and commands may be sent to the targets.
- Managing requests such as IBI and controller request (handoff).
- Adjusting clock speed or write-to-read timing to match the limitations of the target. This adjustment can be done in hardware using dividers and uneven duty cycles, but the software must decide.
- Adjusting the maximum data length.
- Being a target after a controller handoff to another controller.

Additionally, a controller needs three pins unless the SDA pad includes a precise, high-strength pullup. The three-pin model allows one pad/pin to enable a system-provided pullup that is sized to system requirements.

The controller needs an accurate clock, capable of running at a frequency that is a multiple of a frequency between 11 MHz and 12.5 MHz. For example:

- 24 MHz (multiple of 12 MHz)
- 48 MHz (multiple of 12 MHz)
- 25 MHz (multiple of 12.5 MHz)
- 50 MHz (multiple of 12.5 MHz)
- 33 MHz (multiple of 11 MHz)
- 66 MHz (multiple of 11 MHz)

Greater multiples can also be used.

21.4.1.3 I3C target acts like I2C target on I3C buses

If it is assigned an I2C static address, the I3C target acts like an I2C device when it first turns on. If the I3C target is placed on an I2C bus with an I2C controller, then the I3C target stays in I2C mode and operates normally. The software is aware that the I3C target is in I2C mode, because:

- There has not been a SSTATUS[DACHG] interrupt indicating that a dynamic address was assigned.

For full I2C support of Fast Mode (FM) and Fast-mode Plus (Fm+), the pads must support a 50-ns spike filter. This filter must be turned off when the I3C 7Eh broadcast address is received (indicating an I3C controller). You can turn off the spike filters via hardware using the raw net indicating that the address was received or via software. A 50-ns spike filter is not needed for the I3C to operate on an I2C bus. Therefore, the spike filter is not a requirement for the I3C peripheral.

Depending on the configuration, the block supports most I2C features, except for clock stretching. Supported features include extended 10-bit address, DeviceID, software reset, and high-speed mode (affects pads).

21.4.1.4 How a target rejoins the I3C bus

When a target tries to rejoin the I3C bus following a power-up or hard reset, the target needs a new Dynamic Address (DA). The target can rejoin the I3C bus in these ways:

- If the dynamic address is lost, Hot-Join is used.
- If the dynamic address is retained in the peripheral (for example, with state retention flip-flops), SCONFIG[OFFLINE] is used (see below).

When SCONFIG[SLVENA] is 1, SCONFIG[OFFLINE] may become 1. This setting causes the peripheral to rejoin the bus safely. It does so by ensuring that the I3C bus is not in HDR mode, using the same approach as TE0 or TE1 exit. The peripheral waits for the HDR Exit Pattern, or for 60 μs of SCL and SDA lines not changing, whichever occurs first.

After using SCONFIG[OFFLINE], the I3C peripheral cannot safely use IBI until SSTATUS[STOP] is 1. This status ensures that the next START is safe to use for IBI.

If the application must perform an IBI, it should wait for SSTATUS[STOP] to become 1, or for SCL and SDA lines to remain high for 200 μs. This process can be done using the SSTATUS and SINTSET controls.

- If SSTATUS indicates that the bus is not busy and the peripheral interrupts on START or STOP, use a timer to measure 200 μs. If the timer finishes with no START or STOP, it is safe to use IBI.
- If a START causes an interrupt, then the timer should be turned off and the application should wait for a STOP.
- If a STOP causes an interrupt, then it is safe to use IBI.

21.4.1.5 Using the I3C controller for I2C and I3C

The I3C controller operates with this set of built-in capabilities with the application flow:

- Built-in Enter Dynamic Address Assignment (ENTDAA) mechanism to simplify the assignment of dynamic addresses to targets. This feature is used when Set Dynamic Address from Static Address (SETDASA) is not available.
- Request for START + Address with IBI support, including both I2C and I3C modes.
- SDR Write flow via FIFO, with automatic parity in I3C and ACK or NACK detection in I2C..
- SDR Read flow via the FIFO, with an automatic NACK generator for I2C, and optional use of a terminate generator for I3C.
- Request for repeated START + Address or STOP when finished with previous, including both I2C and I3C.
- Auto-IBI mode, which responds immediately to a target-initiated IBI and can be used when in sleep or deep sleep.
- Special message mode for SDR and DDR to simplify for DMA use.
- Automated SCL, SDA, pullup, and High-Keeper controls.
- I2C and I3C Frequency and duty cycle configurations from functional clock.

Note: The Controller Configuration (MCONFIG) and Controller In-band Interrupt Registry and Rules (MIBIRULES) registers must be configured before using the controller.

21.4.1.6 Protocol modes and states

The following modes are activated in I3C:

- I2C mode is the default mode on startup.
  - If no I2C Static Address is used, this default has no effect other than to track frames looking for I3C transitional frames.
  - If an I2C Static Address is used, the device can interact with the I2C bus controller using the address.
- While in I2C mode, I3C Transitory Frame mode occurs whenever the I3C broadcast address is received (7Eh). In particular:
  - This mode occurs when 7Eh is broadcast followed by a SETDASA from the controller. If there is a Static Address match or 01h, the target is assigned a Dynamic Address and it enters I3C SDR mode.

  - This mode occurs when 7Eh is broadcast followed by an ENTDAA from the controller. If the target can send a 48-bit provisioned ID (48b ID), a Dynamic Address is assigned, and the target enters I3C SDR mode.
  - If a Hot-Join arbitrated event occurs (sending 02h when the controller generates another address such as 7Eh), there is a conceptual Hot-Join mode. ENTDAA or SETDASA sets a Dynamic Address to resolve the event, and the target enters I3C SDR mode.

Note: An I3C target can operate as a normal I2C target only when it has a Static Address. Otherwise, it matches nothing in I2C and waits for the above events.

- I3C SDR mode is the standard mode once I3C activates, which is defined by a Dynamic Address being assigned. This mode is the resting mode of I3C, and all devices are normally in SDR mode.
  - RSTDAA CCC causes an exit from SDR mode and a return to I2C mode. This transition is not normally used.
  - SETNEWDA does not affect the SDR mode, it only changes the Dynamic Address of an I3C device that already had a Dynamic Address assigned.
  - When in SDR mode, the device ignores messages to or from its original I2C Static Address.
- I3C CCC Command submode of type Direct or Broadcast.
  - Exit the Broadcast CCC submode by a repeated START or by a STOP.
  - Exit the Direct CCC submode by a 7Eh broadcast address after a repeated START or by a STOP.
  - I3C Dynamic Address Assignment (DAA) CCC command enters DAA mode. This mode is a special mode for dynamic addressing. Use STOP to exit. If a target has a Dynamic Address, the command is ignored.
- I3C SETDASA CCC command enters the Static Address Match submode. This mode allows matching the Static Address of the device (if any).
  - The command is ignored when a target has a Dynamic Address, or when the target in I2C mode has no Static Address.

Note: The special point-to-point address is also matched.

- I3C HDR modes are activated by ENTHDRn CCC commands, where n represents the type of HDR (0 to 7). This mode is valid until an HDR Exit Pattern, whether HDR mode for a slave is supported or not.
  - DDR included.
  - Exit-pattern detection must always be on to prevent errors. Alternatively, it could be set to only activate on ENTHDR.
- I3C HDR-BT modes are also activated by ENTHDRn CCC commands and exited by HDR Exit Pattern.
- Machine learning (ML) data transfer is available for data transfers between ML-capable I3C Devices that have ML functionality enabled.
- A special HDR-DDR syntax is used for transmitting CCC in the HDR-DDR protocol.

- Optionally, multiple I3C target devices can share a single group address. This sharing allows a controller device to send an I3C message to all target devices in a group simultaneously rather than one at a time.
- Internal states such as IBI/no-IBI and low-power mode, are flagged internally. See SSTATUS[EVDET]. These states are exported to the application and affect the engine to prohibit it from performing a prohibited operation.

21.4.1.7 Address match

For an address match to occur, the target matches the I3C broadcast address and either the I2C-style Static Address or the I3C Dynamic Address.

- I3C broadcast address, 111 1110b (written as 7Eh in spec).
- I2C-style Static Address, but only if:
  – The device has a Static Address.
  – Not in I3C mode. Only matches until the ENTDAA or the SETDASA command has assigned a Dynamic Address.

  Note: SETDASA matches the Static Address or the specialone-controller-to-one-target point-to-point address.
- I3C Dynamic Address once assigned by ENTDAA or SETDASA or modified by SETNEWDA.

The address match is inactive (just listening) for all repeated START commands and also for START, unless an IBI, CR, or Hot-Join has been activated. If inactive, it simply tries to match. If not matched, it waits for a repeated START or a STOP. If an IBI, CR, or Hot-Join has been activated, the arbitration mechanism is used for START (but never for a repeated START).

If matching the 7Eh broadcast address, it listens for a CCC until the next repeated START or a STOP.

21.4.2 Operations

This section describes the operations of the I3C module.

21.4.2.1 Reading and writing I2C messages using normal method

Note: I2C FM and I2C FM+ modes are supported for legacy I2C devices as per the I3C standard specifications. See I2C configuration for meeting timing requirement for FM and FM+ modes.

The normal method is as follows.

1. Set up interrupts.:
   a. MCTRLDONE: Indicates when the I3C module completes an MCTRL request.
   b. COMPLETE: Indicates when data has finished sending or being received.
   c. RXPEND: For read operations. Used with the Controller Data Control (MDATACTRL) register to set the FIFO trigger. Can also be used to allow DMA to read out data..
   d. TXNOTFULL: For write operations. Used with the MDATACTRL register to set the FIFO trigger. Can also be used to allow DMA to supply data..

e. IBIWON: Indicates that an IBI, CR, or HJ has won the arbitration on a header address.

f. MERRWARN: Indicates causes of errors and warnings.

2. Configure these fields in the Controller Control (MCTRL) register simultaneously:
   a. Write 1 to MCTRL[REQUEST] (EmitStartAddr).
   b. Write 1 to MCTRL[TYPE] (I2C).
   c. Set MCTRL[IBIRESP] to respond to IBIs in your chosen manner.
   d. Write 1 to MCTRL[DIR] for read, or write 0 for write.
   e. Write the static address of the I2C target to MCTRL[ADDR].
   f. For read operations, you can set MCTRL[RDTERM] to the maximum length to auto-terminate, or you can set it to stop as the data is read out. For example, set it to 1 (with REQUEST = 0) to stop after the next character.
3. Write or read the data.
   – For write operations, write the Controller Write Data Byte (MWDATAB) register for each byte before the last byte, then write the Controller Write Data Byte End (MWDATABE) register for the last byte.
   – This operation can be done or started before setting up interrupts.
   – If there is more data than the FIFO can hold, use the TXNOTFULL interrupt based on the trigger level. This interrupt allows the application to provide more data or to use DMA.
   – For read operations, wait for RXPEND, and then read out data via the Controller Read Data Byte (MRDATAB) register. DMA may also be used.
4. On COMPLETE, the message may be ended with a STOP, or a new message started with a repeated START.
   a. Write 2 to MCTRL[REQUEST] (EmitStop to STOP). Then wait for MCTRLDONE status to be asserted for its completion. (While sending STOP in I2C mode MCONFIG[ODSTOP] should be 1 and MCTRL[TYPE] should be 1.)
   b. Write 1 to MCTRL[REQUEST] (EmitStartAddr to restart). IBI is not possible in this case.

21.4.2.2 Reading/writing I3C messages using normal methods (SDR and HDR-DDR)

The normal method for I3C is the same as the method for I2C with a few differences.

1. Set up interrupts:
   – MCTRLDONE: Indicates when the I3C module completes an MCTRL request.
   – COMPLETE: Indicates when data has finished sending or being received.
   – RXPEND: For read operations. Used with the Controller Data Control (MDATACTRL) register to set the FIFO trigger. Can also be used to allow DMA to read out bytes
   – TXNOTFULL: For write operations. Used with the Controller Data Control (MDATACTRL) register to set the FIFO trigger. Can also be used to allow DMA to supply bytes.
   – IBIWON: Indicates that an IBI, CR, or HJ has won the arbitration on a header address.

   – MERRWARN: Indicates causes of errors and warnings for software to check.

2. Set up the Controller Control (MCTRL) register.

   Option 1: Configure the MCTRL register fields simultaneously in this way:

   a. Write 1 to MCTRL[REQUEST] (EmitStartAddr).
   b. Write 0 to MCTRL[TYPE] (for I3C SDR mode) or write 2 (for DDR mode)
   c. Set MCTRL[IBIRESP] to respond to IBIs in your chosen manner.
   d. Write 1 to MCTRL[DIR] for read, or write 0 for write.
   e. Write the dynamic address of the I3C target to MCTRL[ADDR].
   f. For read operations, you can set MCTRL[RDTERM] to the maximum length to auto-terminate..
   g. For write operations, pre-writing the data (MWDATAB or MWDATAH) is preferred to ensure that there are no time delays waiting on the data.
   h. For DMA with MCTRL, use Controller Write Byte Data 1(to bus) (MWDATAB1) register.

      Note: HDR-DDR mode requires writing an 8-bit command value for read or write. This value must be written into the TX FIFO via Controller Write Data Byte (MWDATAB). The END bit is not used for this byte.

   Option 2: This option is preferred when stopped (bus free condition) in SDR mode, and not in HDR-DDR mode. It allows any target to issue an IBI, and it avoids collisions with an IBI address. Also, it is faster (when the MSB of the Dynamic Address is always 0).

   Configure the MCTRL register in this way:

   a. Write 1 to MCTRL[REQUEST] (EmitStartAddr). No pre-written transmit data can be in the FIFO.
   b. Write 0 to MCTRL[TYPE] (I3C).
   c. Set MCTRL[IBIRESP] to respond to IBIs in your chosen manner.
   d. Write 0 to MCTRL[DIR].
   e. Write 7Eh to MCTRL[ADDR].
   f. Wait for MCTRLDONE (via interrupt, for example), then proceed as in Option 1. The Option 2 method has advantages for IBIs when 7Eh is sent on START (but not on repeated STARTs).

3. Write or read the data.

   – For write operations, write the MWDATAB register for each byte before the last byte, then write the Controller Write Data Byte End (MWDATABE) register for the last byte. For HDR-DDR, the byte with END must be even (second, fourth, sixth, and so on) because DDR uses byte pairs.
   – This operation can be done or started before step 1 (REQUEST = 1) of the Option 1 method, but not before the Option 2 method.
   – If there is more data than the FIFO can hold, use the TXNOTFULL interrupt based on the trigger level, to allow the application to provide more; or use DMA.
   – For read operations, wait for RXPEND, and then read out data via the Controller Read Data Byte (MRDATAB) or Controller Read Data Halfword (MRDATAH) register. DMA may also be used. If using DMA, read using same registers.

4. On COMPLETE, the message may be ended with STOP (or EXIT in HDR mode). Alternatively, a new message may be started with a repeated START (or HDR-Restart in HDR mode).
   – In SDR mode, write 2 to MCTRL[REQUEST] (EmitStop to STOP). Then wait for MCTRLDONE status to be asserted for its completion.
   – For HDR mode, write 6 to MCTRL[REQUEST] (ForceExit) to end HDR mode. Then wait for MCTRLDONE status to be asserted for its completion. When sending the HDR exit pattern, MCONFIG[ODSTOP] must be 0.
   – Write 1 to MCTRL[REQUEST] (EmitStartAddr to start another message). IBI is not possible in this case.

21.4.2.3 Sending a CCC to one or all I3C slaves

The normal Common Command Code (CCC) method is to use an I3C write operation with an address of 7Eh. The first byte is the CCC..

- For Broadcast type, any remaining bytes are sent with the CCC. This operation ends with a STOP or a repeated START and 7Eh.
- For Direct type, this write is followed only by a CCC (or by a CCC and a defining byte, if required by the CCC).

  These bytes are followed by a repeated START and the address of the I3C target (for SETDASA, this address is its I2C Static address). This sequence may be repeated with more repeated STARTs and addresses until done. It may end in STOP or a repeated START and 7Eh. After the repeated start and target address, the values are read or written depending on the CCC.
- I3C provides interrupts (by hardware) for Unhandled and Handled CCC when received by the target. Its state is reflected in the Target Status (SSTATUS) register.

  The I3C spec requires I3C controllers to emit a single START, 7E/W sequence with both SCL High and SCL Low half periods at full open-drain timing (for example, 200 ns). This sequence allows I3C targets acting as I2C legacy devices to turn off their I2C 50-ns spike filters, if they have them.

NOTE: START, 7E/W is notation indicating a START command, followed by the 7Eh broadcast address, followed by a write command.

After that sequence, addresses following START may be sent with Open-Drain Low (for example, 200 ns) but High of Push-Pull timing (for example, 40 ns). The I3C controller should emit this START, 7E/W sequence with MCONFIG[ODHPP] = 0. ODHPP is Open-Drain High period at Push-Pull speeds, and MCONFIG[ODHPP] is usually 1.

NOTE: MCONFIG[ODHPP] is ignored when sending a message to an I2C legacy device on an I3C bus.

NOTE: Each repeated START is just a new EmitStart request. This can be chained by interrupt or pushed by message model using DMA.

21.4.2.4 In-band interrupt (IBI) handling

An IBI occurs when a target sends its address after a START, and that address is numerically the lowest. That is, it is lower than the address sent by the controller and addresses sent by any other targets. When the controller sends 7Eh, the controller always loses the arbitration, by design.

The IBI can occur unexpectedly when any new START (not a repeated START) is sent. The IBI can also occur in response to a target pulling SDA low. This condition can occur in one of two ways:

- The target has set the request to AutoIBI mode, so the IBI occurs automatically. The engine sends 7Eh to allow the target to win the arbitration.
- The application receives the SLVSTART (target START request) interrupt, so it sends 7Eh.

The IBI response is configured by MCTRL[IBIRESP], which can be set to:

- NACK. Always reject the IBI.
- ACK. Always accept the IBI
  - The Controller In-band Interrupt Registry and Rules (MIBIRULES) register must be configured so that the engine knows whether bytes follow.
  - If IBI bytes follow (also known as IBI Mandatory byte), then the COMPLETE field does not become 1 when IBIWON becomes 1. RXPEND becomes 1 for one or more bytes in the Receive FIFO. When the last byte is received, then COMPLETE becomes 1. The controller automatically stops IBI data after nine total bytes (including Mandatory Data byte). MCTRL[RDTERM] can be used with MCTRL[REQUEST] = 0 to end sooner. I3C supports address ACK to Mandatory Byte transition during IBI from Open_Drain (controller ACKs the target address) to Push Pull (target sends SDR data).
- Manual. Allow the decision to be made by the application on a case-by-case basis.
  - The application rewrites it when stopped, pending an IBI.
  - The application can ACK or NACK the IBI based on the IBI address in the MSTATUS register.
  - This mode chooses whether there is an IBI byte or not when accepting.

Accurate timestamping of I3C target data is supported in I3C though Async Mode 0. In I3C Asynchronous Timing Control mode, a target device timestamp event occurs within that target. The target notifies the controller about the event by generating an IBI.

21.4.2.5 Assigning dynamic addresses to I3C devices

If Dynamic Addresses (DAs) are all assigned below 40h (seven-bit values from 3Fh down to 03h, except where not allowed), then MIBIRULES[MSB0] can be 1. This setting optimizes the START timing. When dynamic addresses are assigned, the Controller In-band Interrupt Registry and Rules (MIBIRULES) register must be programmed based on which I3C target uses IBI bytes (known from Bus Characteristics Register (BCR)).

Any Set Dynamic Address from Static Address (SETDASA) assignments can be done via the normal Common Command Code (CCC) model. These assignments cannot be done when using the static address for the directed part.

There is a built-in mechanism to process the Dynamic Address Assignment (DAA) mode

1. Set up interrupts. Normally:

- MCTRLDONE. Indicates when a target has sent its ID and BCR or DCR, and a new DA is needed.
- COMPLETE. Indicates when DAA is done (NACKed by all targets).
- RXPEND. Indicates the reading of the IDs of the targets. Can also be used to allow DMA to read out bytes.
- IBIWON. Indicates when an IBI has occurred, which should not be possible normally
- MERRWARN. Indicates an error.

2. Configure the Controller Control (MCTRL) register.
   - Write 4 to MCTRL[REQUEST] (ProcessDAA).
   - Set MCTRL[IBIRESP] to IBI response, if possible. If it is not possible, set to the first target assignment.
3. Wait for the MCTRLDONE interrupt, while reading ID using the RXPEND interrupt. If MSTATUS[STATE] = 5 (DAA mode) and MSTATUS[BETWEEN] = 1, it is waiting for a DA for the target whose ID was read.
   - Write the dynamic address into the Controller Write Data Byte (MWDATAB) register using bits 6:0, such as 14h for DA = 7'h14.
   - Write 4 to MSTATUS[STATE] (ProcessDAA again). This option writes the DA, then moves to the next DA..
   - If MSTATUS[COMPLETE] = 1 and MSTATUS[STATE] = 0, then all targets have been assigned.
   - If MSTATUS indicates NACK after writing a new DA, the DA was not accepted by the target. The next step is to write 2 to MCTRL[REQUEST] (EmitStop) and start over. It is also acceptable to write 4 to MCTRL[REQUEST] (ProcessDAA again).

21.4.2.6 Using Controller Message mode

Controller Message mode is intended for use with DMA, though Message mode can also be used by the processor. In Message mode, all writes are to the same location, including the control and the data. Reads occur from an associated location.

NOTE: The data writes for message mode work in the same way as the halfword access registers Controller Write Data Halfword (MWDATAH) and Controller Read Data Halfword (MRDATAH)..

To send a message via Single Data Rate (SDR), follow these steps (from DMA or processor)::

1. Write the control request to the Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL) register. This request includes:
   - The address
   - I2CRead or write. or I3C
   - The count of bytes to process
   - How to end (on STOP or ready for repeated START)
2. Process the Data:

- For write operations, write the rest of the data to Controller Write Message Data in SDR mode (MWMSG_SDR_DATA). Use the DMA trigger (or interrupt) to keep the Transmit FIFO full, and the controller stops using the data when the program count MWMSG_SDR_CONTROL register reaches 0..
- For read operations, a DMA trigger (or interrupt) indicates when the RX FIFO is ready to be read out via the Controller Read Message in SDR mode (MRMSG_SDR) register.

3. In-band Interrupt (IBI) behavior is selected in MCTRL[IBIRESP].
4. To exit Message mode with the original message, use END type STOP (MCTRL[REQUEST] = 2) with a zero-length message or by using the MCTRL register.

To send a message via Double Data Rate (DDR), follow these steps (from DMA or processor)::

1. Write the control request to the Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL) register. This request includes:
   - The count of byte-pairs.
   - How to end (on HDR Exit or ready for HDR Restart)
2. Write the second control data to MWMSG_DDR_CONTROL. This second write includes:
   - The address.
   - Read or write.
   - The seven-bit command value.
3. Process the Data:
   - For write operations, write the rest of the data to MWMSG_DDR_DATA. Use the DMA trigger (or interrupt) to keep the Transmit FIFO full, and the controller stops using the data when the program count MWMSG_DDR_CONTROL register reaches 0.
   - For read operations, a DMA trigger (or interrupt) indicates when the RX FIFO is ready to be read out.
4. In-band Interrupt (IBI) behavior is selected in MCTRL[IBIRESP].
5. To exit DDR Message mode with the original message, use END type Exit (MCTRL[REQUEST] = 6) with a zero-length message or by using the MCTRL register.

21.4.2.7 Handing off controllership to another target and getting it back

To hand off controllership, the controller can do one of two things:

- Wait for a Controller Request (CR) (that is, an MSTATUS[IBIWON] = 1 interrupt), indicating that the CR is using MSTATUS[IBITYPE].
- Push the request manually.

  In either case, the controller sends a GETACCMST request, which is a directed GET informing the target that the target is being assigned controllership. If the target accepts this request, it returns its Dynamic Address in bits 7:1 and the negative parity of its dynamic address in bit 0. A STOP must be issued, and MCONFIG[MSTENA] must be set to 2 (switching to the Target mode).

To gain controllership, the target sends an CR using the SCTRL register.

- If MCONFIG[MSTENA] is 2, the GETMSTACC CCC accepts.
- If MCONFIG[MSTENA] is not 2, the GETMSTACC CCC refuses.

After controllership has been granted, MSTATUS[NOWMASTER] is 1. The application must enable the MINTSET[NOWMASTER] bit, so the application is interrupted when a controllership transfer occurs.

21.4.2.8 Controller engine flow diagram

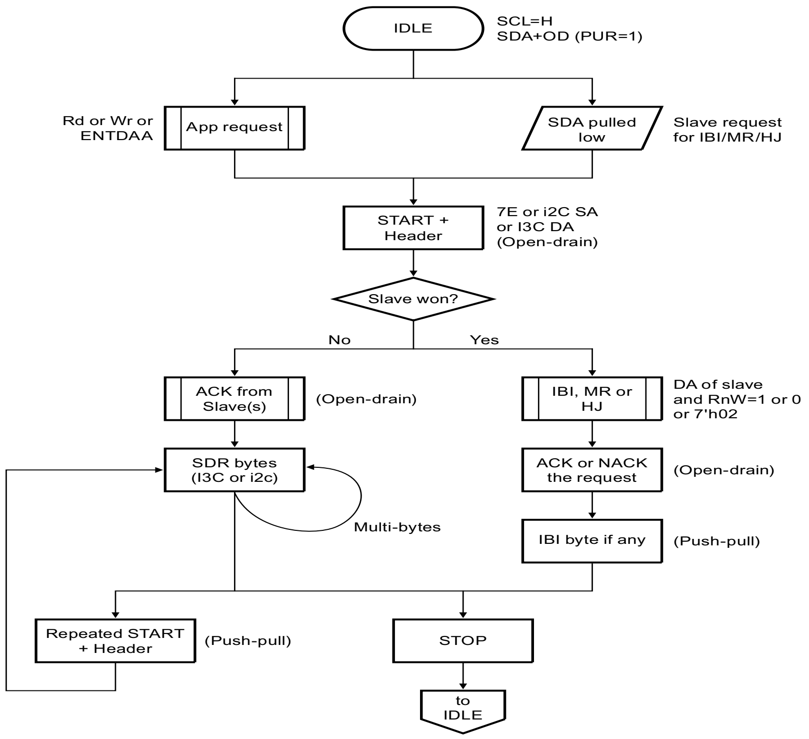


Fig 35. Controller engine flow diagram for SDR (HDR flow progresses from SDR bytes)

21.4.2.9 Target data write from controller

When the address match is valid for the target Dynamic Address and the type is W (write from controller), the target ACKs the address. It does so unless some condition prevents it, such as a full buffer. On ACK, the WRITE state is entered. Once ACKed, it waits for each complete data byte. This process occurs in two steps:

1. Eight bits are clocked in and stored in the next buffer location while in the WRITE state.
2. On the ninth bit:
   - In I3C mode, the W9TH state (In I3C the ninth data bit written by the Controller is the Parity of the preceding eight Data bits.) is used and the parity is checked. The buffer is marked as complete with or without a parity error.

- In I2C mode, the W9TH state (In I2C, the ninth Data bit written by the Controller is an ACK by the Target.) is used and the data is ACKed and stored.

When the address match is valid for the 7Eh broadcast address, the target acknowledges it. On the first data complete, it sets the in_ccc Broadcast or Direct flag (based on bit 7 of the command). If recognized, it parses the command, setting that bit as well. The recognized commands are for Dynamic Address work and modes. The CCC and DAA blocks handle the workload afterward. If not supported, then the commands are passed up to the system.

21.4.2.10 Target data write to controller

When the address match is valid for the target Dynamic Address, and the type is R (read by controller), the target ACKs the address. The target does so unless there is no data waiting for the read. On ACK, the READ state is entered.

Once ACKed, it emits the data byte at a time, with the ninth bit using R9TH state. (In I2C, the ninth Data bit from Target to Controller is an ACK by the Controller. In I3C this bit allows the Target to end a Read, and allows the Controller to Abort a Read.)

- In I3C mode, the T bit is emitted to allow the device to indicate whether this byte is the last (from input signal from upper layers). If it is not last byte, the controller may terminate the read via the T bit.
- In I2C mode, the ninth bit allows the controller to terminate via NACK, or else it allows the read to continue via ACK. On completion of each byte read, the done signal is pulsed to get the next byte.

If the read is terminated by the controller unexpectedly (abort in I3C, NACK before END marked), the application is notified.

21.4.3 Clocks

The controller block has two basic clocks feeding:

- The system clock, which controls the access to the memory mapped registers.
- A functional clock (FCLK), which is used to generate the SCL clock rate on the I2C or I3C bus.

I3C supports adjusting clock frequency and accuracy via the Target Time Control Clock (STCCLOCK) register.

CLK_SLOW is used to check conditions like Bus Availability, Bus Free for IBI, or Hot-Join. The slow clock helps to save power.

- To determine the Bus Available condition for IBI, the target needs a clock to generate the ~1 μs timing. To support this requirement, a slow clock (CLK_SLOW) can be provided to save on power.
- Additionally, the block provides a slow clock gate (slow_gate). If CLK_SLOW can be turned off, slow_gate is 1.
- I3C supports a counter field, SCONFIG[BAMATCH], to calculate the Bus Available condition. BAMATCH provides the count of the slow clock. This field counts 1 μs or more to allow an IBI to drive SDA low when the controller is free. The maximum width and maximum values are controlled by the I3C module.

To provide the slow clock, the system has three basic choices:

- Use the same clock as CLK, which may be divided. If it is variant, the slow match register can be used to change the match rule.
- Provide a truly slow clock, which makes it inexpensive in power to count 1 µs or more.
- Provide some other clock (other than CLK or a truly slow clock), but use the slow_gate signal to gate the clock near its source.

I3C supports a timeout when stalled too long in a frame. This timeout occurs when:

- The Transmit FIFO or Receive FIFO is not handled and the bus is stuck in the middle of a message.
- No STOP was issued and the bus is between messages.
- IBI manual is used and no decision was made.

This model is used with the parameters for the width of the counter and the match rule.

21.4.4 Resets

The I3C resets are:

- Global/system reset fed into block. Everything is reset by this global/system reset. Release is assumed to be synchronized to the system clock (if provided). It is not synchronized to SCL or SDA. Those signals should not be active when the block is released, otherwise I3C would be in Hot-Join mode and therefore not active.
- A software reset is provided by the I3C block.
- STOP state is reset by a START as well.

21.4.4.1 Target Reset RSTACT CCC

The Target Reset mechanism is designed to allow an always-on tiny block to monitor SDA and SCL for a specific pattern based on HDR Exit - extended with repeated START and STOP at the end. This detector can be used to wake or reset the device or just a peripheral, as well as do nothing for any specific reset. The normal use is that the controller emits a RSTACT CCC message saying not to reset when this pattern is emitted just after; devices which are broken reset (since they miss the CCC) and devices in deepest sleep (for example, an unpowered core domain) wake (since they also miss the CCC).

The 1st default action is only to reset the peripheral. If that does not work, the next time will reset the chip/system.

The controller may also request a specific action, which the target can dial in if it supports that (for example, reset the peripheral only).

The Target Reset is connected into the system as shown in below figure. Note that it is outside of the I3C Peripheral, and may be in an always-on domain. Sleep/wake signals are not shown. The common connections are as follows:

Table 311. Target resets

Reset detector	Connect to IP	Connect to System	Notes
I3c_slave_active	raw_slvr_isslave if controller + target	If I3C can be unused	Connect to enable if can be unused — should probably not clear for safety reasons.
iRstAction[3:0]	raw_slvr_reset	_	Result of RSTACT CCC
oRstRstAction	iraw_rst_slavr_reset	_	Clear RSACT state
oResetBlock	iraw_slvr_irq	_	Triggers IRQ in IP
oRstAll	_	System reset	Should perform similar to pin RSTn
oRstCustom	_	Custom Reset if enabled	Must be setup via RSTACT_CONFIG[2]
iDeepestSleep	_	Feed from system or 0	Only used if deepest sleep supported
oWake	_	Wake controller	_

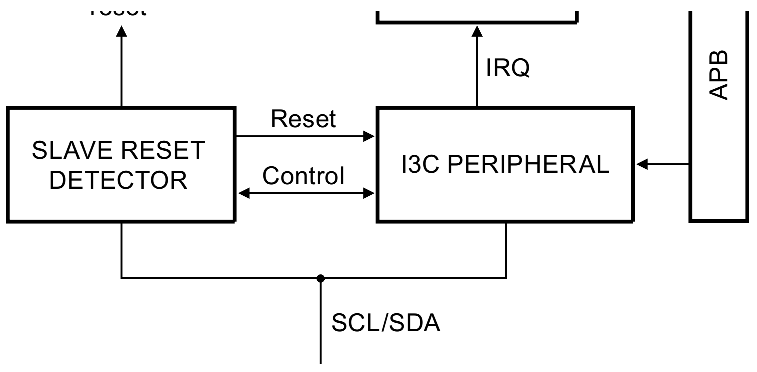


Fig 36. Target reset detector connected in the system (Target Reset may be in Always-on domain)

Below Figure shows the connections of the Target Reset detector, including support for deepest sleep and wake. If those are not used, they are just not connected. If the I3C peripheral may be powered off when the reset detector is powered, the isolation cells are used to protect it, although the nets are logically gated off when iDeepestSleep is 1.

Note that the block wants the SCL and SDA signals to be fed as clocks. This is done outside the block, using any method needed. 3 domains are needed to support the reset.

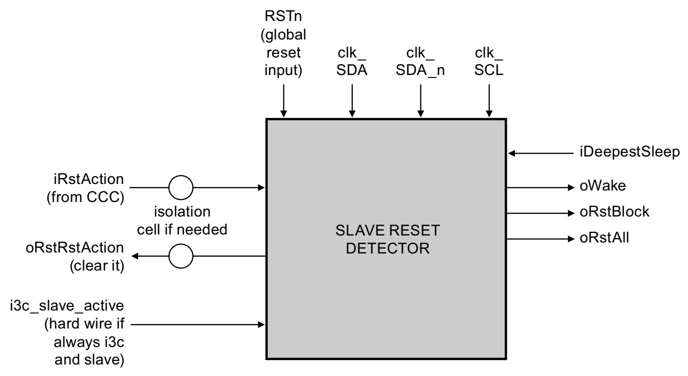


Fig 37. Showing reset detector connections — not all used in many systems

The i3c_slave_active signal must be used with caution, as it disables the reset detector when 0; it should be strapped 1 if always i3c and target, else should be safely controlled (or may defeat the purpose of the Target Reset block). If the i3c peripheral is optional, then it must be selected by some system control feature - that should be what controls i3c_slave_active.

The block is connected to the I3C peripheral through the iRstAction and oRstRsttAction ports. These are clamped off if iDeepestSleep signal is 1; if that kind of sleep is not possible, the iDeepestSleep port should be strapped 0. The 2 nets can be treated as asynchronous even though from the same SCL. This is to avoid having to do setup/hold timing between the modules when the reset detector is in a separate domain. That is, the clk_SCL and clk_SDA_n are not intended to be treated the same clocks as used in the peripheral - this allows freedom in placement of the detector.

Finally, the reset and wake signals are connected to the system; they are 0 when inactive, 1 when actively driven. There are clearing events used to clear: RstAll waits for the Reset input to assert (or i3c_active to go to 0); the RstBlock waits for the I3C peripheral to clear its cause register or to see a RSTACT or GETSTATUS. Wake clears on release of iDeepestSleep.

21.4.5 Interrupts

This section describes all the interrupts (IRQs) that the I3C module generates. All status interrupts are updated in the Controller Status (MSTATUS) and Target Status (SSTATUS) registers.

Supported interrupts occur:

- For pending IBI, CR, or Hot-Join that have been sent by a target.
- When a CCC is received and is handled by the block (SSTATUS[CHANDLED]).
- When an error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR, or CRC error.

21.4.6 External Signals

This table describes the I3C module signals:

Table 312. External signals

Signal	I/O	Description
SCL	I/O	Serial clock
SDA	I/O	Serial clock
PUR	O	Pull up resistance. There is internal pull-up resistance on SDA, which is controlled by the I3C controller. If the internal pullup is not enough, PUR can be used to control an external pull-up resistance on SDA actively.

21.4.7 Initialization

21.4.7.1 Configuration Initialization

The configuration is handled when initializing the I3C module. Configuration is done using the Controller Configuration (MCONFIG) register, which controls the frequencies, duty cycle, optimizations for performance, and other parameters.

Parameters govern several features that are supported in the design.

High-keeper controls are also included in the below configuration list.

Table 313. External signals

Signal I/O		Description
MSTENA	Controller Enable	Determines whether the I3C peripheral starts in Controller mode (Main Controller in I3C terms) or starts in Target mode (and switches to Controller mode later).
HKEEP	High Keeper	Determines how the high-keeper (weak pullup) is implemented, depending on the device capabilities.
PPBAUD	Push-Pull Baud	Sets the push-pull frequency as a divider from the FCLK (alternative clock fed to the peripheral). This frequency sets the SCL half-clock period baseline (used at least for the high time of SCL). The formula for SCL in Push-Pull mode using PPBAUD and PPLOW is: SCL freq (in MHz) = FCLK / ((2 + 2 * PPBAUD) + PPLOW) If PPLOW is 0, then SCL high = SCL low (in ns) = (1 + PPBAUD) * 1000 / FCLK (in MHz) Examples: • If FCLK = 24 MHz, then PPBAUD = 0 yields 12 MHz (42.67 ns per half period) • If FCLK = 50 MHz, then PPBAUD = 1 yields 12.5 MHz (20 + 20 = 40 ns per half period)

Table 313. External signals

Signal	I/O	Description
PPLOW	Push-Pull Low	Changes the duty cycle for Push-Pull. It indicates how many more FCLK cycles to use for low. The formula for SCL in Push-Pull mode using PPBAUD and PPLOW is: SCL freq (in MHz) = FCLK / ((2 + 2 * PPBAUD) + PPLOW) If PPLOW is non-zero, then SCL High (in ns) = (1 + PPBAUD) * 1000 / FCLK (in MHz) SCL Low (in ns) = (1 + PPBAUD + PPLOW) * 1000 / FCLK (in MHz) For example, when FCLK is 50 MHz, PPBAUD is 1, and PPLOW is 1, then the periods are • (1 + 1) * 1000 / 50 = 20 + 20 = 40 ns high • (1 + 1 + 1) * 1000 / 50 = 20 + 20 + 20 = 60 ns low This timing is equivalent to 10 MHz SCL, but this timing maintains the 40 ns high needed so I2C devices do not see the high periods. Note: The PPLOW value does not have any impact on Open-Drain mode and I2C mode SCL rate calculations.
ODBAUD	Open Drain Baud	The number of PPBAUD periods to make up one I3C Open-Drain half-clock baseline. If ODHPP is 0, the ODBAUD half-clock time period = (ODBAUD + 1) * PPBAUD high period If ODHPP is 0, the formula for SCL in Open-Drain mode is: SCL (in MHz) = FCLK / (2 + 2 * PPBAUD) * (ODBAUD + 1) Target to get open-drain half-clock period is 200 ns For example: • If PPBAUD yields 12.5 MHz (40 ns per PPBAUD period), then ODBAUD = 4 can be used to get 200 ns. See also ODHPP for details on short High and long Low. • If PPBAUD = 1, FCLK = 50 MHz, and ODBAUD = 4, then open- drain SCL = 50 / (2 + 2 * 1) * 5 = 2.5 MHz.

Table 313. External signals

Signal	I/O	Description
ODHPP	Open-Drain High Push-Pull	Optional field that allows the I3C open-drain to be long low and short high. The High period of SCL is the PPBAUD period. This period leaves enough time for the pull-up resistor to pull the SDA high when SCL is low. It is also quick when SCL is high and there are no changes happening. ODBAUD low half-clock time period = (ODBAUD + 1) * PPBAUD high period. ODBAUD high half-clock time period = PPBAUD high period If ODHPP is 1, the formula for SCL in Open-Drain mode is: SCL (in MHz) = FCLK / (1 + PPBAUD) * (ODBAUD + 1) + (1 + PBAUD) For example: • If PPBAUD produces 12.5 MHz (40 ns per PPBAUD period), then ODBAUD = 4 and ODHPP = 1. These settings provide a high period of 40 ns and a low period of 200 ns. • If PPBAUD = 1, FCLK = 50 MHz, and ODBAUD = 4 then open- drain SCL = 50 / (1 + 1) * 5 + (1 + 1) = 4.16 MHz.
I2CBAUD	I2C Baud	Indicates how many ODBAUD periods are required to communicate with I2C devices (MCTRL[TYPE] = 2 or MWMSG_SDR_CONTROL[I2C] = 1). For example, if ODBAUD gives 200 ns, and the goal is Fm+ (Fast Mode, 1 MHz), then the sum must be 1 μs. I2CBAUD acts differently for odd and even values. For example, if I2CBAUD = 3, it gives 3 ODBAUD periods low and 2 ODBAUD periods high. • If I2CBAUD = 4, then it gives 3 ODBAUD periods for low and 3 ODBAUD periods for high. • Also, if I2CBAUD = 3, this yields 200 * 3 = 600 ns low and 200 * 2 = 400 ns high, with the sum 600 + 400 = 1000 ns = 1 μs.
SKEW	Skew	The normal SDA skew from SCL is handled by using the time for the SCL to reach its pad and come back to the design. This time is normally 2 ns to 5 ns (or sometimes more). If the SKEW is too fast, then add more delay, the SKEW allows specifying the number of FCLKs to insert.

Additional optimizations:

- Use MIBIRULES[MSB0] to obtain faster START header times. If the controller application assigns all I3C dynamic addresses to be less than 40h (it does not have MSB set), MIBIRULES[MSB0] can be set. When the controller emits 7Eh (broadcast) and a target does not drive the first bit low, the rest of the header can be at push-pull speeds. This speed is two times faster or more, depending on optimizations.
- Auto-emit 7Eh speeds up the frame when used with MIBIRULES[MSB0]. It allows the processor to sleep when the frame starts automatically (in response to a target).

21.4.7.2 Interrupt service flow

Set up interrupts for Controller Status (MSTATUS) in Controller Interrupt Set (MINTSET):

Table 314. MINTSET fields

Field	Description
SLVSTART	Target Start Interrupt. Indicates whether a target is/was requesting a START by holding SDA low.
MCTRLDONE	Controller Control Done Interrupt. Indicates whether the module has completed an MCTRL request.
COMPLETE	Completed Message Interrupt. Indicates whether a message has completed.
RXPEND	Receive Pending Interrupt. Indicates whether a message is being received from a target and bytes are in the input buffer or FIFO.
TXNOTFULL	TX Buffer or FIFO Not Full Interrupt. Indicates whether the buffer, FIFOm or message register can accept another byte or halfword.
IBIWON	In-Band Interrupt (IBI) Won Interrupt. Indicates whether an IBI, CR, or HJ has won the arbitration on a header address, regardless of whether it was NACKed or ACKed.
ERRWARN	Error Or Warning Interrupt. Indicates whether an error occurred, such as improper register use, overrun or underrun of FIFO or buffer, or invalid parity or CRC in a DDR read.
NOWMASTER	Module Is Now Controller Interrupt. Indicates when the module is now a controller. That is, it was previously a target, controllership acceptance was requested from the previous controller, and controllership was accepted.

Set up interrupts for Target Status (SSTATUS) in Target Interrupt Set (SINTSET):

Table 315. SINTSET fields

Field	Description
START	Start Interrupt. A START or repeated START was seen after the START bit was last cleared.
MATCHED	Matched Interrupt. Incoming header matched the I3C Dynamic or I2C Static address of this device (if any) since the bus was last cleared.
STOP	Stop Interrupt. Stopped state detected. A STOP state was present on the bus since the bus was last cleared.
RXPEND	Receive Interrupt. Indicates when receiving a message from the controller that is not being handled by the I3C module (not a CCC message).
TXSEND	Transmit Interrupt. To-bus buffer or FIFO can accept more data to be transmitted.
DACHG	Dynamic Address Change Interrupt. Indicates occurrence of dynamic address (DA) change. Actual DA can be seen in the DYNADDR register. Also used when the MAP Auto feature is configured.
CCC	Common Command Code Interrupt. Indicates whether a CCC has been received, and is not handled by the I3C module.
ERRWARN	Error or Warning Interrupt. An error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR CRC error, or other error or warning condition.
DDRMATCHED	Double Data Rate Interrupt. Indicates when DDR matched for read or write command. Used only if HDR enabled.
CHANDLED	Common Command Code Handled Interrupt. Indicates whether a CCC is being handled by the module.
EVENT	Event Interrupt. For a target, indicates that a pending In-Band Interrupt (IBI), Controller Request (CR), or Hot-Join (HJ) has been sent as requested.
SLVRST	Target Reset. Target reset to peripheral (not whole chip). Helps application perform follow-up tasks such as reinitialization.

21.4.8 Application information

This section describes applications supported by the I3C module.

21.4.8.1 I2C configuration for meeting timing requirement for FM and FM+ modes

I2C FM and I2C FM+ mode is supported as per the I3C standard specifications.

Configuration for FM mode (Below configurations are to meet close to 400 kMhz) But In this case TSU_STA is not met.

Table 316. Configuration for FM mode, where frequency is close to 400 kHz but timing requirement is not met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL FREQ	Timing requirement* violations
24 MHz	0	4	11	370 kHz	T_{SU_STA} - 353 ns
48 MHz	1	4	11	370 kHz	T_{SU_STA} - 353 ns
134 MHz	5	4	11/12	412 kHz/382 kHz	T_{SU_STA} - 267 ns and T_{SU_STO} - 529 ns
160 MHz	6	4	10	380 kHz	T_{SU_STA} - 312 ns

Table 317. Configuration for FM mode, where all timing requirements are met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL FREQ	Timing requirement* met?
24 MHz	1	2	9	363 kHz	Yes
48 MHz	3	2	9	363 kHz	Yes
134 MHz	11	2	9/8	338 kHz / 372 kHz	Yes
160 MHz	13	2	9/8	345 kHz / 422 kHz	Yes

Configuration for FM+ mode, where the frequency is close to 1 MHz. Although, in this case the TSU_STA timing requirement is not met.

Table 318. Configuration for FM+ mode, where frequency is close to 1 MHz but timing requirement is not met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL FREQ	Timing requirement* violations
24 MHz	0	4	3	960 kHz	T_{SU_STA} - 228 ns
48 MHz	1	4	3	960 kHz	T_{SU_STA} - 228 ns
134 MHz	4	4	3	1.087 MHz	T_{SU_STA} - 156 ns
160 MHz	6	4	3	912 kHz	T_{SU_STA} - 181 ns

To meet all the timing specifications, frequency may differ from 1 MHz. Use the following configurations.

Table 319. Configuration for FM+ mode, where all timing requirements are met

FCLK	PPBAUD	ODBAUD	I2CBAUD	SCL FREQ	Timing requirement* met?
24 MHz	0	3	6	749 kHz	Yes
48 MHz	1	3	6	749 kHz	Yes
134 MHz	4	3	6	839 kHz	Yes
160 MHz	6	3	6	713 kHz	Yes

Note: The I3C controller module provides an option to slow down the clock for standard speed mode to support legacy I2C standard targets as well. Although, this option may not necessarily meet the exact timing requirement of the standard mode per the standard I2C timing specification.

*For the timing requirements under consideration, see the "I3C Timing Requirements When Communicating With I2C Legacy Devices" table in the MIPI I3C v1.1.1 specification available on www.mipi.org.

21.5 Basic configuration

Initial configuration of the I3C can be accomplished as follows:

- Enable the clock to the I3C in the SYSAHBCLKCTRL register Section 8.6.23 “System clock control 0 register”. This enables the register interface and the peripheral function clock.
- Select a clock source for the I3C function clock using the I3CFCLKSEL register (Section 8.6.26 “Peripheral clock source select registers”.
- Select a clock source for the I3C slow time control clock using the I3CSLOWTCCLKSEL register (Section 8.6.26 “Peripheral clock source select registers”).
- Select a clock divide for the I3C function clock using the I3CCLKDIV register (Section 8.6.21 “I3C clock divider register”).
- Select a clock divide for the I3C control clock using the CLKCTL1_I3C0FCLKSTCDIV register (Section 8.6.21 “I3C clock divider register”).
- Clear the I3C peripheral reset in the PRESETCTRL0 register (Section 8.6.24 “Peripheral reset control 0 register”).
- The I3C provides interrupts to the NVIC, see Table 46 “Connection of interrupt sources to the NVIC”. To allow interrupts to wake-up the device from deep-sleep mode, enable this in the STARTERP1 register (Table 115 “Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description”).
- Use the IOCON registers to connect the I3C to external pins. See Chapter 12 “I/O Configuration (IOCON)”.
- The I3C provides DMA requests to the DMA controller. See Section 17.3.3 “DMA requests”.

I3C-specific configuration

The configuration is handled once (normally) when initializing the I3C module. Configuration is done using the MCONFIG register, which controls the frequencies, duty cycle, optimizations for performance, and other parameters.

Table 320. Configuration parameters used to initialize the I3C module

Name	Function	Description
MSTENA	Controller Enable	Determines whether the I3C peripheral starts in Master mode (ie. the “Main Master” in I3C terms) or starts in slave mode (and will switch to Master later).
HKEEP	High Keeper	Determines how the high-keeper (weak pullup) is to be implemented, depending on the device capabilities.
PPBAUD	Push-Pull BAUD	Sets the push-pull frequency as a divider from the FCLK (alternative clock fed to the peripheral). This sets the half-clock period baseline (used at least for the high time of SCL). • If FCLK = 24 MHz, then a PPBAUD = 0 yields 12 MHz (42.67 ns per half period) • If FCLK = 50 MHz, then a PPBAUD = 1 yields 12.5 MHz (20+20= 40 ns per half period)
PPLOW	Push-Pull Low	Changes the duty cycle for push-pull. It indicates how many more FCLKs to use for low. For example, with a 50 MHz FCLK, a PPBAUD of 1, and a PPLOW of 1, you would get 20+20= 40 ns high and 20+20+20= 60 ns low. This would be equivalent of 10 MHz SCL timing, but maintaining the 40 ns high needed, so that I2C devices do not see the high periods.

Table 320. Configuration parameters used to initialize the I3C module

Name	Function	Description
ODBAUD	Open Drain BAUD	The number of PPBAUD periods to make up one I3C open-drain half-clock baseline. For example, if PPBAUD yields 12 MHz (40 ns per PPBAUD period), then 5 PPBAUD can be used to get 200 ns. See also ODHPP for details about short high and long low.
ODHPP	Open Drain High Push-Pull	Optional field that allows for the I3C open drain to be long low and short high. The high period of SCL will be the PPBAUD period. This leaves enough time for the pull-up resistor to pull the SDA high when SCL is low, but is quick when SCL is high and there are no changes happening.
I2CBAUD	I2C Baud	Indicates how many ODBAUD periods are needed to communicate with I2C devices.
SKEW	Skew	The normal SDA skew from SCL is handled by using the time for the SCL to reach its pad and come back to the design. This is normally 2 ns to 5 ns (or sometimes worse). If the SKEW is too fast, then to add more delay, the SKEW allows specifying the number of FCLKs to insert.

Additional optimizations:

- To get much faster START header times, MIBIRULES.MSB0 can be used. If the controller application assigns all I3C Dynamic Addresses to be less than 0x40 (it does not have MSB set), then this bit (MSB0) can be set. This means that when the controller emits 0x7E (broadcast) and the 1st bit is not driven Low by a slave, the rest of the header can be at push-pull speeds. This is about 2x faster (or more, depending on optimizations above).
- Auto-emit 7E will speed up the frame when used in conjunction with MSB0, because it allows the processor to be sleeping when the frame starts automatically (in response to a target).

21.6 Pin description

I3C signals are assigned to external pins through via IOCON. See the IOCON description (Chapter 12) to assign functions to pins on the device package.

Table 321. I^2C-bus pin description

Function	Type	Description
I3C_SCL	I/O	Clock for I3C controller or target function.
I3C_SDA	I/O	Data for I3C controlleror target function.
I3C0_PUR	O	Pullup resistor control for I3C controller function.

Recommended IOCON settings are shown in Table 322. See Chapter 12 for details of IOCON settings.

Table 322: Suggested I3C pin settings

IOCON bit(s)	Name	Comment
3:0	FUNC	Select a function for this peripheral.
4	PUPDENA	Set to 0 (pull-down/pull-up resistor not enabled).
5	PUPDSEL	Set to 0.
6	IBENA	Set to 1 (input buffer enabled).
7	SLEWRATE	Generally, set to 0 (standard mode).
8	FULLDRIVE	Generally, set to 0 (normal output drive).
9	AMENA	Set to 0 (analog input mux, if any, disabled).
10	ODENA	Set to 0 (not open-drain).
11	IIENA	Set to 0 (input function not inverted).

21.7 General information

21.7.1 Using registers when the System clock is much faster than the Functional clock

The Controller block has 2 basic clocks:

- System clock: which controls the access into the memory-mapped registers.
- Functional clock (FCLK): which is used to generate the SCL clock rate on the I2C/I3C bus.

Because there are two clocks, requests made to MCTRL and MWMSG_xxx registers have to cross over the clock domains. This is also true of data reads and writes, but there are certain aspects of MCTRL that require consideration.

The normal use will be to write MCTRL and wait for an interrupt indicating DONE and/or COMPLETE. Once that interrupt arrives, the interrupt handler will likely write a new request (like a new message or a STOP).

- When the System clock is only 8x or less faster than the FCLK, there are no special actions needed. The servicing of the interrupt takes long enough to ensure that all clock-crossing details are settled. For example, if the FCLK is 25 MHz and the system clock is 150 MHz, there are no problems.

- When the System clock is more than 8x faster than FCLK (or if the code is using a polling spin loop on MSTATUS), then special considerations may apply:
  - The MSTATUS sticky bit for DONE and COMPLETE will be set, and optionally an interrupt will be triggered.
  - To complete the clock crossing, the internal state will go through a handshake.
  - If a new request is posted before 2 FCLKs have completed, then the new request may be lost.

    For example, if the FCLK is 25 MHz and the System clock is 400 MHz, then 2 FCLKs is 16 beats of the System clock, or really 17 beats (since the clocks are not aligned). If the interrupt handler is entered in under 16 clocks and the interrupt handler tries to write the MCTRL register immediately, then the new request will be lost.

    In reality, most processors need around 12 clocks to enter an interrupt handler, and then the handler code needs to read MSTATUS and make decisions, so this will not be an issue; but if the processor is very fast, then an extra delay would be needed.

    Likewise, if using a polling spin loop, then the timing from detecting MSTATUS DONE or COMPLETE to writing the next MCTRL would need to consider that time required (2 FCLKs).

21.7.2 Controller and Target registers

The Controller registers are integrated with the Target registers. Many Controller registers are aliases of Target registers, but with different meanings. Refer to the register overview (Table 325) for more information and links to detailed register descriptions.

- When Controller operation is enabled (MCONFIG.MSTENA = 1), the Controller registers should be used.
- When Target operation is enabled (MCONFIG.MSTENA = 0), the Target registers should be used.

Table 323. Controller register summary

Name	Offset	Description
MCONFIG	0x0	Controller Configuration.
MCTRL	0x84	Controller Main Control.
MSTATUS	0x88	Controller Status, including interrupt causes.
MIBIRULES	0x8C	Rules for In-Band Interrupt use.
MINTSET	0x90	ControllerInterrupt Set/enable.
MINTCLR	0x94	Controller Interrupt Clear/disable.
MINTMASKED	0x98	Masked interrupts for Controller.
MERRWARN	0x9C	Controller Errors and Warnings.
MDMACTRL	0xA0	Controller DMA Control.
MDATACTRL	0xAC	Controller Data Control.
MWDATAB	0xB0	Controller Write Data Byte.
MWDATABE	0xB4	Controller Write Data Byte End.
MWDATAH	0xB8	Controller Write Data Half-word.
MWDATAHE	0xBC	Controller Write Data Byte End.

Table 323. Controller register summary *…continued*

Name	Offset	Description
MRDATAB	0xC0	Controller Read Data Byte.
MRDATAH	0xC8	Controller Read Data Half-word.
MWMSG_SDR_CONTROL	0xD0	Controller Write Message Control.
MWMSG_SDR_DATA	0xD0	Controller Write Message Data.
MRMSG_SDR	0xD4	Controller Read Message in SDR mode.
MWMSG_DDR_CONTROL	0xD8	Controller Write Message Control in DDR mode.
MWMSG_DDR_DATA	0xD8	Controller Write Message Data in DDR mode.
MRMSG_DDR	0xDC	Controller Read Message in DDR mode.
MDYNADDR	0xE4	Controller Dynamic Address.

Table 324. Slave register summary

Name	Offset	Description
SCONFIG	0x4	Target Configuration.
SSTATUS	0x8	Target Status register.
SCTRL	0xC	Target Control.
SINTSET	0x10	Target Interrupt Set/enable.
SINTCLR	0x14	Target Interrupt Clear/disable.
SINTMASKED	0x18	Masked interrupts for Target .
SERRWARN	0x1C	Target Errors and Warnings.
SDMACTRL	0x20	Target DMA Control.
SDATACTRL	0x2C	Target Data Control.
SWDATAB	0x30	Target Write Data Byte.
SWDATABE	0x34	Target Write Data Byte End.
SWDATAH	0x38	Target Write Data Half-word.
SWDATAHE	0x3C	Target Write Data Half-word End.
SRDATAB	0x40	Target Read Data Byte.
SRDATAH	0x48	Target Read Data Half-word.
SCAPABILITIES	0x60	Target Capabilities.
SDYNADDR	0x64	Target Dynamic Address.
SMAXLIMITS	0x68	Target Maximum Limits.
SIDPARTNO	0x6C	Target ID Part Number.
SIDEXT	0x70	Target ID Extension.
SVENDORID	0x74	Target Vendor ID.
STCCLOCK	0x78	Target Time Control Clock.
SMSGMAPADDR	0x7C	Target Message-Mapped Address
SID	0xFFC	Target Module ID.

21.8 Register description

The reset value reflects the data stored in used bits only. It does not include reserved bits content.

Table 325. Register overview: i3c (base address 0x40060000)

Name	Access	Offset	Description	Reset value	Section
MCONFIG	RW	0x0	Controller Configuration	0x0	21.8.1
SCONFIG	RW	0x4	Target Configuration	0x10000	21.8.2
SSTATUS	W1C	0x8	Target Status	0x1400 [1]	21.8.3
SCTRL	RW	0xC	Target Control	0x0	21.8.4
SINTSET	RW	0x10	Target Interrupt Set	0x80000	21.8.5
SINTCLR	W1C	0x14	Target Interrupt Clear	-	21.8.6
SINTMASKED	RO	0x18	Target Interrupt Mask	0x0	21.8.7
SERRWARN	W1C	0x1C	Target Errors and Warnings	0x0	21.8.8
SDMACTRL	RW	0x20	Target DMA Control	0x10	21.8.9
SDATACTRL	RW	0x2C	Target Data Control	0x80000030	21.8.10
SWDATAB	WO	0x30	Target Write Data Byte	-	21.8.11
SWDATABE	WO	0x34	Target Write Data Byte End	-	21.8.12
SWDATAH	WO	0x38	Target Write Data Half-word	-	21.8.13
SWDATAHE	WO	0x3C	Target Write Data Half-word End	-	21.8.14
SRDATAB	RO	0x40	Target Read Data Byte	0x0	21.8.15
SRDATAH	RO	0x48	Target Read Data Half-word	-	21.8.16
SWDATAB1	WO	0x54	Target Write Data Byte	0x0	21.8.17
Reserved	-	0x5C	Reserved		
SCAPABILITIES	RO	0x60	Target Capabilities	0xE83FFE70	21.8.18
SDYNADDR	RW	0x64	Slave Dynamic Address.	0x0	21.8.19
SMAXLIMITS	RW	0x68	Target Maximum Limits	0x0	21.8.20
SIDPARTNO	RW	0x6C	Target ID Part Number.	0x50000000	21.8.21
SIDEXT	RW	0x70	Target ID Extension.	0x66000000	21.8.22
SVENDORID	RW	0x74	Target Vendor ID.	0x11B	21.8.23
STCCLOCK	RW	0x78	Target Time Control Clock.	0x214	21.8.24
SMSGLAST	RO	0x7C	Target Message Last Matched	0x0	21.8.25
Reserved	-	0x80	Reserved		
MCTRL	RW	0x84	Controller Control.	0x0	21.8.26
MSTATUS	W1C	0x88	Controller Status.	0x1000	21.8.27
MIBIRULES	RW	0x8C	Controller In-band Interrupt Registry and Rules.	0x0	21.8.28
MINTSET	RW	0x90	Controller Interrupt Set.	0x0	21.8.29
MINTCLR	W1C	0x94	Controller Interrupt Clear.	-	21.8.30
MINTMASKED	RO	0x98	Controller Interrupt Mask.	0x0	21.8.31
MERRWARN	W1C	0x9C	Controller Errors and Warnings.	0x0	21.8.32
MDMACTRL	RW	0xA0	Controller DMA Control.	0x10	21.8.33
MDATACTRL	RW	0xAC	Controller Data Control.	0x80000030	21.8.34
MWDATAB	WO	0xB0	Controller Write Data Byte.	-	21.8.35

Table 325. Register overview: i3c (base address 0x40060000) *…continued*

Name	Access	Offset	Description	Reset value	Section
MWDATABE	WO	0xB4	Controller Write Data Byte End.	-	21.8.36
MWDATAH	WO	0xB8	Controller Write Data Half-word.	-	21.8.37
MWDATAHE	WO	0xBC	Controller Write Data Halfword End	-	21.8.38
MRDATAB	RO	0xC0	Controller Read Data Byte.	0x0	21.8.39
MRDATAH	RO	0xC8	Controller Read Data Half-word	0x0	21.8.40
MWDATAB1	WO	0xCC	Controller Write Byte Data 1(to bus)	0x0	21.8.41
MWMSG_SDR_CONTROL	WO	0xD0	Controller Write Message Control in SDR mode	-	21.8.42
MWMSG_SDR_DATA	WO	0xD0	Controller Write Message Data in SDR mode	0x0	21.8.43
MRMSG_SDR	RO	0xD4	Controller Read Message in SDR mode.	0x0	21.8.44
MWMSG_DDR_CONTROL	WO	0xD8	Controller Write Message in DDR mode: First control word	-	21.8.45
MWMSG_DDR_DATA	WO	0xD8	Controller Write Message Data in DDR mode (subsequent writes to this address). Also see MWMSG_DDR_CONTROL register.	0x0	21.8.46
MRMSG_DDR	RO	0xDC	Controller Read Message in DDR mode.	0x0	21.8.47
MDYNADDR	RW	0xE4	Controller Dynamic Address.	0x0	21.8.48
Reserved	-	0x100	Reserved		
Reserved	-	0x10C	Reserved		
Reserved	-	0x114	Reserved		
SMAPCTRL0	RO	0x11C	Map Feature Control 0	0x0	21.8.49
Reserved	-	0x120	Reserved	-	21.8.56
Reserved	-	0x124 -0x12 C	Reserved		
Reserved	-	0x140	Reserved		
Reserved	-	0x144	Reserved		
Reserved	-	0xFF0	Reserved		
SID	R	0xFFC	Slave Module ID.	0x0	21.8.50

[1] Some bits are undefined at reset.

21.8.1 Controller Configuration

Controls all controller states when controller operation is enabled. The MCONFIG register should not be changed when an active transaction is occurring.

Table 326. Controller Configuration (MCONFIG, offset = 0x0)

Bit	Symbol	Value	Description	Reset value
1:0	MSTENA		Indicates if the controller is enabled and what states the controller can use.	0x0
		0	CONTROLLER_OFF. Controller is disabled. The I3C module can only use Target mode.	
		1	CONTROLLER_ON. Controller is enabled. When used upon start-up, this I3C module is the Main Controller by default. The module controls the bus unless the controller is handed off. If the controller is handed off, then MSTENA must become 2, so that it has the capability to become the controller again. Performing the handoff means emitting GETACCMST CCC command. If the command is accepted, the module emits a STOP and sets the MSTENA field to 2 (or 0).	
		2	CONTROLLER_CAPABLE. The I3C module is controller-capable, but the module is operating as a target now. When used from the start, the I3C module starts as a target, but is prepared to switch to Controller mode. To switch to Controller mode, the target emits a Controller Request (CR), or receives a GETACCMST CCC command and accepts it (to switch on the STOP).	
		3	Reserved.	
2	-	-	Reserved.	-
3	DISTO		Disable Timeout Disables the timeout that produces application errors. If the controller has been left in a state other than Stopped for more than 100 s (because 10 kHz is the slowest allowed I3C speed), the timeout sends a MERRWARN interrupt. To prevent the MERRWARN interrupt during development or testing, write 1 to DISTO to disable the timeout. In systems that support timeouts, timeout is disabled automatically during debug.	0x0
		0	Timeout enabled	
		1	Timeout disabled, if timeout is configured.	
5:4	HKEEP		High-Keeper. Indicates how High-Keeper is to be supported. NOTE: Your specific device may not support any or all of these High-Keeper methods. Use HKEEP = 2 when I3C. Use HKEEP = 0/1 when I2C.	0x0
		0	NONE. Use PUR (Pull-Up Resistor). No separate pin_HK_SDA/pin_HK_SCL is used. Only pin_PUR_oena is used for sda. Hold pin_SCL_oena High in this mode, SCL clock is push-pull and pin_SCL_out toggles which is actual clock.	
		1	WIRED_IN. High Keeper controls, use pin_HK_SDA/pin_HK_SCL (High Keeper) controls. SCL may use an HK or that signal (pin_HK_SCL) may only OR into pin_SCL_oena. Uses pin_HK_SDA as well for sda high keeper, along with pin_PUR_oena.	
		2	PASSIVE_SDA. Passive on SDA, can Hi-Z (high impedance) for Bus Free (IDLE) and hold. The pins, pin_HK_SDA and pin_HK_SCL are not used, Only a combination of SDA_oena and pin_PUR_oena are used.	
		3	PASSIVE_ON_SDA_SCL. Passive on SDA and SCL, can Hi-Z (high impedance) both for Bus Free (IDLE), and can Hi-Z SDA for hold. This is for I2C clock stretching mode, where both SCL and SDA are open drain.	
6	ODSTOP	-	Open drain stop.	0x0

Table 326. Controller Configuration (MCONFIG, offset = 0x0) *…continued*

Bit	Symbol	Value	Description	Reset value
		0	Disable open-drain stop. ODSTOP must be disabled when sending an HDR exit pattern.	
		1	Enable open-drain stop. STOP is emitted at open-drain speeds even for I3C messages. In legacy devices, this feature can ensure that the legacy devices see the STOP.	
7	-	-	Reserved.	-
11:8	PPBAUD	-	Push-pull baud rate. The number of FCLK counts makes each push-pull low and normally high period. PPBAUD = 0 when run at 1/2 input FCLK speed. For example, a 24 MHz FCLK produces a 12 MHz SCL, because each FCLK is SCL Low or SCL High. Note: The effect Push-Pull Low (PPLOW) has upon the duty cycle. For example, 24 MHz with 50/50 duty cycle is 12 MHz. However, when PPLOW adds 3 more low beats, the push-pull baud rate becomes 4.8 MHz (from 24 MHz or 5 beats).	0x0
15:12	PPLOW	-	Push-Pull low. Adder for push-pull low, to create a duty-cycle with a longer low period, with up to 15 more FCLKs low than high. PPLOW=0 means a 50:50 duty cycle.	0x0
23:16	ODBAUD	-	Open drain baud rate. ODBAUD= (number of PPBAUD counts) - 1. ODBAUD should not be 0 (not the same as push-pull). This would normally go for 200 ns (see I2CBAUD for I2C counts). When used with ODHPP, this gives 250 ns per clock in I3C. Therefore, if PPBAUD gives 12 MHz, then 1 PPBAUD is 1/2 of 12 MHz or 41.67 ns, and then to get around 200 ns, use 5-1=4 for ODBAUD.	0x0

Table 326. Controller Configuration (MCONFIG, offset = 0x0) *…continued*

Bit	Symbol	Value	Description	Reset value
24	ODHPP	-	Open Drain High Push-Pull See Sending a CCC to I3C targets for open-drain timing check upon the first 7Eh broadcast allowing I3C peripherals acting as I2C legacy devices to turn off their I2C 50-ns spike filters. 0b - ODHPP disabled. Open-Drain SCL High half-clock period is the same as the Open-Drain Low SCL half-period. 1b - ODHPP enabled. Open-Drain High SCL half-lock period is one PPBAUD count for I3C messages. This setting is faster (and works for I3C devices). Any legacy I2C devices on the bus will not see the SCL High at all (less than the spike filter period).	0x0
27:25	SKEW	-	Skew. Number of FCLK counts for an SDA change after SCL for I3C push-pull; this is in addition to the round trip of the SCL line from the pad back to the module (6 ns or more). • SKEW is normally not needed, so assign SKEW=0. • SKEW is only used if SDA is not naturally skewing from an SCL change. • I2C automatically skews SDA (but not PUR) to match I2C rules.	0x0
31:28	I2CBAUD	-	I2C baud rate. The I2C low and high in terms of number of ODBAUD counts. • I2CBAUD >> 1 is the main count load, and it is count - 1 . So, for I2CBAUD >> 1: I2CBAUD = 0 for one ODBAUD beat, and I2CBAUD = 1 for two ODBAUD beats. • If I2CBAUD[28] is 1, then the low time has one extra ODBAUD beat. The I2CBAUD field is normally 3, where ODBAUD gives 200 ns. For I2CBAUD >> 1, I2CBAUD = 1, which means two ODBAUD beats. To meet the requirements for Fast Mode Plus (Fm+), (2 + 1) * 200 = 600 ns low, with 2 * 200 = 400 ns high for 1 s period. For Fast Mode (FM), I2CBAUD is normally 11 (giving 2.6 s) or 6 (giving 2.4 s).	0x0

21.8.2 Target Configuration

Contains fields that must be configured before the module is activated.

Table 327. Target Configuration register (SCONFIG, offset = 0x4)

Bit	Symbol	Description	Reset value
0	SLVENA	Target enable. SLVENA must not be set before registers like SCONFIG (SIDPARTNO, SIDEXT, and others) are set, because these registers affect the data to and from the controller. Target enable is configured just once before the I3C bus comes up. If target enable is used at other times, see Hot-Join. In the case of Hot-Join, the Hot-Join bit (SCAPABILITIES[IBI_MR_HJ]) must be 1 before writing 1 to the target enable bit (SLVENA), so that the device does not see a START or STOP incorrectly. 0b - Target ignores the I2C or I3C bus 1b - Target can operate on the I2C or I3C bus	0x0
1	NACK	Not Acknowledge 0b - Always NACK disable 1b - Always NACK enable. The target rejects all requests to it, except for a Common Command Code (CCC) broadcast. NACK = 1 should be used with caution, because the controller may decide that the target is missing, if NACK is overused.	0x0
2	MATCHSS	Match START or STOP 0b - Match START or STOP disable 1b - Match START or STOP enable. START and STOP sticky SSTATUS bits only become 1 when SSTATUS[MATCHED] is 1. This setting allows START and STOP to be used to detect the end of a message to/from this target.	0x0
3	S0IGNORE	Ignore TE0/TE1 Errors 0b - Do not ignore TE0/TE1 errors 1b - Ignore TE0/TE1 errors. Target does not detect TE0 or TE1 errors, so it does not lock up waiting on an Exit Pattern. This setting should only be used when the bus does not use HDR mode.	0x0
4	HDROK	HDR OK Deprecated. Use the SIDEXT[BCR] field instead. 0b - Disable HDR OK. 1b - Enable HDR OK. Allow HDR-DDR and/or HDR-BT messaging if available by setting the corresponding SIDEXT[BCR] bit to say HDR is available, and the corresponding GETCAPS bit for DDR and/or BT bit permitting use.	0x0
8:5	-	Reserved.	-
9	OFFLINE	Offline If OFFLINE = 1 when the target enable (SCONFIG[SLVENA]) is set to 1, then the I3C module waits for either 60 s of bus quiet or an HDR Exit Pattern. This waiting ensures that the bus is not in HDR mode, and so can safely monitor for the next activity in Single Data Rate (SDR) mode. 0b - Disable 1b - Enables wait to ensure the bus is not in HDR mode.	0x0
15:10	-	Reserved.	-

Table 327. Target Configuration register (SCONFIG, offset = 0x4) *…continued*

Bit	Symbol	Description	Reset value
21:16	BAMATCH	Bus Available Match The Bus Available condition match value for the current slow clock. BAMATCH provides the count of the slow clock to count out 1 s (or more) to allow an In-Band Interrupt (IBI) to drive SDA low when the controller is not doing so. The maximum width and maximum values are controlled by the I3C module.	0x0
24:22	-	Reserved.	-
31:25	SADDR	Static address. Sets the I2C 7-bit static address, otherwise must be 0.	0x0

21.8.3 Target Status

Not all bits are used if the module only acts as a target. The Target Status register indicates sticky status for interrupts and "states" and "modes" related to the I3C bus. The fields are divided into current activity, interrupt maskable actions, then states and modes on the bus.

Table 328. Target Status register (SSTATUS, offset = 0x8)

Bit	Symbol	Value	Description	Reset value
0	STNOTSTOP	-	Status Not Stop Other SSTATUS bits may also be set when busy. STNOTSTOP can also be 1 after an TE0 or TE1 error, when the I3C module is waiting for an Exit Pattern. 0b - I3C module is in a STOP condition. 1b - The bus is busy (has activity).	0x0
1	STMSG	-	Status message If STNOSTOP = 1, STMSG is 0 when a non-matching address is seen, until the next repeated START or STOP occurs. 0b - Bus target not listening or responding. 1b - This bus target is listening to the bus traffic or responding.	0x0
2	STCCCH	-	Status Common Command Code Handler 0b - No CCC message is being handled. 1b - A CCC message is being handled automatically.	0x0
3	STREQRD	-	Status Request Read See also Status High Data Rate (STHDR) for Double Data Rate (DDR) handling. 0b - REQ in process is not an SDR read from this target. 1b - The REQ in process is an SDR read from this target, or an In-Band Interrupt (IBI) is being pushed out.	0x0
4	STREQWR	-	Status Request Write See Status High Data Rate (STHDR) for Double Data Rate (DDR) handling. 0b - REQ in process is not SDR write data from the controller. 1b - REQ in process is SDR write data from the controller to this bus target (or all I3C targets), but not in ENTDAA mode.	0x0

Table 328. Target Status register (SSTATUS, offset = 0x8) *…continued*

Bit	Symbol	Value	Description	Reset value
5	STDAA	-	Status Dynamic Address Assignment 0b - Not in ENTDAA mode. 1b - I3C bus is in Enter Dynamic Address Assignment (ENTDAA) mode, regardless of whether this bus target has a Dynamic Address or not.	0x0
6	STHDR	-	Status High Data Rate 0b - I3C bus not in HDR-DDR mode 1b - The I3C bus is in HDR-DDR mode , regardless of whether HDR mode is supported by this module or not, and regardless of whether the message is to this module or to some other module.	0x0
7	-	-	Reserved.	-
8	START	-	Start This field is not usually needed, but can be used for wake events. 0b - No START seen. 1b - A START or repeated START was seen after the START bit was last cleared.	0x0
9	MATCHED	-	Matched 0b - No header matched. 1b - An incoming header matched the I3C Dynamic or I2C Static address of this device (if any) since the bus was last cleared.	0x0
10	STOP	-	Stop Detects stopped state. The STNOTSTOP state also indicates when the I3C module is in stop mode. A fast STOP/START combination may not trigger the STOP status. In that case, START is always set. 0b - No STOP detected. 1b - Stopped state detected. A STOP state was present on the bus since the bus was last cleared.	0x0
11	RX_PEND	-	Received Message Pending Indicates when receiving a message from the controller that is not being handled by the I3C module (not a CCC message). Such messages are internally processed by the module. For all but External FIFO, this process uses SDATACTRL[RXTRIG] , which defaults to not-empty. If DMA is enabled for receiving, then DMA is signaled as well. RX_PEND automatically becomes 0 if data is read (from FIFO and non-FIFO sources). 0b - No received message is pending. 1b - Received message is pending.	0x0
12	TXNOTFULL	-	Transmit Buffer Is Not Full Indicates whether transmit buffer is not full. If DMA is enabled for transmitting, then it will also be signaled to provide more data. 0b - Transmit buffer full 1b - Transmit buffer not full. To-bus buffer or FIFO can accept more data to be transmitted. For all but External FIFO, this process uses SDATACTRL[TXTRIG] , which defaults to not-full.	0x1

Table 328. Target Status register (SSTATUS, offset = 0x8) *…continued*

Bit	Symbol	Value	Description	Reset value
13	DACHG	-	Dynamic Address Change Indicates occurrence of dynamic address (DA) change. Actual DA can be seen in the DYNADDR register. This field is also used when the MAP Auto feature is configured, changing one or more MAP items. See DYNADDR and MAPCTRLn. DYNAADDR for the main DA (0) indicates whether last change was due to Auto-MAP. 0b - No DA change detected. 1b - DA change detected. The target DA has been assigned, re-assigned, or reset (lost) and is now in the state of being valid or none.	0x0
14	CCC	-	Common Command Code Indicates whether a CCC has been received, and is not handled by the I3C module. There are 2 types of Common Command Codes: •Broadcast CCC. Corresponds with RXPEND, and the first byte is the CCC (command). •Direct CCC, which may never be directed to this device. If Direct CCC are directed to this device, then the TXSEND or RXPEND are triggered, and the RXPEND contains the command. 0b - No CCC received. 1b - CCC received.	0x0
15	ERRWARN	-	Error Warning An error or warning has occurred, such as data underrun, data overrun, parity error, HDR-DDR CRC error, or other error or warning condition. See the Target Errors and Warnings (SERRWARN).	0x0
16	HDRMATCH	-	High Data Rate Command Match Indicates whether HDR command matched the I3C Dynamic Address of this device. The HDR command is available as the first byte and RXPEND are set. The MSB of the command byte indicates whether it is a read or a write command. If the HDR command is a read, and there are to-bus bytes waiting, then the command is ACKed and the data is sent back. Otherwise, the HDR command is NACKED. NOTE: When HDRMATCH is 1, ERRWARN should be checked, because the HPAR error may be encountered after signaling this HDR command. The parity is after the destination address and command. 0b - HDR command did not match. 1b - HDR command matched the I3C Dynamic Address of this device.	0x0
17	CHANDLED	-	Common Command Code Handled Indicates whether a CCC is being handled by the module. This field is a notification only, but it may result in updates to the SSTATUS register. 0b - CCC handling not in progress. 1b - CCC handling in progress.	0x0
18	EVENT	-	Event For a target, indicates that a pending In-Band Interrupt (IBI), Controller Request (CR), or Hot-Join (HJ) has been sent as requested. See the upper status register fields for details. EVENT only asserts when ACKed by a controller. 0b - No event has occurred. 1b - An IBI, CR, or HJ has occurred	0x0

Table 328. Target Status register (SSTATUS, offset = 0x8) *…continued*

Bit	Symbol	Value	Description	Reset value
19	SLVRST	-	Target Reset Target reset to peripheral (not whole chip). Helps application perform follow-up tasks such as re- initialization.	-
21:20	EVDET		Event details. Current details of the last (EVENT=1) or pending event.	undefined
		0	NONE: no event or no pending event	
		1	NO_REQUEST: Request not sent yet. Either there was no START yet, or is waiting for Bus-Available or Bus-Idle (HJ).	
		2	NACKED: Not acknowledged (Request sent and NACKed); the module will try again.	
		3	ACKED: Acknowledged (Request sent and ACKed), so Done (unless the time control data is still being sent).	
23:22	-	-	Reserved.	-
24	IBIDIS	-	In-Band Interrupts Are Disabled While In-Band Interrupts are disabled, CTRL requests are not responded to. 0b - In-Band Interrupts not disabled 1b - In-Band Interrupts disabled	0x0
25	MRDIS	-	Controller Requests Are Disabled While Controller Requests are disabled, CTRL requests are not responded to. 0b - Controller Requests not disabled 1b - Controller Requests disabled	0x0
26	-	-	Reserved.	-
27	HJDIS	-	While Hot-Join is disabled, CTRL requests are not responded to. 0b - Hot-Join not disabled 1b - Hot-Join disabled	0x0
29:28	ACTSTATE		Activity state from Common Command Codes (CCC).	0x0
		0	NO_LATENCY: normal bus operations	
		1	LATENCY_1MS: 1 ms of latency	
		2	LATENCY_100MS: 100 ms of latency	
		3	LATENCY_10S: 10 seconds of latency	
31:30	TIMECTRL		Time control. Indicates if time control is currently enabled.	0x0
		0	NO_TIME_CONTROL: No time control is enabled	
		1	SYNC_MODE. Synchronous mode is enabled	
		2	ASYNC_MODE: Asynchronous standard mode (0 or 1) is enabled	
		3	BOTHSYNCASYNC. Both Synchronous and Asynchronous modes are enabled	

21.8.4 Target Control

Contains controls for the active use of the I3C bus, like event generation (for example, interrupts to the master). The Slave Control register is used to activate various special operations for the Slave, but only if the module is configured to support those operations. This includes events such as IBI and GETSTATUS fields (except the Protocol error, which is automatically set).

Table 329. Target Control register (SCTRL, offset = 0xC)

Bit	Symbol	Value	Description	Reset value
1:0	EVENT		EVENT. • If EVENT is set to non-0, it will request an event. • After being requested, SSTATUS.EVENT and SSTATUS.EVDET will show the status as it progresses. • After completion, the EVENT field will automatically return to 0. • After EVENT is non-0, only 0 can be written to EVENT (to cancel), until the event processing is done (finished).	0x0
		0	NORMAL_MODE: If EVENT is set to 0 after was a non-0 value, event processing will cancel if the event processing has not yet started; if event processing has already been started, then event processing will not be be cancelled.	
		1	IBI: Start an In-Band Interrupt. This will try to push an IBI interrupt onto the I3C bus. If data is associated with the IBI, then the data will be read from the SCTRL.IBIDATA field. If time control is enabled, then this data will also include any time control-related bytes; additionally, the IBIDATA byte will have bit 7 set to 1 automatically (as is required for time control). The IBI interrupt will occur after the 1st (mandatory) IBIDATA, if any.	
		2	MASTER_REQUEST: Start a Master-Request.	
		3	HOT_JOIN_REQUEST: Start a Hot-Join request. A Hot-Join Request should only be used when the device is powered on after the I3C bus is already powered up, or when the device is connected using hot insertion methods (the device is powered up when it is physically inserted onto the powered-up I3C bus). The hot join will wait for Bus Idle, and SCTRL.EVENT=HOT_JOIN_REQUEST must be set before the slave enable (SCONFIG.SLVENA).	
2	-	-	Reserved.	-
3	-	-	Reserved	-
6:4	MAPIDX		Map Index Index of Dynamic Address of the IBI. This index is 0 for the main or base Dynamic Address. Used only if mapping is enabled. See Controller Dynamic Address (MDYNADDR) register for more details.	
7	-		Reserved	
15:8	IBIDATA	-	In-Band Interrupt Data Data byte accompanying the IBI, if the module is enabled for IBI. If SCTRL[IBIDATA] is enabled, then IBI is required.	0x0
19:16	PENDINT	-	Pending Interrupt Should be set to the pending interrupt that the GETSTATUS CCC command returns. The pending interrupt should be maintained by the application, because the controller reads this field. If PENDINT = 0 and •If an IBI interrupt is pending, the GETSTATUS command returns 1. •If an IBI interrupt is not pending, the GETSTATUS field returns 0.	0x0

Table 329. Target Control register (SCTRL, offset = 0xC) *…continued*

Bit	Symbol	Value	Description	Reset value
21:20	ACTSTATE	-	Activity State of Target Should be set to the activity state of the target that the GETSTATUS CCC command returns as activity mode. The activity state should be maintained by the application, because the controller reads this field. If the activity state is not configured, then the GETSTATUS command always returns 0	0x0
23:22	-	-	Reserved.	-
31:24	VENDINFO	-	Vendor Information Should be set to the Vendor Reserved field that the GETSTATUS CCC returns. The vendor information should be maintained by the application, because the controller reads this field. If VENDINFO is not configured, then the GETSTATUS field always returns 0.	0x0

21.8.5 Target Interrupt enable Set (SINTSET)

Sets interrupt enables for select Target Status (SSTATUS) fields. Reading the SINTSET register returns the status of the interrupt enables..

- To activate an interrupt enable, write 1 to its corresponding field in this register (SINTSET).
- To disable an interrupt, write 1 to its corresponding field in the Target Interrupt Clear (SINTCLR) register. Writing 0 to the interrupt enable in this register (SINTSET) does not disable the interrupt.

The Interrupt registers allow the masking of interrupt sources. They also allow the checking of which interrupts have activated. The normal method is to enable an interrupt, and then once the interrupt occurs, clear the interrupt either by writing the SSTATUS register or by performing action on the corresponding data register. The interrupt is level-held, meaning the interrupt stays set until the cause is cleared by some method. The module prevents races; if a new event occurs, that new event is not lost.

- SINTSET sets interrupt enables for Target Status (SSTATUS) fields. Reading the SINTSET register returns the status of the interrupt enables.
- SINTCLR clears interrupt enables for SSTATUS fields.
- SINTMASKED returns the value of the SSTATUS fields ANDed with their interrupt enables.

Table 330. Target Interrupt enable Set (SINTSET, offset = 0x10)

Bit	Symbol	Description	Reset value
7:0	-	Reserved.	-
8	START	Start Interrupt Enable Interrupt on START and repeated START when needed (such as wakeup). See also SINTSET[STOP] . 0b - Disable START interrupt 1b - Enable START interrupt	0x0
9	MATCHED	Match interrupt enable Interrupt on Matching header for I3C Dynamic Address. If configured and if no Dynamic Address set, this interrupt is also for matching a header on a I2C Static Address. See Controller Dynamic Address (MDYNADDR) register. 0b - Disable match interrupt 1b - Enable match interrupt	0x0
10	STOP	Stop Interrupt Enable Interrupt on STOP state on the bus. See SINTSET[START] as the preferred interrupt when needed. This interrupt may not trigger for quick a STOP/START combination, because it relates to the state of being stopped. 0b - Disable STOP interrupt 1b - Enable STOP interrupt	0x0
11	RXPEND	Receive Interrupt Enable Interrupt when receiving a message from controller that is not being handled by the block like where data is consumed by hardware directly when it goes into the FIFO (excludes CCCs being handled automatically). If this interrupt is for a FIFO, it is a receive fullness trigger. If this interrupt is for DMA, then it indicates message end. 0b - Disable Receive interrupt 1b - Enable Receive interrupt	0x0
12	TXSEND	Transmit Interrupt Enable Interrupt when request data by controller (read). This interrupt occurs on the first request (header) as well as when ready for more. The application indicates whether the interrupt is for more data or END. If this interrupt is for a FIFO, it triggers on the Transmit emptiness trigger. If this interrupt is for DMA, then it indicates message end (DMA end or termination). 0b - Disable Transmit interrupt 1b - Enable Transmit interrupt	0x0
13	DACHG	Dynamic Address Change Interrupt Enable Interrupt on Dynamic address defined (SETDASA or ENTDAA) or lost (RSTDAA). See also Controller Dynamic Address (MDYNADDR) register. 0b - Disable DA Change interrupt 1b - Enable DA Change interrupt	0x0
14	CCC	CCC (that was not handled by I3C module) Interrupt Enable For CCCs not handled by the block, RXPEND also interrupts, and SSTATUS[STREQRD] indicates that it is a CCC sending a read request. 0b - Disable CCC interrupt 1b - Enable CCC interrupt	0x0

Table 330. Target Interrupt enable Set (SINTSET, offset = 0x10) *…continued*

Bit	Symbol	Description	Reset value
15	ERRWARN	Error or Warning Interrupt Enable Interrupt when an error or warning has occurred, such as data underrun, data overrun, parity error, or HDR-DDR CRC error. See the Target Errors and Warnings (SERRWARN) register for details of the cause. Only available for errors that are configured features. 0b - Disable error or warning interrupt 1b - Enable error or warning interrupt	0x0
16	DDRMATCHED	Double Data Rate Interrupt Enable Interrupt when DDR matched for read or write command. Used only if HDR enabled. 0b - Disable DDR interrupt 1b - Enable DDR interrupt	0x0
17	CHANDLED	Common Command Code (CCC) Interrupt Enable Interrupt when a CCC is received and handled by the block (is done). SSTATUS shows new results. CHANDLED can be used to track when Activity states and when masks on events (for example, IBIs) occur. Used for CCCs that are enabled. 0b - Disable CCC Handled interrupt 1b - Enable CCC Handled interrupt	0x0
18	EVENT	Event Interrupt Enable Interrupt when Pending IBI, CR, or Hot-Join has been sent as requested. See SSTATUS[EVDET] . Used only if configured to support events. 0b - Disable Event interrupt 1b - Enable Event interrupt	0x0
19	SLVRST	Target Reset Indicates when Target Reset pattern detected and action is set to reset peripheral. Application should reset peripheral or verify it is correct. This interrupt should not be unmasked. Used only if SLVRST is configured. 0b - Disable Target Reset interrupt 1b - Enable Target Reset interrupt	-
31:20		Reserved	

21.8.6 Target Interrupt Clear (SINTCLR)

Clears interrupt enables for select Target Status (SSTATUS) fields. To clear an interrupt enable, write 1 to the corresponding field in this register (SINTCLR). Writing 0 has no effect.

Table 331. Target Interrupt enable Clear register (SINTCLR, offset = 0x14)

Bit	Symbol	Description	Reset value
7:0	-	Reserved.	-
8	START	START Interrupt Enable Clear	-
9	MATCHED	MATCHED Interrupt Enable Clear	-
10	STOP	STOP Interrupt Enable Clear	-
11	RXPEND	RXPEND Interrupt Enable Clear	-
12	TXSEND	TXSEND Interrupt Enable Clear.	-
13	DACHG	DACHG Interrupt Enable Clear	-

Table 331. Target Interrupt enable Clear register (SINTCLR, offset = 0x14) *…continued*

Bit	Symbol	Description	Reset value
14	CCC	CCC Interrupt Enable Clear	-
15	ERRWARN	ERRWARN Interrupt Enable Clear.	-
16	DDRMATCHED	DDRMATCHED Interrupt Enable Clear	-
17	CHANDLED	CHANDLED Interrupt Enable Clear.	-
18	EVENT	EVENT Interrupt Enable Clear	-
19	SLVRST	Target Reset SLVRST interrupt enable clear	-
31:20	-	Reserved.	

21.8.7 Target Interrupt Mask

Returns the status of enabled interrupts (the value of Target Status (SSTATUS) ANDed with the value of Target Interrupt Set (SINTSET)).

Table 332. Target Interrupt Mask register (SINTMASKED, offset = 0x18)

Bit	Symbol	Description t	Reset value
7:0	-	Reserved.	-
8	START	START interrupt mask.	0x0
9	MATCHED	MATCHED interrupt mask.	0x0
10	STOP	STOP interrupt mask.	0x0
11	RXPEND	RXPEND interrupt mask.	0x0
12	TXSEND	TXSEND interrupt mask.	0x1
13	DACHG	DACHG interrupt mask.	0x0
14	CCC	CCC interrupt mask.	0x0
15	ERRWARN	ERRWARN interrupt mask.	0x0
16	DDRMATCHED	DDRMATCHED interrupt mask	0x0
17	CHANDLED	CHANDLED interrupt mask.	0x0
18	EVENT	EVENT interrupt mask.	0x0
31:19	-	Reserved.	-

21.8.8 Target Errors and Warnings (SERRWARN)

Contains errors and warnings from I3C and I2C protocols, which includes internal issues such as overrun and underrun, detected errors and conditions like parity errors, CRC errors, and read terminations by the controller. Related to SSTATUS[ERRWARN] and SINTSET[ERRWARN] interrupt.

Table 333. Target Errors and Warnings register (ERRWARN, offset = 0x1C)

Bit	Symbol	Description	Reset value
0	ORUN	Overrun Error Indicates when the internal from-bus buffer or FIFO is overrun (too many characters are arriving and cannot be processed by the user application fast enough). 0b - No overrun error 1b - Overrun error	0x0
1	URUN	Underrun Error Indicates when the internal to-bus buffer/FIFO is underrun during data read (the application is not providing the data fast enough). The END bit or register should be used if the read was the last one. 0b - No underrun error 1b - Underrun error	0x0
2	URUNNACK	Underrun and Not Acknowledged (NACKED) Error Indicates when the internal to-bus buffer/FIFO is underrun in the read header and so the module NACKED the header. 0b - No underrun and not acknowledged error 1b - Underrun and not acknowledged error	0x0
3	TERM	Terminated Error Indicates when the controller terminates a read from a target when an END is not set (on the same read or the previous read). 0b - No terminated error 1b - Terminated error	0x0
4	INVSTART	Invalid Start Error Indicates an invalid condition with SCL falling before SDA falls, so there is no start. 0b - No invalid start error 1b - Invalid start error	0x0
7:5	-	Reserved.	-
8	SPAR	SDR Parity Error Indicates when an SDR Parity error on a message from the controller occurs. This error also sets the GETSTATUS Protocol Error sticky bit (which becomes 0 after a GETSTATUS read). For read operations, this field becomes 1 when a Read Abort (timeout) occurs due to the controller not driving clock for more than 100 s during an I3C SDR Read. 0b - No SDR Parity error 1b - SDR Parity error	0x0
9	HPAR	HDR Parity Error Indicates when HDR Parity error or framing error on a message from the controller occurs. The corresponding command or data that has the error is usually in the Rx buffer, which can be read using the Target Read Data Byte (SRDATAB) register. 0b - No HDR Parity error 1b - HDR Parity error	0x0
10	HCRC	HDR-DDR CRC Error Indicates when an HDR-DDR Cyclic Redundancy Check (CRC) error on a message from the controller occurs. Causes include an HDR Restart and an Exit being issued before an HDR-DDR message from controller has finished. This error calls into question the data from the entire DDR command frame. 0b - No HDR-DDR CRC error 1b - HDR-DDR CRC error	0x0

Table 333. Target Errors and Warnings register (ERRWARN, offset = 0x1C) *…continued*

Bit	Symbol	Description	Reset value
11	S0S1	TE0 or TE1 Error Indicates when an TE0 or TE1 error has occurred and the target is locked and waiting for an HDR Exit Pattern. Writing 1 to S0S1 causes the module to release the lock, but this method should be used with great care. S0S1 becomes 0 automatically when an Exit Pattern is detected, so writing 1 to S0S1 must be used under controlled circumstances to avoid problems. Before starting to operate normally after this error, the module waits for a START (or repeated START) or STOP. 0b - No TE0 or TE1 error 1b - TE0 or TE1 error	0x0
15:12	-	Reserved.	-
16	OREAD	Over-read Error Indicates that the Target Read Data Byte (SRDATAB) register was read for more bytes than were available by the application. This error also indicates over-read errors for the Target Read Data Halfword (SRDATAH) register, if it is enabled. 0b - No Over-read error 1b - Over-read error	0x0
17	OWRITE	Over-write Error Indicates that the Target Write Data Byte (SWDATAB) or Target Write Data Byte End (SWDATABE) register was written when full. 0b - No Overwrite error 1b - Overwrite error	0x0
31:18	-	Reserved.	-

21.8.9 Target DMA Control (SDMACTRL)

Allows DMA to be used for inbound and outbound messages. This register is limited in value for target use because the target must be reactive. Two common use models are:

- To avoid an overrun in from-bus collection. SCONFIG[MATCHSS] becomes 1, then the processor enables the interrupts for START and STOP and enables the DMA to collect the data. The START or STOP interrupt only occurs after a message is directed to the target (MATCHED is 1). The DMA copied data can then be examined.

  NOTE: Do not enable DMA after a transaction starts. To avoid a RX FIFO overrun, DMA must be enabled only while enabling the target and interrupts.
- To perform larger reads from the to-bus. I3C and I2C reads are preceded by a write that indicates what will be read (or in response to an IBI from the target). Because of this process, the DMA can be used to push through the data.
- For I3C, the last value needs to be handled by the processor, unless the DMA moves wider words and is able to set the END bit (i.e., 16-bit values when in byte mode or For I2C, the last value is determined by the controller, so the DMA may end early or may run out when the controller expects more values.32-bit values when in half-word mode).
- For I2C, the last value is determined by the Master, so the DMA may end early or may run out when the Master still wants more.

Table 334. Target DMA Control register (SDMACTRL, offset = 0x20)

Bit	Symbol	Value	Description	Reset value
1:0	DMAFB		DMA Read (From-bus) Trigger Enables DMA reads. If enabled with 1 or 2, DMAFB requests DMA on receive trigger (see SDATACTRL). It requests until empty unless the DMA is set up as a trigger. DMAFB becomes 0 when SSTATUS[ERRWARN] becomes 1.	0x0
		0	DMA not used	
		1	DMA is enabled for one frame. Automatically becomes 0 on STOP or repeated START. See SCONFIG[MATCHSS] .	
		2	DMA is enabled until it is turned off.	
		3	Reserved	
3:2	DMATB		DMA Write (To-bus) trigger. If enabled with 1 or 2, DMATB will start a request DMA on a TX trigger; see the Slave Data Control Register (SDATACTRL). DMATB will request until full unless the DMA is set up as a trigger. DMATB will cancel on MSTATUS.ERRWARN.	0x0
		0	NOT_USED: DMA is not used	
		1	ENABLE_ONE_FRAME: DMA is enabled for 1 Frame (ended by DMA or terminated). DMATB auto-clears on a STOP or START (see the Match START or STOP bit (SCONFIG.MATCHSS).	
		2	ENABLE: DMA is enabled until turned off. Normally, ENABLE should only be used with Master Message mode.	
		3	Reserved	
5:4	DMAWIDTH		Width of DMA operations. The width of DMA operations, if configured to allow half-word data access.	0x1
		0	BYTE	
		1	BYTE_AGAIN	
		2	HALF_WORD: Half word (16 bits). This will make sure that 2 bytes are free/available in the FIFO.	
		3	Reserved.	
31:6	-	-	Reserved.	-

21.8.10 Target Data Control

Assists in data control when no FIFO is used. Also assists in data control of FIFO when the FIFO is available (regardless of size) allowing some control over the FIFO behavior. This register allows control of when to interrupt based on fullness or emptiness of a buffer or FIFO. It also controls behavior related to width, when the buffer or FIFO is not one byte wide.

Table 335. Target Data Control register (SDATACTRL, offset = 0x2C)

Bit	Symbol	Value	Description	Reset value
0	FLUSHTB	-	Flush the to-bus buffer/FIFO. Used when the master terminates a to-bus (read) message prematurely.	0x0
1	FLUSHFB	-	Flushes the from-bus buffer/FIFO. Not normally used.	0x0
2	-	-	Reserved.	-

Table 335. Target Data Control register (SDATACTRL, offset = 0x2C) *…continued*

Bit	Symbol	Value	Description	Reset value
3	UNLOCK	-	Unlock UNLOCK must be 1 in the same cycle while writing to TXTRIG or RXTRIG. 0b - RXTRIG and TXTRIG fields cannot be changed on a write. 1b - RXTRIG and TXTRIG fields can be changed on a write.	-
5:4	TXTRIG		Trigger level for TX FIFO emptiness. Affects interrupts(if enabled). The default is Trigger on 1 less than full or less (=3).	0x3
		0	Trigger on empty	
		1	Trigger on 1/4 full or less	
		2	Trigger on 1/2 full or less	
		3	Trigger on 1 less than full or less (Default)	
7:6	RXTRIG		Trigger level for RX FIFO fullness. Affects interrupts(if enabled).	0x0
		0	Trigger on not empty	
		1	Trigger on 1/4 or more full	
		2	Trigger on 1/2 or more full	
		3	Trigger on 3/4 or more full	
15:8	-	-	Reserved.	-
20:16	TXCOUNT	-	Count of bytes in TX.	0x0
23:21	-	-	Reserved.	-
28:24	RXCOUNT	-	Count of bytes in RX.	0x0
29	-	-	Reserved.	-
30	TXFULL		Transmit full.	0x0
		0	TX is not full	
		1	TX is full	
31	RXEMPTY		Receive empty.	0x1
		0	RX is not empty	
		1	RX is empty	

21.8.11 Target Write Data Byte

Allows writing a byte to the bus (to controller) unless an external FIFO is used. Writing a byte requires a byte plus an
end-of-data (last) marker bit. A byte should not be written unless there is room, indicated by SSTATUS[TXNOTFULL] = 1.

Table 336. Target Write Data Byte register (SWDATAB, offset = 0x30)

Bit	Symbol	Description	Reset value
7:0	DATA	Data byte to send to master.	-
8	END	End This field is required for I3C, but is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs. 0b - Not the end. There are more bytes in the message. 1b - End. This bit marks the last byte of the message.	-
15:9	-	Reserved.	-
16	END_ALSO	End Also This field is required for I3C, but is optional for I2C. For HDR-DDR, the byte with the END_ALSO must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs. 0b - Not the end. There are more bytes in the message. 1b - End. This bit marks the last byte of the message.	-
31:17	-	Reserved.	-

21.8.12 Target Write Data Byte End

Allows writing a byte to the bus (to controller) unless an external FIFO is used. Unlike SWDATABm writing a byte only requires the byte itself, and is marked as end-of-data (last byte). A byte should not be written unless there is room, indicated by SSTATUS[TXNOTFULL] = 1.
For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs.

Table 337. Target Write Data Byte End (SWDATABE, offset = 0x34)

Bit	Symbol	Description	Reset value
7:0	DATA	Data byte to send to master.	-
31:8	-	Reserved.	-

21.8.13 Target Write Data Half-word

Allows writing a half-word (pair of bytes) to the bus unless an external FIFO is used. Sends the low byte followed by the high byte. The 16th bit marks the end; that is, the last byte of the half-word is the end.
An end-of-data (last) marker bit is allowed (or must be 0). A half-word should not be written unless there is room for both, as indicated by the use of Transmit FIFO level trigger or SDATACTRL[TXCOUNT].

Table 338. Target Write Data Half-word register (SWDATAH, offset = 0x38)

Bit	Symbol	Description	Reset value
7:0	DATA0	Data 0 The first byte to send to the controller	-
15:8	DATA1	Data 1 The second byte to send to the controller	-
16	END	End of message This field always marks DATA1 as the end. This field is required for I3C, but is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs. 0b - Not the end. There are more bytes in the message. 1b - End. This bit marks the last byte of the message.	-
31:17	-	Reserved.	-

21.8.14 Target Write Data Half-word End

Allows writing a half word of data, which is the end (the last byte of the half word is the end). Writes the half word (byte pair) just like Target Write Data Half-word (SWDATAH) , but marks the second byte as end-of-data (last byte).
For HDR-DDR, the byte with the END must be an even (second, fourth, sixth, and so on) because DDR uses byte-pairs.
A half-word should not be written unless there is room for both, as indicated by the use of Transmit FIFO level trigger or SDATACTRL[TXCOUNT] .

Table 339. Target Write Data Half-word End register (SWDATAHE, offset = 0x3C)

Bit	Symbol	Description	Reset value
7:0	DATA0	Data 0 The first byte to send to the controller	-
15:8	DATA1	Data 1 The second byte to send to the controller	-
31:16	-	Reserved.	-

21.8.15 Target Read Data Byte

Allows reading a byte from the bus (controller). A byte should not be read unless there is data waiting, as indicated by SSTATUS[RX_PEND] = 1.

Table 340. Target Read Data Byte register (SRDATAB, offset = 0x40)

Bit	Symbol	Description	Reset value
7:0	DATA0	Data 0 Byte read from the controller	0x0
31:8	-	Reserved.	-

21.8.16 Target Read Data Half-word register

Allows reading a halfword (byte pair) written by the target after an SDR Read or DAA or DDR. This register is only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively. A half word should not be read unless there are at least two bytes of data waiting, as indicated the Receive FIFO level trigger or SDATACTRL[RXCOUNT] .

Table 341. Target Read Data Half-word register (SRDATAH, offset = 0x54)

Bit	Symbol	Description	Reset value
7:0	LSB	LSB Represents the first byte read from the controller (and written by the target).	0x0
15:8	MSB	MSB Represents the second byte read from the controller (and written by the target).	0x0
31:16	-	Reserved.	-

21.8.17 Target Write Data Byte (SWDATAB1)

Allows writing a single byte to the bus (to controller) such that only bits 7:0 are used. Intended for use by DMAs, which do not format the upper part of the APB word.

A byte should not be written unless there is room, as indicated by SSTATUS[TXNOTFULL] = 1.

Table 342. Target Write Data Byte(SWDATAB1, offset = 0x54)

Bit	Symbol	Description	Reset value
31:8	_	Ignored field	0x0
7:0	DATA	Data Byte to send to controller	0x0

21.8.18 Target Capabilities

Indicates which features are available and supported in this I3C module, including master and/or slave capabilities, HDR modes, and others.

Table 343. Target Capabilities register (SCAPABILITIES, offset = 0x60)

Bit	Symbol	Value	Description	Reset value
1:0	IDENA		ID 48b handler. Indicates who handles the ID 48b value.	0x0
		0	APPLICATION: Application handles ID 48b	
		1	HW: Hardware handles ID 48b	
		2	HW_BUT: in hardware but the I3C module instance handles ID 48b.	
		3	PARTNO: a part number register (PARTNO) handles ID 48b	
5:2	IDREG		ID Register Indicates which ID features are in the registers compared to what is in the hardware. 0000b - All ID register features below are disabled. 1xxxb - A Bus Characteristics Register (BCR) is available. x1xxb - A Device Characteristic Register (DCR) is available. xx1xb - An ID Random field is available. xxx1b - ID Instance is a register, and is used if there is no PARTNO register.	0xE

Table 343. Target Capabilities register (SCAPABILITIES, offset = 0x60) *…continued*

Bit	Symbol	Value	Description	Reset value
7:6	HDRSUPP		High Data Rate Support Indicates which HDR modes are supported. 00b - No HDR modes supported 01b - Double Data Rate mode supported All other values are reserved.	0x1
8	-	-	Reserved	-
9	MASTER		Master. Specifies if a master capability is supported or not.	0x1
		0	MASTERNOTSUPPORTED: master capability is not supported.	
		1	MASTERSUPPORTED: master capability is supported.	
11:10	SADDR		Static address. Indicates how the static address is handled.	0x3
		0	NO_STATIC: No static address	
		1	STATIC: Static address is fixed in hardware	
		2	HW_CONTROL: Hardware controls the static address dynamically (for example, from the pin strap)	
		3	CONFIG: SCONFIG register supplies the static address	
15:12	CCCHANDLE		Indicates who manages CCC between I3C module and the user application. 0000b - All handling features below are disabled. 1xxxb - GETSTATUS CCC returns SCTRL[VENDINFO] value. x1xxb - GETSTATUS CCC returns SCTRL[PENDINT] and SCTRL[ACTSTATE] values. xx1xb - The block manages maximum read and write lengths, and max data speed. xxx1b - The block (I3C module) manages events, activities, status, HDR, and if enabled for it, ID and static-address-related items.	0xF
20:16	IBI_MR_HJ		In-Band Interrupts, Controller Requests, Hot-Join Events Indicates which events (IBI, CR, and HJ) are allowed. For example, if this field is 00011b, IBI (bit 0) and IBI_HAS_DATA (bit 1) functionality are both enabled. 0_0000b - Application cannot generate IBI, CR, or HJ. 1_xxxxb - Application can use SCONFIG[BAMATCH] for bus-available timing. x_1xxxb - Application can generate a Hot-Join event. x_x1xxb - Application can generate a Controller Request for a secondary controller. x_xx1xb - When bit 0 = 1, the IBI has data from the SCTRL register. x_xxx1b - Application can generate an IBI.	0x1F
21	TIMECTRL		Time control. Specifies if any time-control type is supported or not.	0x1
		0	NO_TIME_CONTROL_TYPE: No time control is enabled	
		1	ATLEAST1_TIME_CONTROL: at least one time-control type is supported	
22	-	-	Reserved.	-
25:23	EXTFIFO		External FIFO. Indicates whether External FIFOs are enabled. If External FIFOs are not enabled, then check FIFOTX and FIFORX for the internal FIFO.	0x0
		0	NO_EXT_FIFO: No external FIFO is available	
		1	Standard available or free external FIFO	
		2	Request track external FIFO	
		3	reserved	

Table 343. Target Capabilities register (SCAPABILITIES, offset = 0x60) *…continued*

Bit	Symbol	Value	Description	Reset value
27:26	FIFOTX		FIFO transmit. Indicates if TX (to-bus) is enabled and what size it is.	undefined
		0	FIFO_2BYTE: 2-byte TX FIFO, the default FIFO transmit value (FIFOTX)	
		1	FIFO_4BYTE: 4-byte TX FIFO	
		2	FIFO_8BYTE: 8-byte TX FIFO	
		3	FIFO_16BYTE: 16-byte TX FIFO	
29:28	FIFORX		FIFO receive. Indicates if RX (from-bus) is enabled and what size it is.	0x2
		0	FIFO_2BYTE: 2 (or 3)-byte RX FIFO, the default FIFO receive value (FIFORX)	
		1	FIFO_4BYTE: 4-byte RX FIFO	
		2	FIFO_8BYTE: 8-byte RX FIFO	
		3	FIFO_16BYTE: 16-byte RX FIFO	
30	INT		Interrupts support.	0x1
		0	Interrupts are not supported	
		1	Interrupts are supported	
31	DMA		DMA support.	0x1
		0	DMA is not supported	
		1	DMA is supported	

21.8.19 Slave Dynamic Address

Contains the dynamic address after being assigned; otherwise =0.

The Slave Dynamic Address register is filled in with the assigned address once the Master has assigned it via SETDASA or ENTDAA CCC commands. It will clear if the RESETDAA CCC is used. The current validity state is also indicated via the SSTATUS.DAVALID bit, which can be used to interrupt the processor.

If configured to allow write, this is normally only used to restore the DA after a power-down (an ultra-low power state that loses power to peripherals but retains the DA somewhere else). This is not needed if state-retention flops are used for the DA. This mechanism only allows writes when Slave is disabled (and it will be ignored otherwise). If the Master uses RSTDAA or SETNEWDA, then it will over-ride this mechanism and cede (yield) to the master-assigned DA. Note that when when enabling the Slave, the SCONFIG.OFFLINE bit should also be set. This will wait for evidence that the bus is not in I3C HDR mode; and will exit when an HDR Exit pattern is seen or when 60 usec has expired. This makes it safe to monitor START and STOP. If the application needs to do an IBI, then the application should either wait for a STOP (see STATUS) or make sure that 200 usec have gone by with no activity (no START or STOP) before the app emits the IBI.

The MAPIDX/MAPSA model allows writing additional DAs (and SAs) into a list (the number in the list is pre-configured); any DA or SA with DAVALID=0 are never matched. The additional DAs may be based on the bridge target CCC, or done by a move (read current DA and then copy into upper map, then invalidate the 0 map) when matching ENTDAA over and over. Any mapped location can be invalidated by writing the MAPIDX and DAVALID=0. The mapped ones are not reset by the RSTDA CCC. See the SMSGMAPADDR register.

To copy the base DA to the mapped set, the configuration has to be set up to allow it, otherwise they are distinct mechanisms.

- If used for the copy model, then a special reset feature is allowed. In this case, the SDYNADDR register is written with KEY=CB19 and the rest 0. This will clear the DA and then self-clear.
- If used in I3C mode, a base DA is required, so the last one should not be cleared (or writing one with DAVALID=1 is necessary).

Table 344. Slave Dynamic Address register (SDYNADDR, offset = 0x64)

Bit	Symbol	Value	Description	Reset value
0	DAVALID		Dynamic address valid. Determines if a Dynamic Address is assigned.	0x0
		0	DANOTASSIGNED: a Dynamic Address is not assigned	
		1	DAASSIGNED: a Dynamic Address is assigned	
7:1	DADDR	-	Dynamic address. This is the assigned Dynamic Address, when DAVALID is 1.	0x0
11:8	MAPIDX	-	Mapped Dynamic Address. Selects which mapped DA to write to, with MAPIDX=0 meaning the DA is not mapped (which is the normal use of write SDYNADDR if allowed). This allows for a list of matching DAs or SAs (I2C static addresses). Note that the mechanism is write-only.	-
12	MAPSA	-	Map a Static Address. If MAPSA=1 on a write with MAPIDX!=0, then this sets a static address into the list;, otherwise a dynamic address is used.	-
15:13	-	-	Reserved.	-
31:16	KEY	-	Key. Must set to 0xA4D9 to write DADDR field (and set DAVALID bit to 1). Only writable when a slave is not enabled (for restoring after power-down sleep with auto-restore). The mapped locations and base may be written when a slave is enabled, but care should be taken to not do that when there are transactions on the I3C bus. KEY reads back as 1 if overwritten, else KEY is 0 if assigned by the master, including when the master changes it. If address mapping is allowed, then writing with KEY=0xCB19 is used to clear the base DA.	0x0

21.8.20 Target Maximum Limits

Indicates the limits set by the controller (or the original requested limits). The maximum limits are not enabled in the hardware design, including maximum read and write lengths. If the maximum read and write lengths are enabled, then the current setting (including default request) shows up in this register (SMAXLIMITS).

Table 345. Target Maximum Limits register (SMAXLIMITS, offset = 0x68)

Bit	Symbol	Description	Reset value
11:0	MAXRD	Indicates the limits set by the controller (or the original requested limits). The maximum limits are not enabled in the hardware design, including maximum read and write lengths. If the maximum read and write lengths are enabled, then the current setting (including default request) shows up in this register (SMAXLIMITS).	0x0
15:12	-	Reserved.	-
27:16	MAXWR	Maximum Write Length Indicates the maximum write length, which must be between 8 to 4095 (saturation). The application must not set the maximum write length to a higher value than the maximum write length set by the controller.	0x0
31:28	-	Reserved.	-

21.8.21 Target ID Part Number

Allows an application to write the ID part-number. The application must write a value into the PARTNO field, because normally, 0 is not valid.

Table 346. Target ID Part Number register (SIDPARTNO, offset = 0x6C)

Bit	Symbol	Description	Reset value
31:0	PARTNO	Part number.	0x0

21.8.22 Target ID Extension

Allows an application to write the ID extension of the Device Characteristic Register (DCR) and/or the Bus Characteristics Register (BCR).

Table 347. Target ID Extension register (SIDEXT, offset = 0x70)

Bit	Symbol	Description	Reset value
7:0	-	Reserved.	-
15:8	DCR	Device Characteristic Register. Set the Device Characteristic Register (DCR) if configured for it. (SIDEXT)	0x0
23:16	BCR	Bus Characteristics Register Sets the value for the Bus Characteristics Register (BCR), if this field is configured. This field controls features such as Secondary Controller and slow speed requirements.	0x0
31:24	-	Reserved.	-

21.8.23 Target Vendor ID

Allows an application to write the Vendor ID. The default value is the chip vendor ID, and is set from the constant field. If using the chip vendor ID, the part number (PARTNO) does not collide with other uses. The MIPI Vendor ID is available to all companies (MIPI membership is not required). To get a vendor ID make a request at the mipi.org website

Table 348. Target Vendor ID register (SVENDORID, offset = 0x74)

Bit	Symbol	Description	Reset value
14:0	VID	Vendor ID Can be set to the 15-bit MIPI Vendor ID.	0x11B
31:15	-	Reserved.	-

21.8.24 Target Time Control Clock

Allows an application to dynamically set the time control clock and accuracy information. The clock frequency and accuracy are constants set by the hardware. If the clock can be adjusted (that is, divided) or if the accuracy could vary with knowable information, then the clock may be set via this register. This register must be updated whenever the clock source is changed.

Table 349. Target Time Control Clock register (STCCLOCK, offset = 0x78)

Bit	Symbol	Description	Reset value
7:0	ACCURACY	Clock Accuracy Indicates the clock accuracy in 1/10ths of %. For example, a value of 15 indicates an accuracy of 1.5%. Default set by parameters if configured.	0x14
15:8	FREQ	Clock Frequency Indicates the clock frequency in 0.5-MHz steps. For example, a value of 20 in this field indicates a frequency of 10 MHz. Default set by parameters if configured..	0x2
31:16	-	Reserved.	-

21.8.25 Target Message Last Matched (SMSGLAST)

Shows the current/last message (SSTATUS[MATCHED]) with information on which DA or SA (if more than one via Mapped), Group (if used), and what mode was used among SDR, DDR, or BT. The SMSGLAST register holds the last 3 matches.

Table 350. Target Message Last Matched register (SMSGLAST, offset = 0x7C)

Bit	Symbol	Value	Description	Reset value
3:0	MAPLAST		Matched address index Matched address index for current or last matched message. 0 for the base address. Only valid if mapped address list is enabled or groups allowed. If LASTGROUP, then 0, 1, or 2 for group number.	0x0
4	LASTSTATIC		Last Static Address Matched 1 if last matched address was an I2C Static address vs. an I3C Dynamic Address	0x0
5	LASTGROUP		Last Match Was Group 1 if last matched address was a group.	0x0
7:6	LASTMODE		Last Mode Indicates which mode the access was in:	0x0
		00	SDR mode (or I2C if static) was the last mode.	
		01	HDR-DDR was the last mode.	
		10	HDR-BT was the last mode	
		11	Reserved	
11:8	MAPLASTM1		Previous match index 1 0 for the base address.	0x0
12	_	_	Reserved	
13	LASTGROUP M1	_	Last Match 1 Previous Was Group Previous match was a group.	
15:14	LASTMODE1		Last Mode 1 Indicates which mode the previous access was in:	0x0
		00	SDR mode (or I2C if static) was the previous last mode.	
		01	HDR-DDR was the previous last mode.	
		10	HDR-BT was the previous last mode	
		11	Reserved	

Table 350. Target Message Last Matched register (SMSGLAST, offset = 0x7C) *…continued*

Bit	Symbol	Value	Description	Reset value
19:16	MAPLASTM2		Previous match index 2 0 for the base address.	0x0
20	_	_	Reserved	
21	LASTGROUP M2		Last Match 2 Previous Was Group Previous match was a group.	0x0
23:22	LASTMODE2		Last Mode 2 Indicates which mode the previous access was in:	0x0
		00	SDR mode (or I2C if static) was the last mode 2 previous accesses ago.	
		01	HDR-DDR was the last mode 2 previous accesses ago.	
		10	HDR-BT was the last mode 2 previous accesses ago.	
		11	Reserved	
31:24	_	_	Reserved	

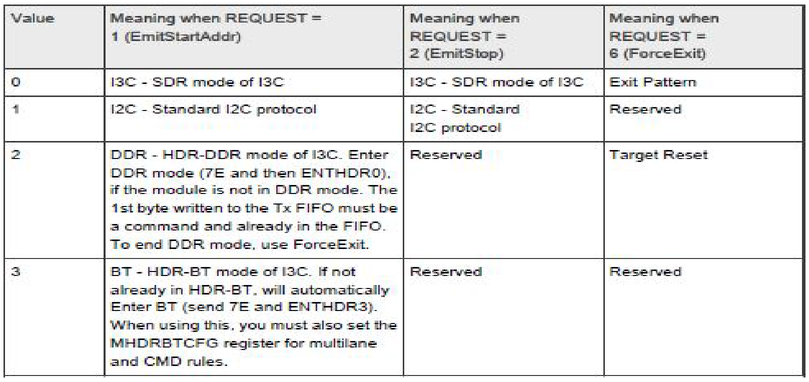
21.8.26 Controller Control

Starts activities on the I3C or I2C bus. Also see the MWMSG register. A request cannot be changed when a message is in progress; the REQUEST field will become 0 automatically.

NOTE: Write MCTRL fields as per use case or as mentioned in Operating modes. Bit read/write checking may not work independently.

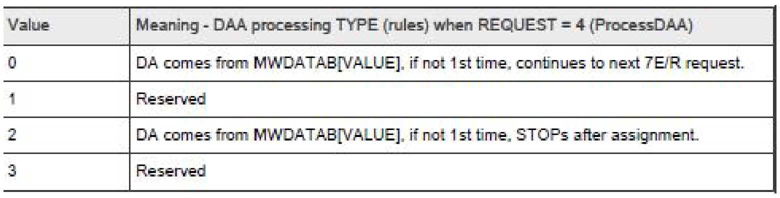
NOTE: If MCONFIG[MSTENA] is set for I2C Controller mode (legacy I2C), only REQUEST = 1 and REQUEST = 2 are accepted. Also, fields are constrained to I2C-supported fields.

Table 351. Controller Control register (MCTRL, offset = 0x84)

Bit	Symbol	Value	Description	Reset value
2:0	REQUEST		Emits the requested operation when performing in pieces, instead of performing by message. The Controller Status (MSTATUS) register should be checked because some requests can only be made in some states. For example, the system cannot enter SDR mode from DDR mode, and it cannot use an incorrect request in DAA mode.	0x0
		000b	NONE. Indicates that no request is present. The REQUEST field returns to NONE when finished with any request. The MSTATUS register indicates the state of the controller. See also AutoIBI mode. NONE is only written as 0 in these cases: when writing 1 to MCTRL[RDTERM] (to stop a read in progress) or when setting MCTRL[IBIRESP] for MSG use.	
		001b	EMITSTARTADDR. Emit START with address and direction, either from a stopped state or in the middle of a Single Data Rate (SDR) message. If from a stopped state (IDLE), then emit start may be prevented by an event (like IBI, CR, HJ). In this case the appropriate interrupt is signaled. Emit START can be resubmitted	
		010b	EMITSTOP. Emit a STOP on bus. Must be in SDR mode. In Dynamic Address Assignment (DAA) mode, emitting stop exits DAA mode.	
		011b	IBIACKNACK. Manual IBI ACK or NACK. When MCTRL[IBIRESP] has indicated a hold on an IBI to allow a manual decision, this request completes it. Uses MCTRL[IBIRESP] to provide the information.	
		100b	PROCESSDAA. If not currently in Dynamic Address Assignment (DAA) mode, emits START, 7E, ENTDAA sequence, then emits 7E/R to process the first target. Stops just before the new Dynamic Address (DA) is to be emitted. The DA is written using Controller Write Data Byte (MWDATAB), then Process DAA is requested again to write the new address, and then it starts the next unless marked to STOP. An MSTATUS indicating NACK means DA was not accepted (for example, parity error). If PROCESSDAA is NACKED on the 7E/R request, meaning no more targets need a DA, then a COMPLETE is signaled (along with DONE) and a STOP issued. If TYPE = 2 or TYPE = 3, the DA is assigned and then it emits a STOP (instead of starting a new 7E/R request). If TYPE = 1 or TYPE = 3, then the DA is taken from the ADDR field (bits 6:0).	
		101b	Reserved	
		110b	Force Exit and Target Reset. Emit an Exit Pattern from any state. End Double Data Rate (DDR) mode (including MSGDDR), including a STOP afterward. If MCTRL[TYPE] is 2, perform a Target Reset action (RSTACT can prevent the reset in the target). Target Reset may follow immediately after RSTACT CCC or after STOP.	
		111b	AUTOIBI. Hold in a stopped state, but auto-emit a START, 7E sequence when the target holds SDA low for an IBI. Actual IBI handling is defined by MCTRL[IBIRESP].	
3	-	-	Reserved.	-
5:4	TYPE	Bus Type with EmitStartAddr 		0x0

Value	Meaning when REQUEST = 1 (EmitStartAddr)	Meaning when REQUEST = 2 (EmitStop)	Meaning when REQUEST = 6 (ForceExit)
0	I3C - SDR mode of I3C	I3C - SDR mode of I3C	Exit Pattern
1	I2C - Standard I2C protocol	I2C - Standard I2C protocol	Reserved
2	DDR - HDR-DDR mode of I3C. Enter DDR mode (7E and then ENTHDR0), if the module is not in DDR mode. The 1st byte written to the Tx FIFO must be a command and already in the FIFO. To end DDR mode, use ForceExit.	Reserved	Target Reset
3	BT - HDR-BT mode of I3C. If not already in HDR-BT, will automatically Enter BT (send 7E and ENTHDR3). When using this, you must also set the MHDRBTCFG register for multilane and CMD rules.	Reserved	Reserved

Table 351. Controller Control register (MCTRL, offset = 0x84) *…continued*

Bit	Symbol	Value	Description	Reset value
				

Value	Meaning - DAA processing TYPE (rules) when REQUEST = 4 (ProcessDAA)
0	DA comes from MWDATAB[VALUE], if not 1st time, continues to next 7E/R request.
1	Reserved
2	DA comes from MWDATAB[VALUE], if not 1st time, STOPs after assignment.
3	Reserved

Bit	Symbol	Value	Description	Reset value
		The meaning of the field settings change according to REQUEST value. 00b - I3C. When REQUEST = 1 (EmitStartAddr) or 2 (EmitStop), the SDR mode of I3C. When REQUEST = 6 (ForceExit), the Exit pattern. When REQUEST = 4, DA comes from MWDATAB[VALUE], if not 1st time, continues to next 7E/R request. 01b - I2C. When REQUEST = 1 (EmitStartAddr) or 2 (EmitStop), the Standard I2C protocol. When REQUEST = 4, reserved. When REQUEST = 6, reserved. 10b - DDR. When REQUEST = 1, the HDR-DDR mode of I3C. Enter DDR mode (7E and then ENTHDR0), if the module is not in DDR mode. The 1st byte written to the Tx FIFO must be a command and already in the FIFO. To end DDR mode, use ForceExit. When REQUEST = 2, reserved. When REQUEST = 4, DA comes from MWDATAB[VALUE], if not 1st time, STOPs after assignment. When REQUEST = 6, Target Reset. 11b - Reserved		
7:6	IBIRESP		In-Band Interrupt (IBI) response. Indicates the response to use when you get an IBI from START, and when to force using a IBI ACK NACK request when completing a manual IBI. Completion of a manual IBI means that the target DA is known, and so the mandatory byte (or not) is specified by the application when acknowledging. Note: Controller Request and Hot-Join always cause IBIRESP = 3 (Manual) so the application must decide. The Mctrl[IBIRESP] field is also used when a message is emitted in Message mode using the MWMSG_SDR or MWMSG_DDR registers. If an IBI with MDB (mandatory byte) is ACKed, the controller limits it to a max of eight more bytes after the MDB. RDTERM = 1 can be used with Request = None to terminate data from a target sooner.	0x0
		00b	ACK (acknowledge). When REQUEST = 1 (EmitStartAddr) or REQUEST = 7 (AutoIBI), ACK with mandatory byte (or not), decided by the Controller In-band Interrupt Registry and Rules (MIBIRULES) register. When REQUEST = 3 (IBIAckNack), ACK with no mandatory byte.	
		01b	NACK (reject). Not acknowledge	
		10b	Acknowledge with mandatory byte. When REQUEST = 1 or REQUEST = 7, ignore the MIBIRULES register. Do not use this setting unless only targets *with a mandatory byte* can cause an IBI. When REQUEST = 3, ACK with mandatory byte.	
		11b	Manual. When REQUEST = 1 or REQUEST = 7, stop and wait for a decision using the IBI ACK NACK request. When REQUEST = 3, reserved.	

Table 351. Controller Control register (MCTRL, offset = 0x84) *…continued*

Bit	Symbol	Value	Description	Reset value
8	DIR	-	Direction.	0x0
		0	DIRWRITE: Write	
		1	DIRREAD: Read	
15:9	ADDR	-	Address I3C Dynamic Address or I2C Static Address. Some values are not allowed based on which bus is used (I3C or I2C).	0x0
23:16	RDTERM	-	Read Terminate Counter Determines when to terminate a read operation. For I2C, controls when to NACK a read. For I3C, can be used to terminate (end) a read. RDTERM = 0 has no effect. RDTERM = 1 terminates after the next character. RDTERM = 2 terminates after the next two characters. Supports up to 255 characters. In DDR mode, RDTERM terminates the read based on word counts (for DDR) instead of byte counts (for SDR). Note: The value 1 may be written any time to this field, but a number larger than 1 should be written when starting EmitStartAddress. Self clears on COMPLETE.	0x0
31:24	-	-	Reserved.	-

21.8.27 Controller Status

Status for the controller, including which events cause interrupts. The peripherals share the IRQ (called Parallel-to-Target status).

Because a peripheral can either be in Controller or Target mode, but not both at the same time, only one (Target or Controller peripheral) can be the cause of the IRQ. If there is an IRQ and the peripheral is a controller, then this register (MSTATUS) has the status.
If there is an IRQ and the peripheral is a target, then the Target Status (SSTATUS) register has the status. Self clears on COMPLETE.

If MCONFIG is set for I2C only, some states and bits will never be active (for example, DAA or IBI)..

Table 352. Controller Status register (MSTATUS, offset = 0x88)

Bit	Symbol	Value	Description	Reset value
2:0	STATE		State Of The Controller Indicates the current controller state	0x0
		000b	IDLE. Bus has stopped.	
		001b	SLVREQ. Target request. The bus has stopped but a target is holding SDA low. If using auto-emit IBI (MCTRL[REQUEST] = 7), the controller will not remain in this state.	
		010b	MSGSDR. Single Data Rate Message mode, from using MWMSG_SDR	
		011b	NORMACT. Normal active SDR mode, from using MCTRL and MWDATAn and MRDATAn registers. The controller remains in this state until a STOP is issued.	
		100b	MSGDDR. Double Data Rate Message mode, from using MWMSG_DDR or using the normal method with DDR. The controller remains in the DDR state until the controller exits using EXIT (emitting the Exit pattern).	
		101b	DAA: in Enter Dynamic Address Assignment (ENTDAA) mode	
		110b	IBIACK: waiting for an In-Band Interrupt (IBI) ACK/NACK decision	
		111b	IBIRCV: Receiving an In-Band Interrupt (IBI); this IBIRCV state is used after IBI/MR/HJ has won the arbitration, and IBIRCV state is also used for IBI mandatory byte (if any) and any bytes that follow.	
3	-	-	Reserved.	-
4	BETWEEN	-	Between Between messages or Dynamic Address Assignments (DAA). Active when: •MSTATUS[STATE] is MSGSDR, DDR, or DAA, and the state is between messages/DAAs. It is expecting a new messages/DAAs to start (or STOP or Exit). •MSTATUS[STATE] is NORMACT. The module is waiting on the Transmit FIFO to be not empty or the Receive FIFO to be not full. 0b - Inactive. For other cases. 1b - Active. If STATE is MSGSDR, DDR, DAA, or NORMACT.	0x0
5	NACKED	-	Not Acknowledged Indicates whether the last Start and Address sequence was NACKed (was not ACKed by the addressed target). 0b - Not NACKed 1b - NACKed (not acknowledged)	0x0
7:6	IBITYPE		In-Band Interrupt (IBI) Type Indicates the type of IBI of the last event that won the arbitration, whether the interrupt is ACKED or NACKED or pending. 00b - NONE. No IBI. This status occurs when MSTATUS[IBIWON] becomes 0. 01b - In-Band Interrupt 10b - Controller Request 11b - Hot-Join	0x0
8	SLVSTART	-	Target Start Indicates whether a target is/was requesting a START by holding SDA low. Handling starts automatically when MCTRL[REQUEST] = 7 (AutoIBI). 0b - Target not requesting START 1b - Target requesting START	0x0

Table 352. Controller Status register (MSTATUS, offset = 0x88) *…continued*

Bit	Symbol	Value	Description	Reset value
9	MCTRLDONE	-	Controller Control Done Indicates whether the module has completed an MCTRL request. MCTRLDONE automatically becomes 0 when writing a new control. When MCTRL[REQUEST] = 1 (EmitStartAddr), MCTRLDONE becomes 1 when the address goes out (and is ACKed, NACKed, or ended in an IBI). If ACKed, MSTATUS[COMPLETE] becomes 1 when the write or read data has completed. When MCTRL[REQUEST] = 4 (ProcessDAA), MCTRLDONE becomes 1 when the module is ready to emit the Dynamic Address (DA) for the target, or when no more targets are ACKing. This condition can be determined by using the MSTATUS[BETWEEN] and MSTATUS[STATE] fields. 0b - Not done 1b - Done	0x0
10	COMPLETE	-	Complete Indicates whether a message has completed. • With MWMSG_SDR or MWMSG_DDR, this condition occurs when MWMSG_SDR_CONTROL[LEN] or MWMSG_DDR_CONTROL[LEN] reaches 0. • When MCTRL[REQUEST] = 1 (EmitStartAddr), this condition occurs after the end of a write operation, or after a read operation has terminated or ended. • With an IBI (AutoIBI or EmitStartAddr), this condition occurs at the end of IBI data (if any). 0b - Not complete 1b - Complete	0x0
11	RXPEND	-	RXPEND Indicates whether a message is being received from a target and bytes are in the input buffer or FIFO. • If using a FIFO, this message is at least one FIFO trigger's worth (a minimum of one byte in the FIFO). • If DMA is enabled for receiving, the DMA is signaled. RXPEND becomes 0 when the data is read. 0b - No receive message pending 1b - Receive message pending	0x0
12	TXNOTFULL	-	TX Buffer or FIFO Not Full Indicates whether the buffer, FIFOm or message register can accept another byte or halfword. FIFO uses trigger level. If DMA enabled for transmitting, it transfers data as long as it is not full. 0b - Receive buffer or FIFO full 1b - Receive buffer or FIFO not full	0x1

Table 352. Controller Status register (MSTATUS, offset = 0x88) *…continued*

Bit	Symbol	Value	Description	Reset value
13	IBIWON	-	In-Band Interrupt (IBI) Won Indicates whether an IBI, CR, or HJ has won the arbitration on a header address, regardless of whether it was NACKed or ACKed. Arbitration requires manual intervention for CR and HJ, and optionally requires it for IBI if the MCTRL[IBIRESP] = 3 (Manual). 0b - No IBI arbitration won 1b - IBI arbitration won	0x0
14	-	-	Reserved.	-
15	ERRWARN		Error Or Warning Indicates whether an error occurred, such as improper register use, overrun or underrun of FIFO or buffer, or invalid parity or CRC in a DDR read. See the Controller Errors and Warnings (MERRWARN) register. 0b - No error or warning 1b - Error or warning	0x0
18:16	-	-	Reserved.	-
19	NOWMASTER	-	Module Is Now Controller Indicates when the module is now a controller. That is, it was previously a target, controllership acceptance was requested from the previous controller, and controllership was accepted. The reverse operation (controller becomes a target) does not need an interrupt, because the application grants it through the GETACCMST CCC. 0b - Module has not become controller 1b - Module has become controller	0x0
23:20	-	-	Reserved.	-
30:24	IBIADDR	-	IBI Address Indicates the address of: The IBI, when MSTATUS[IBITYPE] = 1. The Controller Request (CR), when MSTATUS[IBITYPE] = 2. 7'h2 when Hot-Join, when MSTATUS[IBITYPE] = 3.	0x0
31	-	-	Reserved.	-

21.8.28 Controller In-band Interrupt Registry and Rules

Contains the rules for using IBI, and keeps a registry of the targets that use the IBI byte.

Defines the IBI mandatory byte rules for the targets. Determines which targets do (or do not) have a mandatory byte.

Concerning ADDRn fields: The address is six bits. Assuming that MIBIRULES[MSB0] = 1, the most significant bit of each address is 0. In this case, each address is seven bits (usually written as A7 to A1) and A7 must be 0. If the application does not use that optimal convention, the Manual method of IBI ACK handling must be used (see MCTRL[IBIRESP]).

By default, the ADDRn values indicate the targets with a mandatory byte. If MIBIRULES[NOBYTE] = 1, the ADDRn values indicate targets that do not have a mandatory IBI byte.

NOTE: A7 = 0 is only needed for targets that use IBI. For legacy I2C devices, A7 can use any valid value, because I2C devices cannot cause an IBI

Table 353. Master In-band Interrupt Registry and Rules register (MIBIRULES, offset = 0x8C)

Bit	Symbol	Description	Reset value
5:0	ADDR0	ADDR0 Address of target with or without Mandatory IBI byte. If 0, then the address does not apply.	0x0
11:6	ADDR1	ADDR1 Address of target with or without Mandatory IBI byte. If 0, then the address does not apply.	0x0
17:12	ADDR2	ADDR2 Address of target with or without Mandatory IBI byte. If 0, then the address does not apply	0x0
23:18	ADDR3	ADDR3 Address of target with or without Mandatory IBI byte. If 0, then the address does not apply.	0x0
29:24	ADDR4	ADDR4 Address of target with or without Mandatory IBI byte. If 0, then the address does not apply.	0x0
30	MSB0	Most Significant Address Bit Is 0 Assigning 1 to this field allows the START header to be optimized. 0b - MSB is not 0. 1b - For all I3C dynamic addresses, MSB is 0.	0x0
31	NOBYTE	No IBI byte Indicates whether the ADDRn fields refer to targets with or without a mandatory IBI byte. 0b - With mandatory IBI byte 1b - Without mandatory IBI byte	0x0

21.8.29 Controller Interrupt Set

Set interrupt enables for select bits in MSTATUS. Reading the MINTSET register returns the status of the interrupt enables.

- To activate an interrupt enable, write 1 to it.
- To disable an interrupt enable, write 1 to the appropriate bit in the Interrupt Clear Register (MINTCLR). Writing 0 to the interrupt enable in this register (MINTSET) does not disable the interrupt.

The Interrupt registers allow masking interrupt sources, as well as checking which interrupts have activated in the MSTATUS register. The normal method is to enable an interrupt and then once that interrupt fires, the interrupt is either cleared (by writing the MSTATUS register) or cleared by action (on the corresponding data register). The interrupt is level held, meaning that the interrupt stays set until the cause is cleared, by some method. The module prevents races, so that if a new event comes in, that new event will not be lost.

These interrupts are parallel to the Slave INTSET/INTCLR set, and only one interrupt set is active depending on state (Master or Slave operation).

- MINTSET: sets interrupt enables for MSTATUS bits. Reading MINTSET register returns the status of the interrupt enables.
- MINTCLR: clears interrupt enables for MSTATUS bits.
- MINTMASKED: returns the value of the MSTATUS bits ANDed with their interrupt enables.

Table 354. Master Interrupt Set register (MINTSET, offset = 0x90)

Bit	Symbol	Description	Reset value
7:0	-	Reserved.	-
8	SLVSTART	Target Start Interrupt Enable 0b - Disable 1b - Enable	0x0
9	MCTRLDONE	Controller Control Done Interrupt Enable 0b - Disable 1b - Enable	0x0
10	COMPLETE	Completed Message Interrupt Enable 0b - Disable 1b - Enable.	0x0
11	RXPEND	RX pending interrupt enable.	0x0
12	TXNOTFULL	TX buffer/FIFO is not full interrupt enable 0b - Disable 1b - Enable	0x0
13	IBIWON	In-Band Interrupt (IBI) Won Interrupt Enable 0b - Disable 1b - Enable	0x0
14	-	Reserved.	-
15	ERRWARN	Error or Warning (ERRWARN) Interrupt Enable 0b - Disable 1b - Enable	0x0
18:16	-	Reserved.	-
19	NOWMASTER	Now master (now this I3C module is a master) interrupt enable. 0b - Disable 1b - Enable	0x0
31:20	-	Reserved.	-

21.8.30 Controller Interrupt Clear

Clear interrupt enables for select MSTATUS bits. Writing a 1 clears the corresponding interrupt enable; writing a 0 has no effect.

Table 355. Controller Interrupt Clear register (MINTCLR, offset = 0x94)

Bit	Symbol	Description	Reset value
7:0	-	Reserved.	-
8	SLVSTART	SLVSTART Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
9	MCTRLDONE	MCTRLDONE Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
10	COMPLETE	COMPLETE Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
11	RXPEND	RXPEND Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
12	TXNOTFULL	TXNOTFULL Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
13	IBIWON	IBIWON Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
14	-	Reserved.	-
15	ERRWARN	ERRWARN Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
18:16	-	Reserved.	-
19	NOWMASTER	NOWCONTROLLER Interrupt Enable Clear 0b - No effect 1b - Corresponding interrupt enable becomes 0	-
31:20	-	Reserved.	-

21.8.31 Controller Interrupt Mask

Returns the status of enabled interrupts (the value of MSTATUS ANDed with MINTSET).

Table 356. Controller Interrupt Mask register (MINTMASKED, offset = 0x98)

Bit	Symbol	Description	Reset value
7:0	-	Reserved.	-
8	SLVSTART	SLVSTART Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x0
9	MCTRLDONE	MCTRLDONE Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x0

Table 356. Controller Interrupt Mask register (MINTMASKED, offset = 0x98) *…continued*

Bit	Symbol	Description	Reset value
10	COMPLETE	COMPLETE Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x0
11	RXPEND	RX pending interrupt mask.	0x0
12	TXNOTFULL	TXNOTFULL Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x1
13	IBIWON	IBIWON Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x0
14	-	Reserved.	-
15	ERRWARN	ERRWARN Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x0
18:16	-	Reserved.	-
19	NOWMASTER	NOWCONTROLLER Interrupt Mask 0b - Interrupt not enabled and/or not active 1b - Interrupt enabled and active	0x0
31:20	-	Reserved.	-

21.8.32 Controller Errors and Warnings

Contains errors and warnings from I3C/I2C protocol. When any errors or warnings are not 0, then the MSTATUS[ERRWARN] bit is 1.

Parallel-to-target ERRWARN:

- In Controller mode, use this register (MERRWARN).
- In Target mode, use the Target Errors and Warnings (SERRWARN) register.

The error fields in both registers (MERRWARN and SERRWARN) are similar in meaning.

Table 357. Controller Errors and Warnings register (MERRWARN, offset = 0x9C)

Bit	Symbol	Description	Reset value
0	-	Reserved.	-
1	URUN	Underrun error Indicates an underrun for HDR-BT. This error occurs when attempting to write from an empty transmit FIFO. 0b - No error 1b - Error	
2	NACK	Not Acknowledge Error Indicates an error caused by the target or targets NACKing (not acknowledging) the last address. If 7Eh was the address, then all targets NACKed the last address. If the MCTRL register is written, this field automatically becomes 0. Note: In HDR mode, this error occurs when an address is not accepted (as opposed to NACKed). 0b - No error 1b - Error	0x0
3	WRABT	Write Abort Error Indicates an error caused by the I2C target NACKing the write data, terminating the message. For example, the controller is writing in I2C and the Target NACKED the write. If the MCTRL register is written, this field automatically becomes 0. 0b - No error 1b - Error	0x0
4	TERM	Terminate Error Indicates an error when this controller terminates a target read because the read exceeded the count for the message. This error only valid when using the MWMSG_SDR or MWMSG_DDR register. If the MWMSG_SDR or MWMSG_DDR register is written, this field automatically becomes 0. 0b - No error 1b - Error	0x0
8:5	-	Reserved.	-
9	HPAR	High Data Rate Parity Indicates a parity error from a DDR read, including a bad preamble on a read. Does not stop the read, because it is not safe to terminate; the read data may become mis-framed. Ends on a run of 1 second. 0b - No error 1b - Error	0x0
10	HCRC	High Data Rate CRC Error Indicates that a Cyclic Redundancy Check (CRC) error occurred from a DDR read. 0b - No error 1b - Error	0x0
11	-	Reserved.	-
12	-	Reserved.	
13	-	Reserved.	
14	-	Reserved.	
15	-	Reserved.	

Table 357. Controller Errors and Warnings register (MERRWARN, offset = 0x9C) *…continued*

Bit	Symbol	Description	Reset value
16	OREAD	Over-read Error Indicates an error caused by trying to read from the Controller Read Data Byte (MRDATAB) register when the FIFO is empty. 0b - No error 1b - Error	0x0
17	OWRITE	Over-write Error Indicates an error caused by trying to write to the Controller Write Data Byte Register (MWDATAB) when the FIFO is full. 0b - No error 1b - Error	0x0
18	MSGERR	Message error. Indicates an error caused by: • Trying to write to or read from MWMSG_SDR register while in a DDR message. • Trying to write to or read from MWMSG_DDR register while in a SDR message. • Trying to read from HRMSG_SDR or HRMSG_DDR registers when no message has yet started. 0b - No error 1b - Error	0x0
19	INVREQ	Indicates an error caused by an invalid use of a request: • Not using IBI ACK NACK when stopped in manual hold for IBI acknowledgment. • Using a request other than ForceStop or ForceExit while in a message. Other requests are valid when the message is done. • Other mismatched uses (for example, IBI ACK NACK while in normal states). 0b - No error 1b - Error	0x0
20	TIMEOUT	Indicates an error caused by the module stalling for too long in a frame. This stalling occurs when: • The Transmit FIFO or Receive FIFO is not handled, and the bus is stuck in the middle of a message. • No STOP is issued and between messages. • IBI manual is used and no decision has been made. The maximum stall period is 10 kHz or 100 us. 0b - No error 1b - Error	0x0
21	-	Reserved.	
31:22	-	Reserved.	-

21.8.33 Controller DMA Control

Allows DMA to be used for inbound and outbound messages. DMA is much more useful for a controller than for a target, because the controller is directing the bus traffic and actions. DMA can be used with a controller in these ways:

- Push or pull data to accompany an MCTRL[REQUEST] = 1 (EmitStartAddr) request written by the processor.

- Implementing message mode completely controlled by DMA.

Table 358. Controller DMA Control register (MDMACTRL, offset = 0xA0)

Bit	Symbol	Value	Description	Reset value
1:0	DMAFB		DMA From Bus.	0x0
			Represents the DMA Read (from-bus) trigger. When DMAFB = 1 or DMAFB = 2, the I3C module requests DMA when a receive trigger occurs. See Controller Data Control (MDATACTRL) register. The I3C module requests until empty, unless the DMA is set up as a trigger.	
			DMAFB becomes 0 when MSTATUS[ERRWARN] = 1.	
		00b	DMA is not used	
		01b	Enable DMA for one frame. STOP or repeated START causes DMAFB to automatically become 0. See SCONFIG[MATCHSS] .	
		10b	Enable DMA until DMA is turned off	
		11b	Reserved	
3:2	DMATB		DMA to Bus.	0x0
			Represents the DMA Write (to-bus) trigger. When DMAFB = 1 or DMAFB = 2, the I3C module starts request DMA when a transmit trigger occurs. See Controller Data Control (MDATACTRL) register. The I3C module requests until full, unless the DMA is set up as a trigger.	
			DMAFB becomes 0 when MSTATUS[ERRWARN] = 1.	
		00b	DMA is not used	
		01b	Enable DMA for one frame (ended by DMA or Terminated). STOP or START causes DMATB to become 0. See SCONFIG[MATCHSS] .	
		10b	Enable DMA until DMA is turned off. Normally, DMA ENABLE should only be used in Controller Message mode.	
		11b	Reserved	
5:4	DMAWIDTH		DMA Width.	0x1
			Specifies the data width of DMA operations.	
		00b	Byte	
		10b	Halfword (16 bits). This setting ensures that two bytes are free/available in FIFO.	
		11b	Reserved.	
7:6	-	-	Reserved.	
31:7	-	-	Reserved.	-

21.8.34 Controller Data Control

Assists in data control when there is no FIFO. It also assists the FIFO when FIFO is available (regardless of size). This assistance allows some control over the FIFO behavior. In particular, it provides control over when to interrupt when a particular state of fullness or emptiness is reached. It also controls behavior related to width, when the width is not one-byte wide.

Note: When flushing a FIFO while DMA is in use, disable the DMA channel first. FIFO flush should not be used when a message (in that direction) is in flight.

The Controller Data Control register is an alias of the Target Data Control (SDATACTRL) register.

Table 359. Controller Data Control register (MDATACTRL, offset = 0xAC)

Bit	Symbol	Description	Reset value
0	FLUSHTB	Flush To-bus Buffer or FIFO. Used when the controller terminates a to-bus message (read) prematurely. 0b - No action 1b - Flush the buffer	0x0
1	FLUSHFB	Flush From-bus Buffer or FIFO. FLUSHFB is not normally used. 0b - No action 1b - Flush the buffer	0x0
2	-	Reserved.	-
3	UNLOCK	Unlock. Note: UNLOCK must be 1 in the same cycle while writing to TXTRIG or RXTRIG. 0b - Locked. RXTRIG and TXTRIG fields cannot be changed on a write. 1b - Unlocked. RXTRIG and TXTRIG fields can be changed on a write.	0x0
5:4	TXTRIG	Transmit Trigger Level. Indicates trigger level for Transmit emptiness when using a FIFO. Affects TXNOTFULL interrupt 00b - Trigger when empty 01b - Trigger when 1/4 full or less 10b - Trigger when 1/2 full or less 11b - Default. Trigger when 1 less than full or less	0x3
7:6	RXTRIG	Receive Trigger Level Indicates trigger level for Receive fullness when using a FIFO. Affects RXPEND interrupt. 00b - Trigger when not empty 01b - Trigger when 1/4 full or more 10b - Trigger when 1/2 full or more 11b - Trigger when 3/4 full or more	0x0
15:8	-	Reserved.	-
20:16	TXCOUNT	Transmit Byte Count. Contains the count of bytes waiting in the TXFIFO. This count is the number of bytes the application has written to the Transmit FIFO that have not yet gone onto the I3C bus.	0x0
21	-	Reserved.	
23:22	-	Reserved.	-
28:24	RXCOUNT	Receive Byte Count Contains the count of bytes in the Receive FIFO or buffer.	0x0
29	-	Reserved.	-
30	TXFULL	Transmit Is Full 0b - Transmit FIFO or buffer is not yet full. 1b - Transmit FIFO or buffer is full.	0x0
31	RXEMPTY	Receive Is Empty 0b - Receive FIFO or buffer is not yet empty. 1b - Receive FIFO or buffer is empty.	0x1

21.8.35 Controller Write Data Byte

Allows writing bytes to send onto the bus. Only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

The MWDATAB register is the alias of the Target Write Data Byte (SWDATAB) register.

Table 360. Controller Write Data Byte (MWDATAB, offset = 0xB0)

Bit	Symbol	Description	Reset value
7:0	DATA	Data byte Represents the byte written to the target (stored in Transmit FIFO). • If I3C, the block will compute the parity. • If I2C, the block will handle the ACK/NACK.	-
8	END	End of message. Indicates the end of a message. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs. 0b - Not the end. More bytes are assumed to be in the message. 1b - End. The END bit marks the last byte of the message.	-
15:9	-	Reserved.	-
16	END_ALSO	End of message. Indicates end of message, used to end outbound message normally. Every message must indicate when it is the last message to be sent. This method can be used with the MDATABE register. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END_ALSO must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs. 0b - Not the end. More bytes are assumed to be in the message. 1b - End. The END bit marks the last byte of the message.	-
31:17	-	Reserved.	-

21.8.36 Controller Write Data Byte End

Allows writing the last byte to send onto the bus. Only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively. The MWDATABE register is the alias of the Target Write Data Byte End (SWDATABE) register.

Note: MWDATAB can also indicate END using bit 8.

Table 361. Controller Write Data Byte End register (MWDATABE, offset = 0xB4)

Bit	Symbol	Description	Reset value
7:0	VALUE	Data Represents the last byte written to the target (stored in Transmit FIFO). • If I3C, the block computes the parity. • If I2C, the block manages the ACK or NACK.	-
31:8	-	Reserved.	-

21.8.37 Controller Write Data Halfword

Allows writing a halfword (pair of bytes) to the bus unless an external FIFO is used. Sends the low byte followed by the high byte. An end-of-data (last) marker bit is allowed (or must be 0). A halfword should not be written unless there is room for both, as indicated by the use of Transmit FIFO level trigger or MDATACTRL[TXCOUNT].

The MWDATAH register is the alias of the Target Write Data Half-word (SWDATAH) register.

Table 362. Controller Write Data Halfword (MWDATAH, offset = 0xB8)

Bit	Symbol	Description	Reset value
7:0	DATA0	Data Byte 0	-
15:8	DATA1	Data byte 1. The 2nd byte to send to the master. Represents the second byte that is sent to the target (written to Transmit FIFO).	-
16	END	End of message. Indicates the end of a message. For this register, this field always marks DATA1 as the end. This field is required for I3C and is optional for I2C. For HDR-DDR, the byte with the END must be an even byte (second, fourth, sixth, and so on) because DDR uses byte-pairs. 0b - Not the end. More bytes are assumed to be in the message. 1b - End. The END bit marks the last byte of the message.	-
31:17	-	Reserved.	-

21.8.38 Controller Write Data Halfword End

Allows writing a half word of data, which is the end (the last byte of the half word is the end). Writes the half word (byte pair) just like Controller Write Data Halfword (MWDATAH) but marks the second byte as end-of-data (last byte).

A half-word should not be written unless there is room for both, as indicated by the use of Transmit FIFO level trigger or MDATACTRL[TXCOUNT].

The MWDATAHE register is the alias of the Target Write Data Half-word End (SWDATAHE) register.

Table 363. Controller Write Data Halfword End (MWDATAHE, offset = 0xBC)

Bit	Symbol	Description	Reset value
7:0	DATA0	Data Byte 0 Represents the first byte that is sent to the target (written to Transmit FIFO).	-
15:8	DATA1	Data Byte 1 Represents the second byte that is sent to the target (written to Transmit FIFO).	-
31:16	-	Reserved.	-

21.8.39 Controller Read Data Byte

Allows reading bytes written by the target after an SDR Read, or DAA or DDR. Only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

This register is the alias of the Target Read Data Byte (SRDATAB) register.

Table 364. Controller Read Data Byte (MRDATAB, offset = 0xC0)

Bit	Symbol	Description	Reset value
7:0	VALUE	Value Represents the byte read from the controller (and written by the target).	0x0
31:8	-	Reserved.	-

21.8.40 Controller Read Data Half-word register

Allows reading a halfword (byte pair) written by the target after an SDR Read or DAA or DDR. This register is only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.
A halfword should not be read unless there are at least two bytes of data waiting, as indicated the Receive FIFO level trigger or MDATACTRL[RXCOUNT] .

Table 365. Controller Read Data Half-word register (MRDATAH, offset = 0xC8)

Bit	Symbol	Description	Reset value
7:0	LSB	LSB Represents the first byte read from the controller (and written by the target).	0x0
15:8	MSB	MSB Represents the second byte read from the controller (and written by the target).	0x0
31:16	-	Reserved.	-

21.8.41 Controller Write Byte Data 1(to bus)

Allows writing bytes to send onto the bus, and is intended for DMAs which do not clear the upper bits of the word. It does not have the END bits. This register is only used when using Controller Control (MCTRL) to start the message. If MWMSG_SDR or MWMSG_DDR is used to start a message, that interface must be used exclusively.

Table 366. Controller Write Byte Data 1(to bus) (MWDATAB1, offset = 0xCC)

Bit	Symbol	Description	Reset value
7:0	Value	Value Represents byte to write out. The block computes the parity for I3C, or manages the ACK or NACK for I2C.	0x0
31:8	-	Reserved.	-

21.8.42 Controller Write Message Control in SDR mode

Allows setting up and writing 16-bit words in Single Data Rate (SDR) mode. The MWMSG_SDR_ register is modal and has two modes: control and data.

- For the first write to set up a new message, this register functions as the MWSMSG_SDR_CONTROL register.
- For subsequent writes, this register functions as the MWMSG_SDR_DATA register.
- The control information is not pipelined. Write control information registers for a start and then again only when the data to be sent is done.

When not in the middle of a message, the MWMSG_SDR_CONTROL register is used to start a new message (or emit a STOP). After starting a message, the MWMSG_SDR_DATA register is used until the Length (see LEN field) counts down, or until data with END = 1 is used.

The MWMSG_SDR_CONTROL register is written with the 16-bit control word if currently stopped or after the end of the previous message. If MSTATUS[STATE] = 2 (MSGSDR) and MSTATUS[BETWEEN] = 0, the register (at this offset address) functions as the MWMSG_SDR_DATA register. Otherwise, this register (at this offset address) functions as the MWMSG_SDR_CONTRL register, as long as MSTATUS[STATE] is not in another mode.

The control word contains the byte length (6-bit), the address, the direction, and how it ends (stop, ready for next, continuation with more length). If the command is START and an event (IBI, CR, HJ) occurs, MCTRL[IBIRESP] is used to determine action, and the corresponding interrupt will occur. In that case, the message is restarted.

The MWMSG_SDR_CONTROL and MWMSG_SDR_DATA registers are oriented to DMA operations, but the MWMSG_SDR_CONTROL register can also be written by the processor (instead of using MCTRL and MxDATAB).

Table 367. Controller Write Message Control in SDR mode (MWMSG_SDR_CONTROL, offset = 0xD0)

Bit	Symbol	Value	Description	Reset value
0	DIR		Direction.	-
		0b	Write.	
		1b	Read.	
7:1	ADDR	-	Address Contains address to be written.	-
8	END	-	End of SDR message. Indicates end of SDR message. MSTATUS[COMPLETE] = 1 when done. The end can happen: • Before LEN bytes are read in I3C mode, if a target ends sooner. • Before LEN bytes are written in I2C mode, if a target NACKs. Sets MSTATUS.COMPLETE when done. This can be happen before LEN bytes are read in I3C if a slave ends sooner; or this can happen before LEN bytes are written in I2C if a slave NACKs.	-
		0b	Not the end. SDR message ends waiting for a new SDR message (issues a repeated START for a new message).	
		1b	End. SDR message ends at the STOP.	
9	-	-	Reserved.	-
10	I2C		I2C Specifies if the message is I2C or I3C.	-
		0b	I3C message.	
		1b	I2C message.	
15:11	LEN	-	Length. Indicates the byte length of the message. If LEN = 0, then only END is used (and it will not use the address if Stopped). LEN = 1 must not be used. The minimal LEN size is 2 bytes (LEN = 2).	-
31:16	-	-	Reserved.	-

21.8.43 Controller Write Message Data in SDR mode

Contains the 16-bit word to be written in Single Data Rate (SDR) mode. This register also functions as the MWMSG_SDR_CONTROL register.

Table 368. Controller Write Message Data in SDR mode (MWMSG_SDR_DATA, offset = 0xD0)

Bit	Symbol	Description	Reset value
15:0	DATA16B	Data	0x0
16	-	Reserved	-
31:17	-	Reserved	-

21.8.44 Controller Read Message in SDR mode

Allows reading 16-bit words from a target in SDR message mode. The MRMSG_SDR register is used to read 16-bit words from an active message started with MWMSG_SDR. These words are intended to be read by DMA.

Table 369. Controller Read Message in SDR mode (MRMSG_SDR, offset = 0xD4)

Bit	Symbol	Description	Reset value
15:0	DATA	Data Contains the 16-bit word read from the target. • If the length (LEN) was an odd number, the upper byte is 0. • If the target ends before LEN is finished, the module treats the read as completed. • If the target is not done before LEN is finished and END is not a continuation, the read is terminated.	0x0
31:16	-	Reserved.	-

21.8.45 Controller Write Message in DDR mode: First Control Word

Allows setting up and writing 16-bit words in Double Data Rate (DDR) mode. The register has three modes.

- The first write to set up a new message, functions as the MWSMSG_DDR_CONTROL register.
- For subsequent writes, this register functions as the MWMSG_DDR_DATA register
- The control information is not pipelined. Write control information registers at the start and then again only once the data to be sent is done.

When not in the middle of a message, the MWMSG_DDR_CONTROL register is used to start a new message (or emit a STOP). After starting a message, the MWMSG_DDR_DATA register is used until the Length (see LEN field) counts down or until data with END = 1 is used.

The MWMSG_DDR_CONTROL register is written with the 16-bit Control word if currently stopped or after the end of the previous message. If MSTATUS[STATE] = 4 (MSGDDR) and MSTATUS[BETWEEN] = 0, then the register (at this offset address) functions as the MWMSG_DDR_DATA register. Otherwise, this register (at this offset address) functions as the MWMSG_DDR_CONTROL register, as long as MSTATUS[STATE] is not in another mode.

The main control word contains the 16-bit word length and how it ends (stop, ready for next, continuation with more length). Then the command word contains the command and address for read or write. If the command is START and an event (IBI, CR, HJ) occurs, MCTRL[IBIRESP] is used to determine action, and the corresponding interrupt will occur. In that case, the message is restarted.

Note: The module handles preamble, parity, and CRC.

The MWMSG_DDR_CONTROL, and MWMSG_DDR_DATA registers are oriented to DMA operations, but the MWMSG_DDR_CONTROL register can also be written by the processor (instead of using MCTRL and MxDATAB).

Table 370. Controller Write Message in DDR mode: First Control Word (MWMSG_DDR_CONTROL, offset = 0xD8)

Bit	Symbol	Description	Reset value
15:0	ADDRCMD	Address Command. First Data write after control write with LEN!=0. This is formatted as: • 15:9 Target Address to read or write. • 8 Reserved, should be 0. • 7 1 if Read, 0 if Write. • 6:0 CMD as 7-bit value, always for controller.	-
31:16	-	Reserved.	-

21.8.46 Controller Write Message Data in DDR mode

Contains the 16-bit word to be written in Double Data Rate (DDR) mode. This register also functions as the MWMSG_DDR_CONTROL register.

Table 371. Controller Write Message Data in DDR mode (MWMSG_DDR_DATA, offset = 0xD8)

Bit	Symbol	Value	Description	Reset value
15:0	DATA16B		Data	0x0
16	_	_	Reserved	-
31:17	_	_	Reserved	-

21.8.47 Controller Read Message in DDR mode

Allows reading 16-bit words from a target in Double Data Rate (DDR) message mode from an active message started with MWMSG_DDR. These words are intended to be read by DMA.

Table 372. Controller Read Message in DDR mode (MRMSG_DDR, offset = 0xDC)

Bit	Symbol	Description	Reset value
15:0	DATA	Data Contains the 16-bit word read from a target. The first byte is the LSB, and is in DATA[7:0]. The second byte is the MSB, and is in DATA[15:8]. • If the target ends before the entire length of the message (MWMSG_DDR[LEN]) is read, the module considers the DATA read as completed. In I3C mode, the target can indicate the end of message (the last byte). Otherwise, the controller terminates the message if the message is more than the controller will accept. • If the target has not yet finished sending DATA before the entire length of the message (MWMSG_DDR[LEN]) is read and END is not a continuation, the DATA read will be terminated.	0x0
31:16	-	Reserved.	-

21.8.48 Controller Dynamic Address

Allows an I3C module to write its own Dynamic Address (DA) when the I3C module changes from Controller mode to Target mode.

If this device is the Main Controller (the controller during bus initialization), then this device may use this mechanism to assign itself its DA. When the device hands off control to a secondary controller, it becomes a target itself. This DA must be written before switching to Target mode and must not be changed once in Target mode (it is not clock-safe to do so). It must be written with a valid address value in DADDR if DAVALID = 1.

Note: The Main Controller also uses DEFSLVS CCC to define the target addresses, including itself; this mechanism is how secondary controllers know this address. If the controller is not the Main Controller, then this mechanism should not be used.

Table 373. Controller Dynamic Address (MDYNADDR, offset = 0xE4)

Bit	Symbol	Description	Reset value
0	DAVALID	Dynamic address valid 0b - No valid DA assigned 1b - Valid DA assigned	0x0
7:1	DADDR	Dynamic address Contains the assigned Dynamic Address when DAVALID = 1.	0x0
31:8	-	Reserved.	-

21.8.49 Map Feature Control 0

The SMAPCTRLn registers are named SMAPCTRL0, and so on based on the number of mapped addresses. MAPCTRL0 represents the Primary DA or SA onwards being the Mapped addresses.

The features of the SMAPCTRLn registers depend on configurations. SMAPCTRL0 acts differently, as shown below.

In general, this mechanism is intended to replace the DYNADDR register for all MAP related uses.

When using the Auto-MAP and DASA or AASA, the slot is changed from SA to DA. If the controller then issues RSTDAA, the application must rewrite the static addresses and enable them, as they will be marked disabled.

Table 374. Map Feature Control 0 (SMAPCTRL0, offset = 0x11C)

Bit	Symbol	Description	Reset value
0	ENA	Enable Primary Dynamic Address With MAP enabled. 0b - Disable 1b - Enable	0x0
7:1	DA	Dynamic Address Contains primary DA when ENA = 1. When ENA = 0, static address is used (but not shown here). With MAP enabled.	0x0
10:8	CAUSE	Cause Indicates the cause of the most recent DA assignment, which caused a SSTATUS[DACHG] interrupt. With MAP enabled. 000b - No information. This value occurs when not configured to write DA. 001b - Set using ENTDAA 010b - Set using SETDASA, SETAASA, or SETNEWDA 011b - Cleared using RSTDAA 100b - Auto MAP change happened last. The change may have changed this DA as well (for example, ENTDAA, and SETAASA), but at least one MAP entry automatically changed after. All other values are reserved.	0x0
15:11	_	Reserved	_
16	_	Reserved	_
31:17	_	Reserved	_

21.8.50 Slave Module ID

The BlockID, if enabled, allows software to detect the module and its version information.

Table 375. Slave Module ID register (SID, offset = 0xFFC)

Bit	Symbol	Description	Reset value
31:0	ID	The ID meaning is specific to each use of the I3C module. ID = 0x023630000000.	0xEDCB0000

21.9 Glossary

CCC	Common Command Codes
CDC	Clock-domain-crossing
CRC	Cyclic Redundancy Check
DDR	Double Data Rate
FM	Fast Mode
FM+	Fast Mode Plus
HDR	High Data Rate
HDR-BT	HDR Bulk Transfer
HDR-DDR	High Data Rate - Double Data Rate
HJ	Hot Join
IBI	In-Band interrupt
IRQ	Interrupt
CR	Controller Request
PIO	Programmable I/O
SDR	Single Data Rate

UM11607

Chapter 22: Windowed Watchdog Timer (WWDT)

Rev. 3 — April 2023 **User manual**

22.1 How to read this chapter

The watchdog timer is identical on all LPC86x parts.

22.2 Features

- Internally resets chip if not reloaded during the programmable time-out period.
- Optional windowed operation requires reload to occur between a minimum and maximum time-out period, both programmable.
- Optional warning interrupt can be generated at a programmable time prior to watchdog time-out.
- Programmable 24-bit timer with internal fixed pre-scaler.
- Selectable time period from 1,024 watchdog clocks ($T_{WDCLK} \times 256 \times 4$) to over 67 million watchdog clocks ($T_{WDCLK} \times 2^{24} \times 4$) in increments of 4 watchdog clocks.
- “Safe” watchdog operation. Once enabled, requires a hardware reset or a Watchdog reset to be disabled.
- Incorrect feed sequence causes immediate watchdog event if enabled.
- The watchdog reload value can optionally be protected such that it can only be changed after the “warning interrupt” time is reached.
- Flag to indicate Watchdog reset.
- The Watchdog clock (WDCLK) source is the Low-power oscillator.
- The Watchdog timer can be configured to run in Deep-sleep or Power-down mode.
- Debug mode.

22.3 Basic configuration

The WWDT is configured through the following registers:

- Power to the register interface (WWDT PCLK clock): In the SYSAHBCLKCTRL register, set bit 17 in Table 94.
- Enable the WWDT clock source (the low-power oscillator) in the PDRUNCFG register (Table 118). This is the clock source for the timer base.
- For waking up from a WWDT interrupt, enable the watchdog interrupt for wake-up in the STARTERP1 register (Table 115).

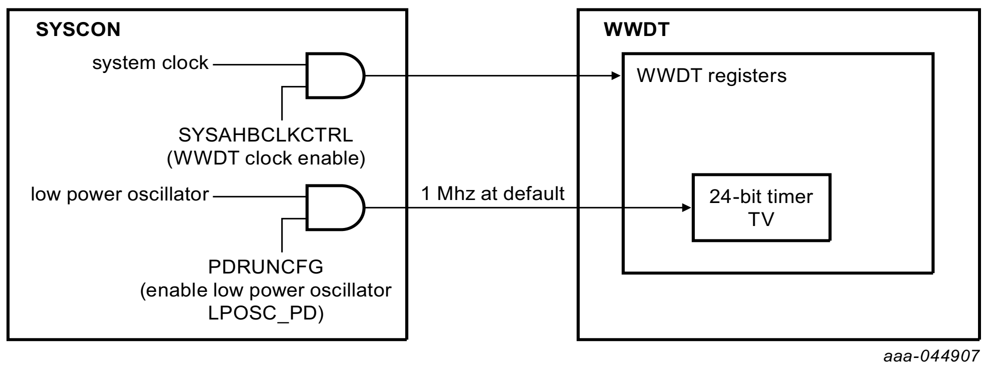


Fig 38. WWDT clocking

22.4 Pin description

The WWDT has no external pins.

22.5 General description

The purpose of the Watchdog Timer is to reset or interrupt the microcontroller within a programmable time if it enters an erroneous state. When enabled, a watchdog reset is generated if the user program fails to feed (reload) the Watchdog within a predetermined amount of time.

When a watchdog window is programmed, an early watchdog feed is also treated as a watchdog event. This allows preventing situations where a system failure may still feed the watchdog. For example, application code could be stuck in an interrupt service that contains a watchdog feed. Setting the window such that this would result in an early feed will generate a watchdog event, allowing for system recovery.

The Watchdog consists of a fixed (divide by 4) pre-scaler and a 24-bit counter which decrements when clocked. The minimum value from which the counter decrements is 0xFF. Setting a value lower than 0xFF causes 0xFF to be loaded in the counter. Hence the minimum Watchdog interval is ($T_{WDCLK} \times 256 \times 4$) and the maximum Watchdog interval is ($T_{WDCLK} \times 2^{24} \times 4$) in multiples of ($T_{WDCLK} \times 4$). The Watchdog should be used in the following manner:

- Set the Watchdog timer constant reload value in the TC register.
- Set the Watchdog timer operating mode in the MOD register.
- Set a value for the watchdog window time in the WINDOW register if windowed operation is desired.
- Set a value for the watchdog warning interrupt in the WARNINT register if a warning interrupt is desired.
- Enable the Watchdog by writing 0xAA followed by 0x55 to the FEED register.
- The Watchdog must be fed again before the Watchdog counter reaches zero in order to prevent a watchdog event. If a window value is programmed, the feed must also occur after the watchdog counter passes that value.

When the Watchdog Timer is configured so that a watchdog event will cause a reset and the counter reaches zero, the CPU will be reset, loading the stack pointer and program counter from the vector table as for an external reset. The Watchdog time-out flag (WDTOF) can be examined to determine if the Watchdog has caused the reset condition. The WDTOF flag must be cleared by software.

When the Watchdog Timer is configured to generate a warning interrupt, the interrupt will occur when the counter matches the value defined by the WARNINT register.

22.5.1 Block diagram

The block diagram of the Watchdog is shown below in the Figure 39. The synchronization logic (PCLK - WDCLK) is not shown in the block diagram.

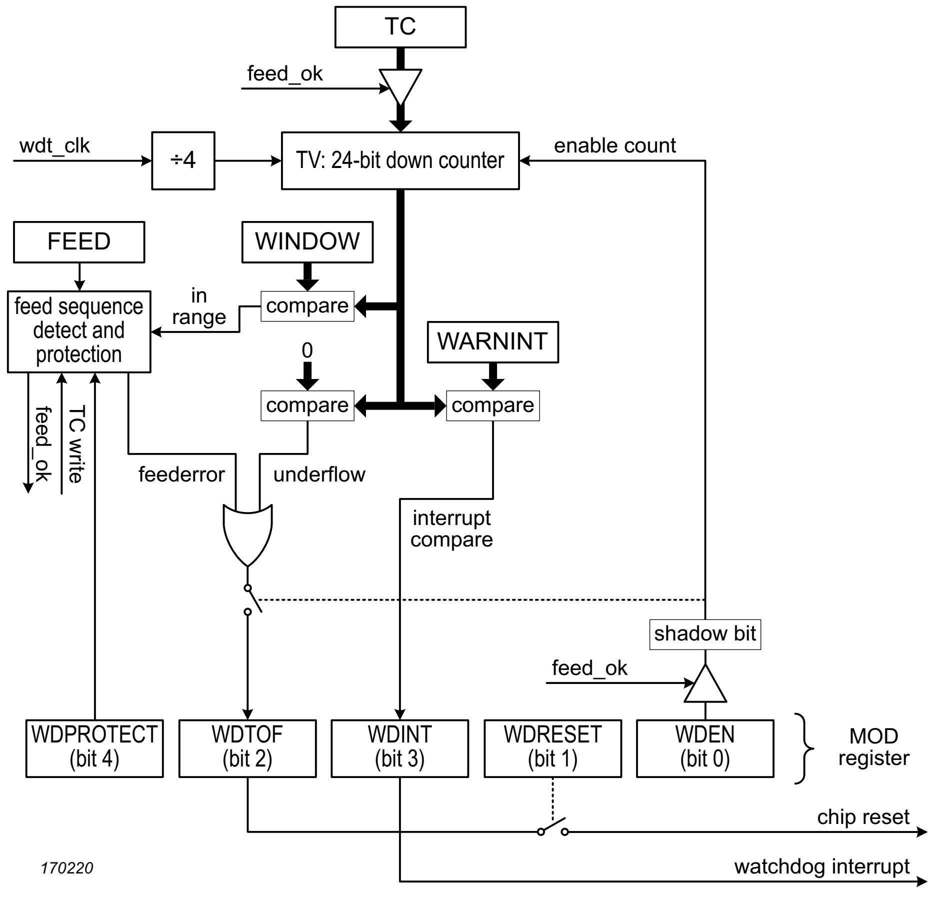


Fig 39. Windowed Watchdog timer block diagram

22.5.2 Clocking and power control

The watchdog timer block uses two clocks: PCLK and WDCLK. PCLK is used for the APB accesses to the watchdog registers and is derived from the system clock (see Figure 7). The WDCLK is used for the watchdog timer counting and is derived from the low-power oscillator.

The synchronization logic between the two clock domains works as follows: When the MOD and TC registers are updated by APB operations, the new value will take effect in 3 WDCLK cycles on the logic in the WDCLK clock domain.

When the watchdog timer is counting on WDCLK, the synchronization logic will first lock the value of the counter on WDCLK and then synchronize it with PCLK, so that the CPU can read the WDTV register.

Remark: Because of the synchronization step, software must add a delay of three WDCLK clock cycles between the feed sequence and the time the WDPROTECT bit is enabled in the MOD register. The length of the delay depends on the selected watchdog clock WDCLK.

22.5.3 Using the WWDT lock features

The WWDT supports several lock features which can be enabled to ensure that the WWDT is running at all times:

- Disabling the WWDT clock source
- Changing the WWDT reload value

22.5.3.1 Disabling the WWDT clock source

If bit 5 in the WWDT MOD register is set, the WWDT clock source is locked and can not be disabled either by software or by hardware when Sleep, Deep-sleep or Power-down modes are entered. Therefore, the user must ensure that the low-power oscillator for each power mode is enabled **before** setting bit 5 in the MOD register.

In Deep power-down mode, no clock locking mechanism is in effect because no clocks are running. However, an additional lock bit in the PMU can be set to prevent the part from even entering Deep power-down mode (see Table 235 “Deep power down control register (DPDCTRL, address 0x4002 0014) bit description”).

22.5.3.2 Changing the WWDT reload value

If bit 4 is set in the WWDT MOD register, the watchdog time-out value (TC) can be changed only after the counter is below the value of WDWARNINT and WDWINDOW.

The reload overwrite lock mechanism can only be disabled by a reset of any type.

22.6 Register description

The Watchdog Timer contains the registers shown in Table 376.

The reset value reflects the data stored in used bits only. It does not include the content of reserved bits.

Table 376. Register overview: Watchdog timer (base address 0x4000 0000)

Name	Access	Address offset	Description	Reset value	Reference
MOD	R/W	0x000	Watchdog mode register. This register contains the basic mode and status of the Watchdog Timer.	0	Table 377
TC	R/W	0x004	Watchdog timer constant register. This 24-bit register determines the time-out value.	0xFF	Table 379
FEED	WO	0x008	Watchdog feed sequence register. Writing 0xAA followed by 0x55 to this register reloads the Watchdog timer with the value contained in WDTC.	NA	Table 380
TV	RO	0x00C	Watchdog timer value register. This 24-bit register reads out the current value of the Watchdog timer.	0xFF	Table 381
-	-	0x010	Reserved	-	-
WARNINT	R/W	0x014	Watchdog Warning Interrupt compare value.	0	Table 382
WINDOW	R/W	0x018	Watchdog Window compare value.	0xFF FFFF	Table 383

22.6.1 Watchdog mode register

The WDMOD register controls the operation of the Watchdog. Note that a watchdog feed must be performed before any changes to the WDMOD register take effect.

Table 377. Watchdog mode register (MOD, 0x4000 0000) bit description

Bit	Symbol	Value	Description	Reset value
0	WDEN		Watchdog enable bit. Once this bit has been written with a 1, it cannot be re-written with a 0. Once this bit is set to one, the watchdog timer starts running after a watchdog feed.	0
		0	The watchdog timer is stopped.	
		1	The watchdog timer is running.	
1	WDRESET		Watchdog reset enable bit. Once this bit has been written with a 1 it cannot be re-written with a 0.	0
		0	A watchdog time-out will not cause a chip reset.	
		1	A watchdog time-out will cause a chip reset.	
2	WDTOF		Watchdog time-out flag. Set when the watchdog timer times out, by a feed error, or by events associated with WDPROTECT. Cleared by writing 0. Causes a chip reset if WDRESET = 1.	0 (only after external reset)

Table 377. Watchdog mode register (MOD, 0x4000 0000) bit description

Bit	Symbol	Value	Description	Reset value
3	WDINT		Warning interrupt flag. Set when the timer reaches the value in WDWARNINT. Cleared by writing 1.	0
4	WDPROTECT		Watchdog update mode. This bit can be set once by software and is only cleared by a reset.	0
		0	The watchdog time-out value (TC) can be changed at any time.	
		1	The watchdog time-out value (TC) can be changed only after the counter is below the value of WDWARNINT and WDWINDOW.	
5	LOCK		A 1 in this bit prevents disabling or powering down the low-power oscillator. This bit can be set once by software and is only cleared by any reset.	0
31:6	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Once the **WDEN**, **WDPROTECT**, or **WDRESET** bits are set they can not be cleared by software. Both flags are cleared by an external reset or a Watchdog timer reset.

WDTOF The Watchdog time-out flag is set when the Watchdog times out, when a feed error occurs, or when PROTECT =1 and an attempt is made to write to the TC register. This flag is cleared by software writing a 0 to this bit.

WDINT The Watchdog interrupt flag is set when the Watchdog counter reaches the value specified by WARNINT. This flag is cleared when any reset occurs, and is cleared by software by writing a 1 to this bit.

In all power modes except Deep power-down mode, a Watchdog reset or interrupt can occur when the watchdog is running and has an operating clock source. The low-power oscillator can be configured to keep running in Sleep, Deep-sleep modes, and Power-down modes.

If a watchdog interrupt occurs in Sleep, Deep-sleep mode, or Power-down mode, and the WWDT interrupt is enabled in the NVIC, the device will wake up. Note that in Deep-sleep and Power-down modes, the WWDT interrupt must be enabled in the STARTERP1 register in addition to the NVIC.

See the following registers:

Table 115 "Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description"

Table 378. Watchdog operating modes selection

WDEN	WDRESET	Mode of Operation
0	X (0 or 1)	Debug/Operate without the Watchdog running.
1	0	Watchdog interrupt mode: the watchdog warning interrupt will be generated but watchdog reset will not. When this mode is selected, the watchdog counter reaching the value specified by WDWARNINT will set the WDINT flag and the Watchdog interrupt request will be generated.
1	1	Watchdog reset mode: both the watchdog interrupt and watchdog reset are enabled. When this mode is selected, the watchdog counter reaching the value specified by WDWARNINT will set the WDINT flag and the Watchdog interrupt request will be generated, and the watchdog counter reaching zero will reset the microcontroller. A watchdog feed prior to reaching the value of WDWINDOW will also cause a watchdog reset.

22.6.2 Watchdog Timer Constant register

The TC register determines the time-out value. Every time a feed sequence occurs the value in the TC is loaded into the Watchdog timer. The TC resets to 0x00 00FF. Writing a value below 0xFF will cause 0x00 00FF to be loaded into the TC. Thus the minimum time-out interval is $T_{WDCLK} \times 256 \times 4$.

If the WDPROTECT bit in WDMOD = 1, an attempt to change the value of TC before the watchdog counter is below the values of WDWARNINT and WDWINDOW will cause a watchdog reset and set the WDTOF flag.

Table 379. Watchdog Timer Constant register (TC, 0x4000 0004) bit description

Bit	Symbol	Description	Reset Value
23:0	COUNT	Watchdog time-out value.	0x00 00FF
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

22.6.3 Watchdog Feed register

Writing 0xAA followed by 0x55 to this register will reload the Watchdog timer with the WDTC value. This operation will also start the Watchdog if it is enabled via the WDMOD register. Setting the WDEN bit in the WDMOD register is not sufficient to enable the Watchdog. A valid feed sequence must be completed after setting WDEN before the Watchdog is capable of generating a reset. Until then, the Watchdog will ignore feed errors.

After writing 0xAA to WDFEED, access to any Watchdog register other than writing 0x55 to WDFEED causes an immediate reset/interrupt when the Watchdog is enabled, and sets the WDTOF flag. The reset will be generated during the second PCLK following an incorrect access to a Watchdog register during a feed sequence.

It is good practice to disable interrupts around a feed sequence, if the application is such that an interrupt might result in rescheduling processor control away from the current task in the middle of the feed, and then lead to some other access to the WDT before control is returned to the interrupted task.

Table 380. Watchdog Feed register (FEED, 0x4000 0008) bit description

Bit	Symbol	Description	Reset Value
7:0	FEED	Feed value should be 0xAA followed by 0x55.	NA
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

22.6.4 Watchdog Timer Value register

The WDTV register is used to read the current value of Watchdog timer counter.

When reading the value of the 24-bit counter, the lock and synchronization procedure takes up to 6 WDCLK cycles plus 6 PCLK cycles, so the value of WDTV is older than the actual value of the timer when it's being read by the CPU.

Table 381. Watchdog Timer Value register (TV, 0x4000 000C) bit description

Bit	Symbol	Description	Reset Value
23:0	COUNT	Counter timer value.	0x00 00FF
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

22.6.5 Watchdog Timer Warning Interrupt register

The WDWARNINT register determines the watchdog timer counter value that will generate a watchdog interrupt. When the watchdog timer counter is less than or equal to the value defined by WARNINT, an interrupt will be generated after the subsequent WDCLK.

A match of the watchdog timer counter to WARNINT occurs when the bottom 10 bits of the counter s less than or equal to the 10 bits of WARNINT, and the remaining upper bits of the counter are all 0. This gives a maximum time of 1,023 watchdog timer counts (4,096 watchdog clocks) for the interrupt to occur prior to a watchdog event. If WARNINT is 0, the interrupt will occur at the same time as the watchdog event.

Table 382. Watchdog Timer Warning Interrupt register (WARNINT, 0x4000 0014) bit description

Bit	Symbol	Description	Reset Value
9:0	WARNINT	Watchdog warning interrupt compare value.	0
31:10	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

22.6.6 Watchdog Timer Window register

The WINDOW register determines the highest WDTV value allowed when a watchdog feed is performed. If a feed sequence occurs when WDTV is greater than the value in WINDOW, a watchdog event will occur.

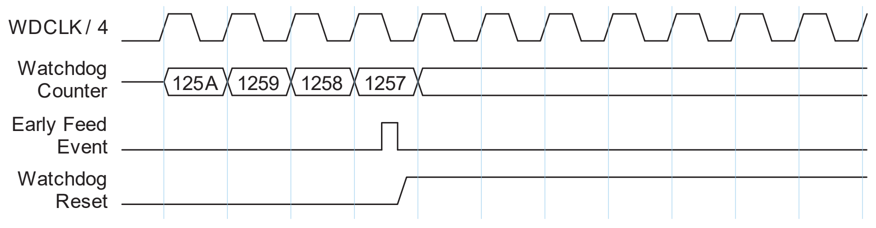
WINDOW resets to the maximum possible WDTV value, so windowing is not in effect.

Table 383. Watchdog Timer Window register (WINDOW, 0x4000 0018) bit description

Bit	Symbol	Description	Reset Value
23:0	WINDOW	Watchdog window value.	0xFF FFFF
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

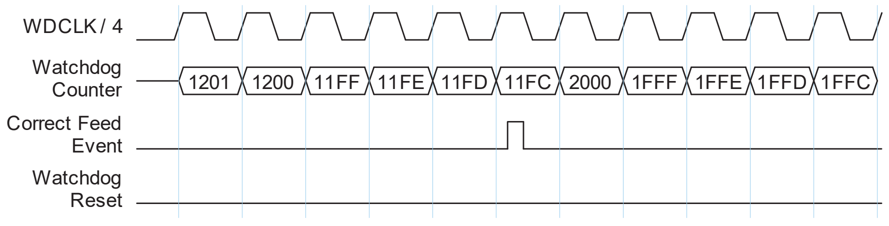
22.7 Functional description

The following figures illustrate several aspects of Watchdog Timer operation.



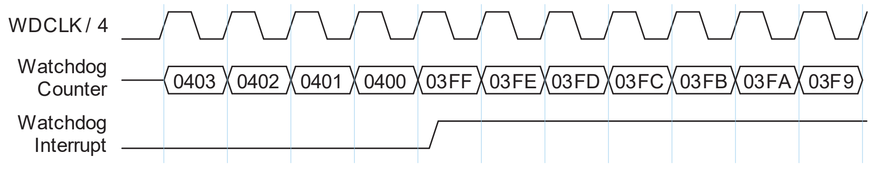
Conditions:
WINDOW = 0x1200
WARNINT = 0x3FF
TC = 0x2000

Fig 40. Early watchdog feed with windowed mode enabled



Conditions:
WDWINDOW = 0x1200
WDWARNINT = 0x3FF
WDTC = 0x2000

Fig 41. Correct watchdog feed with windowed mode enabled



Conditions:
WINDOW = 0x1200
WARNINT = 0x3FF
TC = 0x2000

Fig 42. Watchdog warning interrupt

Chapter 23: Self-wake-up timer (WKT)

23.1 How to read this chapter

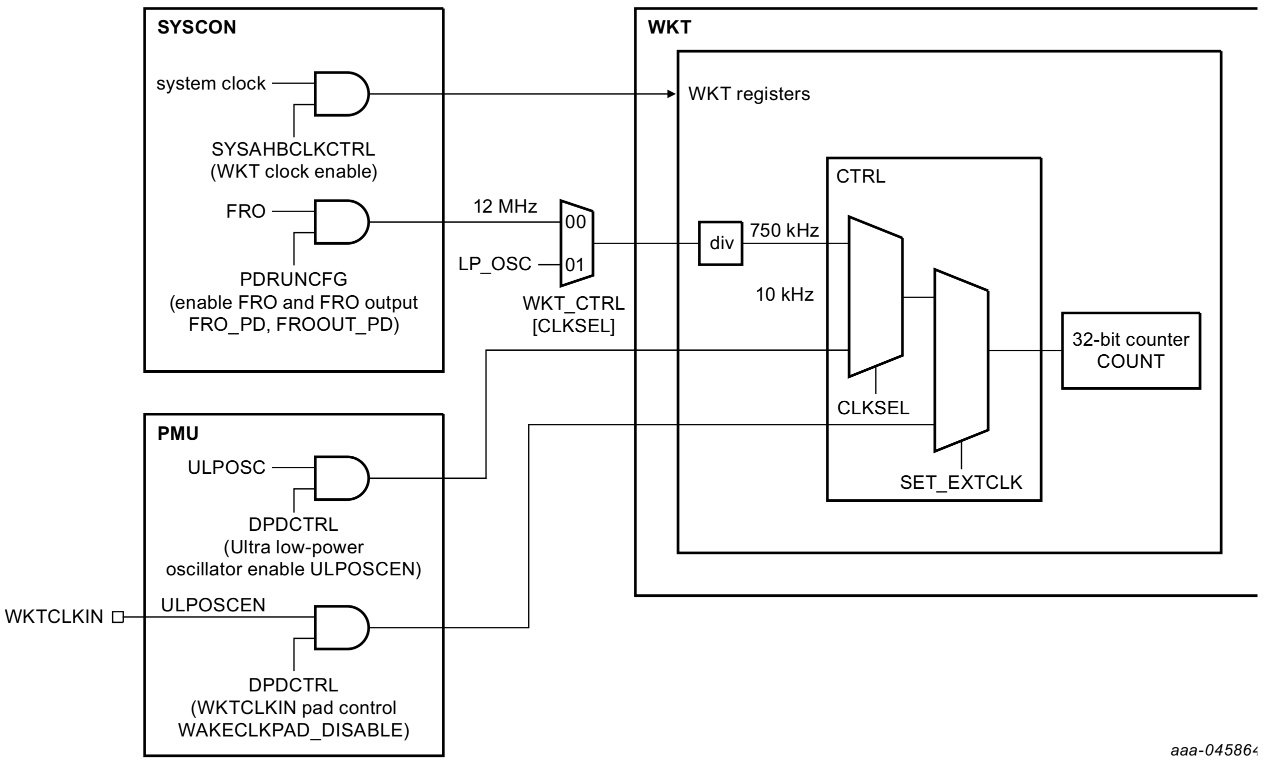
The self-wake-up timer is available on all LPC86x parts.

23.2 Features

- 32-bit, loadable down counter. Counter starts automatically when a count value is loaded. Time-out generates an interrupt/wake up request.
- The WKT resides in a separate, always-on power domain.
- The WKT supports three clock sources: The FRO, the internal ultra low-power oscillator, or the WKTCLKIN pin. The ultra low-power oscillator and the external clock are valid clock sources in all power modes including deep power-down. The FRO can be used in sleep and active mode only.
- Depending on the clock source, the WKT can be used for waking up the part from any low power mode or for general-purpose timing.

23.3 Basic configuration

- In the SYSAHBCLKCTRL register, set bit 9 (Table 94) to enable the clock to the register interface.
- Clear the WKT reset using the PRESETCTRL register (Table 95).
- The WKT interrupt is connected to interrupt #15 in the NVIC. See Table 46.
- Enable the ultra low power oscillator in the PMU (Table 235).
- Enable the FRO and FRO output in the PDRUNCFG register if used as the clock source for the timer (Table 118).
- To use an external clock source for the self-wake-up timer, enable the clock input for pin PIO0_28 in the DPDCTRL register (Table 235) and enable the external clock option in the self-wake-up timer CTRL register (see Table 385). The external clock source can be used in all power modes including deep power-down mode.
- Disable the external clock input in the DPDCTRL register to minimize power consumption if not using the external clock source option. See Table 235.
- Disable the WAKEUP function in the DPDCTRL register to minimize power consumption if the part does not need to wake up from deep power-down mode via a pin. See Table 235.
- See Section 16.7.1 to enable the various power down modes.



Turn off the ultra low-power oscillator in deep power-down mode if not used for wake-up timing.

Fig 43. WKT clocking

23.4 Pin description

The WKT can use a clock input on the external pin PIO0_28 for clocking the wake-up timer in sleep, deep-sleep, power-down, and deep power-down modes. Select the external clock source by setting bit SET_EXTCLK in the CTRL register (see Table 385).

23.5 General description

The self-wake-up timer is a 32-bit, loadable down counter. Writing any non-zero value to this timer automatically enables the counter and launches a count-down sequence. When the counter is being used as a wake up timer, this write can occur just prior to entering a reduced power mode.

When a starting count value is loaded, the self-wake-up timer automatically turns on, counts from the pre-loaded value down to zero, generates an interrupt and/or a wake up request, and then turns itself off until re-launched by a subsequent software write.

23.5.1 WKT clock sources

The self-wake-up timer can be clocked from two alternative clock sources:

- A 750 kHz clock derived from the FRO oscillator. This is the default clock,
- A 10 kHz, low-power clock with a dedicated on-chip oscillator as clock source.
- An external clock on the WKTCLKIN pin.

The FRO-derived clock is much more accurate than the alternative, low-power clock. However, the FRO is not available in most low-power modes. This clock must not be selected when the timer is being used to wake up from a power mode where the FRO is disabled.

The alternative clock source is a (nominally) 10 kHz, low-power clock, sourced from a dedicated oscillator. This oscillator resides in the always-on voltage domain, so it can be programmed to continue operating in Deep power-down mode when power is removed from the rest of the part. This clock is also be available during other low-power modes when the FRO clock is shut-down.

The ultra low-power oscillator is not accurate (approximately +/- 40 % over process and temperature). The frequency may still drift while counting is in progress due to reduced chip temperature after a low-power mode is entered.

An external clock on the WKTCLKIN pin can be used to time the self-wake-up timer in all low power modes, including deep power-down.

23.6 Register description

Table 384. Register overview: WKT (base address 0x4000 8000)

Name	Access	Address offset	Description	Reset value	Reference
CTRL	R/W	0x0	Self-wake-up timer control register.	0	Table 385
COUNT	R/W	0xC	Counter register.	-	Table 386

23.6.1 Control register

The WKT interrupt must be enabled in the NVIC to wake up the part using the self-wake-up counter.

Table 385. Control register (CTRL, address 0x4000 8000) bit description

Bit	Symbol	Value	Description	Reset value
0	CLKSEL		Select the self-wake-up timer clock source. **Remark:** This bit only has an effect if the SEL_EXTCLK bit is not set.	0
		0	Divided FRO clock. This clock runs at 750 kHz and provides time-out periods of up to approximately 95 minutes in 1.33 μs increments. **Remark:** This clock is not available in Deep-sleep, power-down, deep power-down modes. Do not select this option if the timer is to be used to wake up from one of these modes.	
		1	Low power clock. This is the (nominally) 10 kHz clock and provides time-out periods of up to approximately 119 hours in 100 μs increments. The accuracy of this clock is limited to +/- 40 % over temperature and processing. **Remark:** This clock is available in all power modes. Prior to use, the ultra low-power oscillator must be enabled. The oscillator must also be set to remain active in Deep power-down if needed.	

Table 385. Control register (CTRL, address 0x4000 8000) bit description

Bit	Symbol	Value	Description	Reset value
1	ALARMFLAG		Wake-up or alarm timer flag.	-
		0	No time-out. The self-wake-up timer has not timed out. Writing a 0 to has no effect.	
		1	Time-out. The self-wake-up timer has timed out. This flag generates an interrupt request which can wake up the part from any reduced power mode including Deep power-down if the clock source is the ultra low power oscillator. Writing a 1 clears this status bit.	
2	CLEARCTR		Clears the self-wake-up timer.	0
		0	No effect. Reading this bit always returns 0.	
		1	Clear the counter. Counting is halted until a new count value is loaded.	
3	SEL_EXTCLK		Select external or internal clock source for the self-wake-up timer. The internal clock source is selected by the CLKSEL bit in this register if SET_EXTCLK is set to internal.	0
		0	Internal. The clock source is the internal clock selected by the CLKSEL bit.	
		1	External. The self-wake-up timer uses the external WKTCLKIN pin.	
31:4	-		Reserved.	-

23.6.2 Count register

Do not write to this register while the counting is in progress.

Remark: In general, reading the timer state is not recommended. There is no mechanism to ensure that some bits of this register don't change while a read is in progress if the read happens to coincide with an self-wake-up timer clock edge. If you must read this value, it is recommended to read it twice in succession.

Table 386. Counter register (COUNT, address 0x4000 800C) bit description

Bit	Symbol	Description	Reset value
31:0	VALUE	A write to this register pre-loads start count value into the timer and starts the count-down sequence. A read reflects the current value of the timer.	-

Chapter 24: FlexTimer (FTM)

24.1 How to read this chapter

The FlexTimer module is available on all LPC86x devices.

24.2 Introduction

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

The LPC86x device has two FlexTimers, FTM0 and FTM1. The first FlexTimer (FTM0) provides six channels and includes support for motor control (including Fault Control). The second FlexTimer (FTM1) provides four channels. This timer does not include Fault Control but includes a Quadrature Decoder.

Both FlexTimers can be clocked up to 60 MHz. When the FlexTimers are operated at a rate higher than the CPU, they must be exactly 2x the CPU/AHB frequency. Otherwise they can use the same clock as the CPU/AHB.

Both FlexTimers are provided with a selection of hardware triggers, support DMA, and synchronization using the GTB feature. See 7 “Peripheral Input Multiplexing Connections” for details of the FlexTimer triggering and fault inputs.

24.2.1 Modes of operation

When the chip is in an active Debug mode, the FTM temporarily suspends all counting until the chip returns to normal user operating mode. During Stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During Wait mode, the FTM continues to operate normally. If the FTM does not need to produce a real time reference or provide the interrupt sources needed to wake the chip from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

24.2.2 Block diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–7).

Note: The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance. For example, if a module instance supports only six channels, references to channel numbers 6 and 7 do not apply for that instance.

The following figure shows the FTM structure. The central component of the FTM is the 16-bit counter with programmable initial and final values and its counting can be up or up-down.

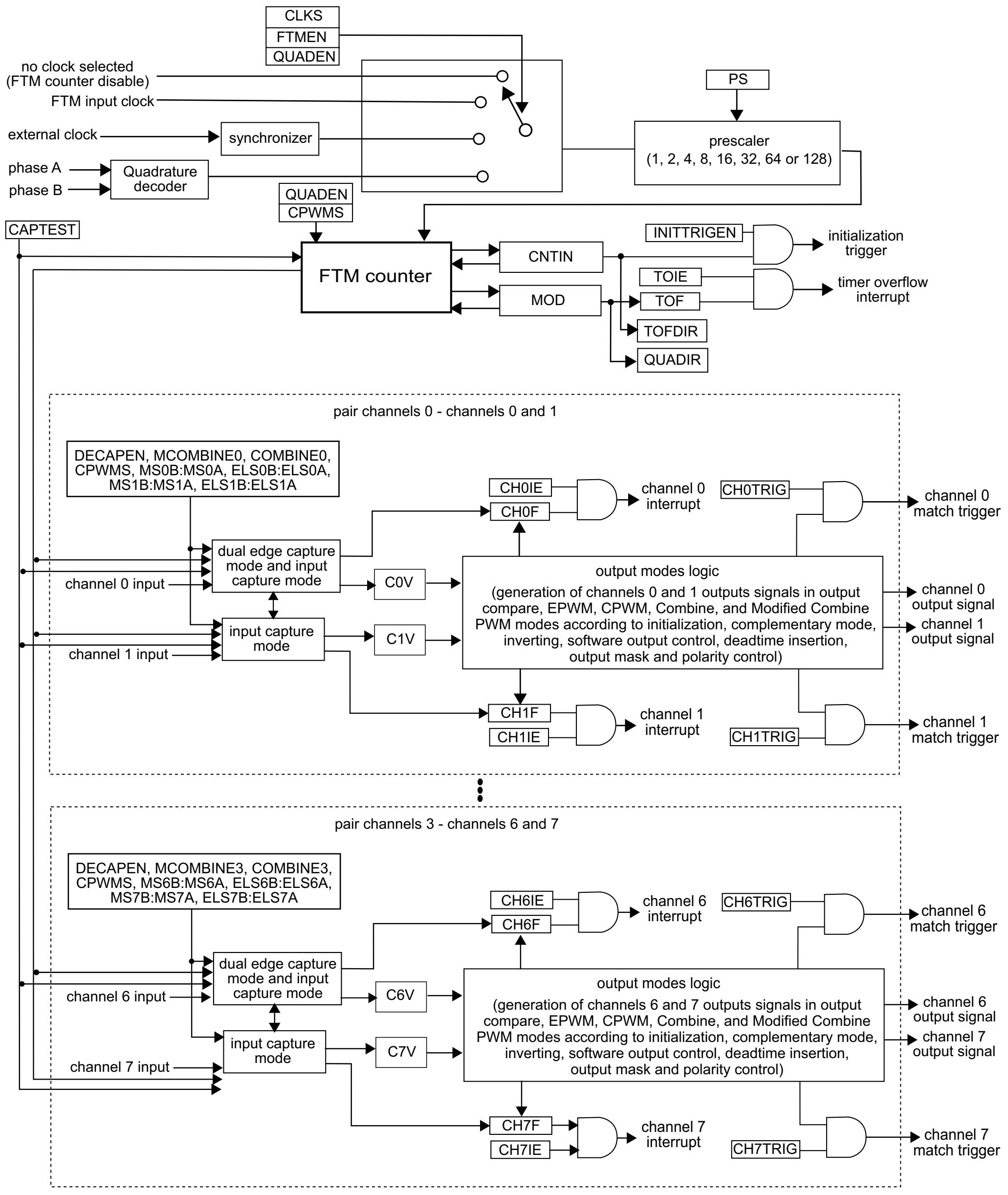

CLKS
FTMEN
QUADEN
no clock selected (FTM counter disable)
FTM input clock
external clock
synchronizer
phase A
phase B
Quadrature decoder
PS
prescaler (1, 2, 4, 8, 16, 32, 64 or 128)
QUADEN
CPWMS
CAPTEST
FTM counter
CNTIN
MOD
INITTRIGEN
initialization trigger
TOIE
TOF
timer overflow interrupt
TOFDIR
QUADIR
pair channels 0 - channels 0 and 1
DECAPEN, MCOMBINE0, COMBINE0, CPWMS, MS0B:MS0A, ELS0B:ELS0A, MS1B:MS1A, ELS1B:ELS1A
CH0IE
CH0F
channel 0 interrupt
CH0TRIG
channel 0 match trigger
dual edge capture mode and input capture mode
channel 0 input
C0V
output modes logic (generation of channels 0 and 1 outputs signals in output compare, EPWM, CPWM, Combine, and Modified Combine PWM modes according to initialization, complementary mode, inverting, software output control, deadtime insertion, output mask and polarity control)
channel 0 output signal
channel 1 output signal
input capture mode
channel 1 input
C1V
CH1F
CH1IE
channel 1 interrupt
CH1TRIG
channel 1 match trigger
pair channels 3 - channels 6 and 7
DECAPEN, MCOMBINE3, COMBINE3, CPWMS, MS6B:MS6A, ELS6B:ELS6A, MS7B:MS7A, ELS7B:ELS7A
CH6IE
CH6F
channel 6 interrupt
CH6TRIG
channel 6 match trigger
dual edge capture mode and input capture mode
channel 6 input
C6V
output modes logic (generation of channels 6 and 7 outputs signals in output compare, EPWM, CPWM, Combine, and Modified Combine PWM modes according to initialization, complementary mode, inverting, software output control, deadtime insertion, output mask and polarity control)
channel 6 output signal
channel 7 output signal
input capture mode
channel 7 input
C7V
CH7F
CH7IE
channel 7 interrupt
CH7TRIG
channel 7 match trigger

Fig 44. FTM block diagram

24.3 Features

- FTM source clock is selectable
  - Source clock can be the FTM input clock (AHB bus clock), the fixed frequency clock, or an external clock.
  - Selecting external clock connects FTM clock to a chip level input pin therefore allowing to synchronize the FTM counter with an off chip clock source
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128

- 16-bit counter
  - It can be a free-running counter or a counter with initial and final value
  - The counting can be up or up-down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In Input Capture mode:
  - The capture can occur on rising edges, falling edges or both edges
  - An input filter can be selected for some channels
- In Output Compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of match triggers
- Software control of PWM outputs
- Up to 4 fault inputs for global fault control
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- The generation of an interrupt when the fault condition is detected
- The generation of an interrupt when a register reload point occurs
- Synchronized loading of write buffered FTM registers
- Half cycle and Full cycle register reload capacity
- Write protection for critical registers
- Backwards compatible with TPM
- Testing of input capture mode
- Direct access to input pin states
- Dual edge capture for pulse and period width measurement
- Quadrature decoder with relative position counting, and interrupt on position count or capture of position count on external event
- The FTM channels can be selected to generate a trigger pulse on channel output instead of a PWM
- Dithering capability to simulate fine edge control for both PWM period or PWM duty cycle

24.4 Basic configuration

Configure the FlexTimers for receiving and transmitting data:

- In the SYSAHBCLKCTRL register, set bit 21 and/or bit 22 (Table 94 “System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description”) to enable the clock to the register interface.
- Clear the FTM peripheral resets using the PRESETCTRL register (Table 95 “Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description”), bit 21 and 22.
- The global FTM interrupts are connected to the interrupts #6 and #7 in the NVIC. See Table 46 “Connection of interrupt sources to the NVIC”.
- Use the switch matrix registers to connect FTMn input signals (FAULT pins, EXTCLK, FTM1_QD_PHA and FTM1_QD_PHB) to external pins. See 11 “Switch matrix (SWM)”.
- Use the switch matrix registers to connect FTMn outputs (FTM0/1_CHn) to external pins. See 11 “Switch matrix (SWM)”.
- The FTMn output trigger signals (init trigger and/or ext trigger) can be connected to the DMA trigger inputs via the DMA_ITRIG_PINMUX registers. See FTM, ADC and DMA interconnection use case figure below, and chapter 7 “Peripheral Input Multiplexing Connections”.
- The FTMn output trigger signals can be connected to the ADC trigger inputs via the ADC input trigger mux Seq_A and Seq_B. See FTM, ADC and DMA interconnection use case figure below and chapter 7 “Peripheral Input Multiplexing Connections”.
- The FTM0 output trigger signals can be connected to the FTM1 trig inputs (or vise versa) to support longer timer period and other complex timing control. See FTM, ADC and DMA interconnection use case figure below, and chapter 7 “Peripheral Input Multiplexing Connections”.
- For detail FTM function description including FTM clock source selection, PWM mode, dead time, initialization trigger, external trigger, PWM synchronization, fault control, polarity control, capture mode, quadrature decoder and etc, See Section 24.8 “Functional description”.

  For detailed FTM initialization, See Section 24.11 “Initialization Procedure”.

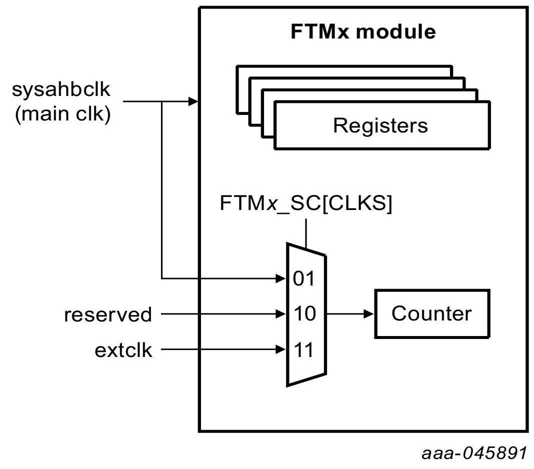


Fig 45. FTM clock diagram

24.5 Pin description

Table 387. Flextimer pin assignments

Function name	Type	Selection 0	Selection 1	Selection 2	Selection 3
FTM0_EXTCLK	I	P0_24	P0_30	-	Not connected
FTM0_CH0	I/O	P0_17	P1_1	-	Not connected
FTM0_CH1	I/O	P0_18	P1_2	P0_16	Not connected
FTM0_CH2	I/O	P0_19	P1_3	P1_2	Not connected
FTM0_CH3	I/O	P0_20	P1_4	P0_27	Not connected
FTM0_CH4	I/O	P0_21	P1_5	P0_25	Not connected
FTM0_CH5	I/O	P0_22	P1_6	P0_24	Not connected
FTM0_FAULT0	I	P0_10	P1_7	P0_28	Not connected
FTM0_FAULT1	I	P0_11	P1_12	P1_3	Not connected
FTM0_FAULT2	I	P0_13	P1_13	-	Not connected
FTM0_FAULT3	I	P0_23	P1_14	-	Not connected
FTM1_EXTCLK	I	P0_25	P0_29	-	Not connected
FTM1_CH0	I/O	P0_15	P1_8	-	Not connected
FTM1_CH1	I/O	P0_16	P1_9	-	Not connected
FTM1_CH2	I/O	P0_26	P0_31	-	Not connected
FTM1_CH3	I/O	P0_27	P1_0	-	Not connected
FTM1_QD_PHA	I	P0_24	P0_29	-	Not connected
FTM1_QD_PHB	I	P0_25	P0_30	-	Not connected

24.6 FTM signal descriptions

Table 388 “FTM signal descriptions” shows the user-accessible signals for the FTM.

Table 388. FTM signal descriptions

Signal	Type	Description	Function
EXTCLK	Input	External clock. FTM external clock can be selected to drive the FTM counter.	The external clock input signal is used as the FTM counter clock if selected by CLKS[1:0] bits in the SC register. This clock signal must not exceed 1/4 of FTM input clock frequency. The FTM counter prescaler selection and settings are also used when an external clock is selected.
CHn	Input/ Output	FTM channel (n), where n can be 7-0	Each FTM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.
FAULTj	Input	Fault input (j), where j can be 3-0	The fault input signals are used to control the CHn channel output state. If a fault is detected, the FAULTj signal is asserted and the channel output is put in a safe state. The behavior of the fault logic is defined by the FAULTM[1:0] control bits in the MODE register and FAULTEN bit in the COMBINE register. Note that each FAULTj input may affect all channels selectively since FAULTM[1:0] and FAULTEN control bits are defined for each pair of channels. Because there are several FAULTj inputs, maximum of 4 for the FTM module, each one of these inputs is activated by the FAULTjEN bit in the FLTCTRL register.
PHA	Input	Quadrature decoder phase A input. Input pin associated with quadrature decoder phase A.	The quadrature decoder phase A input is used as the Quadrature Decoder mode is selected. The phase A input signal is one of the signals that control the FTM counter increment or decrement in the ”Quadrature Decoder Mode”.
PHB	Input	Quadrature decoder phase B input. Input pin associated with quadrature decoder phase B.	The quadrature decoder phase B input is used as the Quadrature Decoder mode is selected. The phase B input signal is one of the signals that control the FTM counter increment or decrement in the ”Quadrature Decoder Mode”.

24.7 Register description

The FTM module contains the registers shown in "Register overview: FTM".

This section presents a high-level summary of the FTM registers and how they are mapped.

The registers and bits of an unavailable function in the FTM remain in the memory map and in the reset value, but they have no active function.

Note: The number of channels supported can vary for each instance of the FTM module on a chip. See the chip-specific FTM information to see how many channels are supported for each module instance.

Accesses to reserved addresses result in transfer errors. Registers for absent channels are considered reserved. Double buffered register writes must be done using 32-bit operations.

Note: For registers in the following table with *Protection*, see the REG_PROT details for more information.

Note:

FTM0 base address: 5000_9000h

FTM1 base address: 5000_A000h

Table 389. Register overview: FTM

Name	Access	Offset	Description	Reset value[1]	Section	Protection
SC	R/W	0x00	"Status And Control (SC)"	0	24.7.1	Yes
CNT	R/W	0x04	"Counter (CNT)"	0	24.7.2	Yes
MOD	R/W	0x08	"Modulo (MOD)"	0	24.7.3	Yes
C0SC	R/W	0x0C	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C0V	R/W	0x10	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C1SC	R/W	0x14	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C1V	R/W	0x18	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C2SC	R/W	0x1C	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C2V	R/W	0x20	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C3SC	R/W	0x24	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C3V	R/W	0x28	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C4SC	R/W	0x2C	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C4V	R/W	0x30	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C5SC	R/W	0x34	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C5V	R/W	0x38	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C6SC	R/W	0x3C	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C6V	R/W	0x40	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes
C7SC	R/W	0x44	"Channel (n) Status And Control (C0SC - C7SC)"	0	24.7.4	Yes
C7V	R/W	0x48	"Channel (n) Value (C0V - C7V)"	0	24.7.5	Yes

Table 389. Register overview: FTM

Name	Access	Offset	Description	Reset value[1]	Section	Protection
CNTIN	R/W	0x4C	"Counter Initial Value (CNTIN)"	0	24.7.6	Yes
STATUS	R/W	0x50	"Capture And Compare Status (STATUS)"	0	24.7.7	No
MODE	R/W	0x54	"Features Mode Selection (MODE)"	0000_0004	24.7.8	No
SYNC	R/W	0x58	"Synchronization (SYNC)"	0	24.7.9	Yes
OUTINIT	R/W	0x5C	"Initial State For Channels Output (OUTINIT)"	0	24.7.10	Yes
OUTMASK	R/W	0x60	"Output Mask (OUTMASK)"	0	24.7.11	Yes
COMBINE	R/W	0x64	"Function For Linked Channels (COMBINE)"	0	24.7.12	Yes
DEADTIME	R/W	0x68	"Deadtime Configuration (DEADTIME)"	0	24.7.13	No
EXTTRIG	R/W	0x6C	"FTM External Trigger (EXTTRIG)"	0	24.7.14	Yes
POL	R/W	0x70	"Channels Polarity (POL)"	0	24.7.15	No
FMS	R/W	0x74	"Fault Mode Status (FMS)"	0	24.7.16	No
FILTER	R/W	0x78	"Input Capture Filter Control (FILTER)"	0	24.7.17	Yes
FLTCTRL	R/W	0x7C	"Fault Control (FLTCTRL)"	0	24.7.18	No
QDCTRL	R/W	0x80	"Quadrature Decoder Control And Status (QDCTRL)"	0	24.7.19	Yes
CONF	R/W	0x84	"Configuration (CONF)"	0	24.7.20	Yes
FLTPOL	R/W	0x88	"FTM Fault Input Polarity (FLTPOL)"	0	24.7.21	No
SYNCONF	R/W	0x8C	"Synchronization Configuration (SYNCONF)"	0	24.7.22	Yes
INVCTRL	R/W	0x90	"FTM Inverting Control (INVCTRL)"	0	24.7.23	Yes
SWOCTRL	R/W	0x94	"FTM Software Output Control (SWOCTRL)"	0	24.7.24	Yes
PWMLOAD	R/W	0x98	"FTM PWM Load (PWMLOAD)"	0	24.7.25	Yes
HCR	R/W	0x9C	"Half Cycle Register (HCR)"	0	24.7.26	No
MOD_MIRROR	R/W	0x200	"Mirror of Modulo Value (MOD_MIRROR)"	0	24.7.27	Yes
C0V_MIRROR - C7V_MIRROR	R/W	0x204-0x220	"Mirror of Channel (n) Match Value (C0V_MIRROR - C7V_MIRROR)"	See Description	24.7.28	Yes

[1] Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

24.7.1 Status And Control (SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, FTM configuration, clock source, and prescaler factor. This register also contains the output enable control bits and the reload opportunity flag control.

These controls relate to all channels within this module.

Table 390. Status And Control Register (SC, offset 0x00) bit description

Bit	Symbol	Description	Reset Value
2:0	PS	Prescale Factor Selection Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next FTM input clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 000b - Divide by 1 001b - Divide by 2 010b - Divide by 4 011b - Divide by 8 100b - Divide by 16 101b - Divide by 32 110b - Divide by 64 111b - Divide by 128	0
4:3	CLKS	Clock Source Selection Selects one of the three FTM counter clock sources. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00b - No clock selected. This in effect disables the FTM counter. 01b - FTM input clock 10b - Reserved 11b - External clock	0
5	CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the FTM to operate in Up-Down Counting mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - FTM counter operates in Up Counting mode. 1b - FTM counter operates in Up-Down Counting mode.	0
6	RIE	Reload Point Interrupt Enable Enables the reload point interrupt. 0b - Reload point interrupt is disabled. 1b - Reload point interrupt is enabled.	0

Table 390. Status And Control Register (SC, offset 0x00) bit description

Bit	Symbol	Description	Reset Value
7	RF	Reload Flag The RF bit is set at each selected reload point. See "Reload Points". The RF bit is cleared by reading the SC register while RF is set and then writing a 0 to RF bit. Writing 1 to RF has no effect. If another selected reload point happens between the read and write operations, the write operation has no effect; therefore, RF remains set. 0b - A selected reload point did not happen. 1b - A selected reload point happened.	0
8	TOIE	Timer Overflow Interrupt Enable Enables FTM overflow interrupts. 0b - Disable TOF interrupts. Use software polling. 1b - Enable TOF interrupts. An interrupt is generated when TOF equals one.	0
9	TOF	Timer Overflow Flag Set by hardware when the FTM counter passes the value in the MOD register. The TOF bit is cleared by reading the SC register while TOF is set and then writing a 0 to TOF bit. Writing a 1 to TOF has no effect. If another FTM overflow occurs between the read and write operations, the write operation has no effect; therefore, TOF remains set indicating an overflow has occurred. In this case, a TOF interrupt request is not lost due to the clearing sequence for a previous TOF. 0b - FTM counter has not overflowed. 1b - FTM counter has overflowed.	0
15:10	Reserved	Reserved	0
16	PWMEN0	Channel 0 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0b - Channel output port is disabled. 1b - Channel output port is enabled.	0
17	PWMEN1	Channel 1 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0b - Channel output port is disabled. 1b - Channel output port is enabled.	0
18	PWMEN2	Channel 2 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0b - Channel output port is disabled. 1b - Channel output port is enabled.	0
19	PWMEN3	Channel 3 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. 0b - Channel output port is disabled. 1b - Channel output port is enabled.	0

Table 390. Status And Control Register (SC, offset 0x00) bit description

Bit	Symbol	Description	Reset Value
20	PWMEN4	Channel 4 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. **Note:** This field is not supported in every instance. 0b - Channel output port is disabled. 1b - Channel output port is enabled.	0
21	PWMEN5	Channel 5 PWM enable bit This bit enables the PWM channel output. This bit should be set to 0 (output disabled) when an input mode is used. **Note:** This field is not supported in every instance. 0b - Channel output port is disabled. 1b - Channel output port is enabled.	0
31:22	Reserved	Reserved. Read value is undefined, only zero should be written.	0

24.7.2 Counter (CNT)

The CNT register contains the FTM counter value.

Reset clears the CNT register. Writing any value to COUNT updates the counter with its initial value, CNTIN.

Table 391. Counter Register (CNT, offset 0x04) bit description

Bit	Symbol	Description	Reset value
15:0	COUNT	Counter Value	0
31:16	Reserved	Reserved	0

24.7.3 Modulo (MOD)

The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock cycle, and the next value of FTM counter depends on the selected counting method; see ”Counter”.

Memory map and register definition

Writes to the MOD register are done on its write buffer. The MOD register is updated with its write buffer value according to ”Registers updated from write buffers”. If FTMEN = 0, a write to SC register resets manually this write coherency mechanism.

Initialize the FTM counter, by writing to CNT, before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Table 392. Modulo Register (MOD, offset 0x08) bit description

Bit	Symbol	Description	Reset value
15:0	MOD	MOD Modulo Value	0
31:16	Reserved	Reserved	0

24.7.4 Channel (n) Status And Control (C0SC - C7SC)

CnSC contains channel (n) status bits and control bits that select the channel (n) mode and its functionality.

Note: Each module instance supports a different number of registers.

Table 393. Channel (n) Status And Control Register ((C0SC - C7SC), offset Ch + (a x 8h)) bit description

Bit	Symbol	Description	Reset value
0	Reserved	Reserved	0
1	ICRST	FTM counter reset by the selected input capture event. FTM counter reset is driven by the selected event of the channel (n) in the Input Capture mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - FTM counter is not reset when the selected channel (n) input event is detected. 1b - FTM counter is reset when the selected channel (n) input event is detected.	0
2	ELSA	Channel (n) Edge or Level Select Used on the selection of the channel (n) mode. See "Channel Modes". This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
3	ELSB	Channel (n) Edge or Level Select Used on the selection of the channel (n) mode. See "Channel Modes". This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
4	MSA	Channel (n) Mode Select Used on the selection of the channel (n) mode. See "Channel Modes". This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
5	MSB	Channel (n) Mode Select Used on the selection of the channel (n) mode. See "Channel Modes". This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
6	CHIE	Channel (n) Interrupt Enable Enables channel (n) interrupt. 0b - Disable channel (n) interrupt. Use software polling. 1b - Enable channel (n) interrupt.	0
7	CHF	Channel (n) Flag Set by hardware when an event occurs on the channel (n). CHF is cleared by reading the CnSC register while CHF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect. If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF. 0b - No channel (n) event has occurred. 1b - A channel (n) event has occurred.	0

Table 393. Channel (n) Status And Control Register ((C0SC - C7SC), offset Ch + (a x 8h)) bit description

Bit	Symbol	Description	Reset value
8	TRIGMOD E	Trigger mode control This bit controls the trigger generation on FTM channel outputs. This mode is allowed only if when FTM channel is configured to EPWM or CPWM modes. If a match in the channel occurs, a trigger pulse with one FTM clock cycle width will be generated in the channel output. See ”Channel trigger output”. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Channel outputs will generate the normal PWM outputs without generating a pulse. 1b - If a match in the channel occurs, a trigger generation on channel output will happen. The trigger pulse width has one FTM clock cycle.	0
9	CHIS	Channel (n) Input State The CHIS bit has the value of the channel (n) input after the double-sampling or the filtering (if the channel (n) filter is enabled) both them are inside the FTM. NOTE: The CHIS bit should be ignored when the channel (n) is not in an input mode. NOTE: When the pair channels is on dual edge mode, the channel (n+1) CHIS bit is the channel (n+1). Input value and not the channel (n) input value (this signal is the input signal used by the dual edge mode). 0b - The channel (n) input is zero. 1b - The channel (n) input is one.	0
31:10	Reserved	Reserved	0

24.7.5 Channel (n) Value (C0V - C7V)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.
In Input Capture , Capture Test, and Dual Edge Capture modes, any write to a CnV register is ignored.

In output modes, writes to the CnV register are done on its write buffer. The CnV register is updated with its write buffer value according to ”Registers updated from write buffers”. If FTMEN = 0, a write to CnSC register resets manually this write coherency mechanism.

Note: Each module instance supports a different number of registers.

Table 394. Channel (n) Value ((C0V - C7V), offset 10h + (a x 8h)) bit description

Bit	Symbol	Description	Reset value
15:0	VAL	Channel Value Captured FTM counter value of the input modes or the match value for the output modes	0
31:16	Reserved	Reserved	0

24.7.6 Counter Initial Value (CNTIN)

The Counter Initial Value register contains the initial value for the FTM counter.

Writing to the CNTIN register latches the value into a buffer. The CNTIN register is updated with the value of its write buffer according to ”Registers updated from write buffers”.

When the FTM clock is initially selected, by writing a non-zero value to the CLKS bits, the FTM counter starts with the value 0x0000. To avoid this behavior, before the first write to select the FTM clock, write the new value to the CNTIN register and then initialize the FTM counter by writing any value to the CNT register.

Table 395. Counter Initial Value (CNTIN, offset 0x04) bit description

Bit	Symbol	Description	Reset value
15:0	INIT	INIT Initial Value Of The FTM Counter	0
31:16	Reserved	Reserved	0

24.7.7 Capture And Compare Status (STATUS)

The STATUS register contains a copy of the status flag CHF bit in CnSC for each FTM channel for software convenience.
Each CHF bit in STATUS is a mirror of CHF bit in CnSC. All CHF bits can be checked using only one read of STATUS. All CHF bits can be cleared by reading STATUS followed by writing 0x00 to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. CHF is cleared by reading STATUS while CHF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect.

If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case, a CHF interrupt request is not lost due to the clearing sequence for a previous CHF.

Table 396. Capture and Compare Status (STATUS, offset 0x50) bit description

Bit	Symbol	Description	Reset value
0	CH0F	Channel 0 Flag See the register description. 0b - No channel event has occurred. 1b - A channel event has occurred	0
1	CH1F	Channel 1 Flag See the register description. 0b - No channel event has occurred. 1b - A channel event has occurred	0
2	CH2F	Channel 2 Flag See the register description. 0b - No channel event has occurred. 1b - A channel event has occurred	0
3	CH3F	Channel 3 Flag See the register description. 0b - No channel event has occurred. 1b - A channel event has occurred	0

Table 396. Capture and Compare Status (STATUS, offset 0x50) bit description

Bit	Symbol	Description	Reset value
4	CH4F	Channel 4 Flag See the register description. NOTE: This field is not supported in every instance. 0b - No channel event has occurred. 1b - A channel event has occurred	0
5	CH5F	Channel 5 Flag See the register description. NOTE: This field is not supported in every instance. The following table includes only supported registers. 0b - No channel event has occurred. 1b - A channel event has occurred	0
6	CH6F	Channel 6 Flag See the register description. NOTE: This field is not supported in every instance. 0b - No channel event has occurred. 1b - A channel event has occurred	0
7	CH7F	Channel 7 Flag See the register description. NOTE: This field is not supported in every instance. 0b - No channel event has occurred. 1b - A channel event has occurred	0
31:8	Reserved	Reserved	0

24.7.8 Features Mode Selection (MODE)

This register contains the global enable bit for FTM-specific features and the control bits used to configure:

- Fault control mode and interrupt
- Capture Test mode
- PWM synchronization
- Write protection
- Channel output initialization

These controls relate to all channels within this module.

Table 397. Features Mode Selection (MODE, offset 0x54) bit description

Bit	Symbol	Description	Reset value
0	FTMEN	FTM Enable This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - TPM compatibility. Free running counter and synchronization compatible with TPM. 1b - Free running counter and synchronization are different from TPM behavior.	0
1	INIT	Initialize The Channels Output When a 1 is written to INIT bit the channels output is initialized according to the state of their corresponding bit in the OUTINIT register. Writing a 0 to INIT bit has no effect. The INIT bit is always read as 0.	0
2	WPDIS	Write Protection Disable When write protection is enabled (WPDIS = 0), write protected bits cannot be written. When write protection is disabled (WPDIS = 1), write protected bits can be written. The WPDIS bit is the negation of the WPEN bit. WPDIS is cleared when 1 is written to WPEN. WPDIS is set when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPDIS has no effect. 0b - Write protection is enabled. 1b - Write protection is disabled.	1
3	PWMSYNC	PWM Synchronization Mode Selects which triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. See ”PWM synchronization”. The PWMSYNC bit configures the synchronization when SYNCMODE is 0. 0b - No restrictions. Software and hardware triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. 1b - Software trigger can only be used by MOD and CnV synchronization, and hardware triggers can only be used by OUTMASK and FTM counter synchronization.	0
4	CAPTEST	Capture Test Mode Enable Enables the capture test mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Capture test mode is disabled. 1b - Capture test mode is enabled.	0

Table 397. Features Mode Selection (MODE, offset 0x54) bit description

Bit	Symbol	Description	Reset value
6:5	FAULTM	Fault Control Mode Defines the FTM fault control mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00b - Fault control is disabled for all channels. 01b - Fault control is enabled for even channels only (channels 0, 2, 4, and 6), and the selected mode is the manual fault clearing. 10b - Fault control is enabled for all channels, and the selected mode is the manual fault clearing. 11b - Fault control is enabled for all channels, and the selected mode is the automatic fault clearing.	0
7	FAULTIE	Fault Interrupt Enable Enables the generation of an interrupt when a fault is detected by FTM and the FTM fault control is enabled. 0b - Fault control interrupt is disabled. 1b - Fault control interrupt is enabled.	0
31:8	-	Reserved	-

24.7.9 Synchronization (SYNC)

This register configures the PWM synchronization.

A synchronization event can perform the synchronized update of MOD, CnV, and OUTMASK registers with the value of their write buffer and the FTM counter initialization.

Note:

The software trigger, SWSYNC bit, and hardware triggers TRIG0, TRIG1, and TRIG2 bits have a potential conflict if used together when SYNCMODE = 0. Use only hardware or software triggers but not both at the same time, otherwise unpredictable behavior is likely to happen.

The selection of the loading point, CNTMAX and CNTMIN bits, is intended to provide the update of MOD, CNTIN, and CnV registers across all enabled channels simultaneously. The use of the loading point selection together with SYNCMODE = 0 and hardware trigger selection, TRIG0, TRIG1, or TRIG2 bits, is likely to result in unpredictable behavior.

The synchronization event selection also depends on the PWMSYNC (MODE register) and SYNCMODE (SYNCONF register) bits. See ”PWM synchronization”.

Table 398. Synchronization (SYNC, offset 0x58) bit description

Bit	Symbol	Description	Reset value
0	CNTMIN	Minimum Loading Point Enable Selects the minimum loading point to PWM synchronization (”Synchronization Points”). If CNTMIN is 1, the selected loading point is when the FTM counter reaches its minimum value (CNTIN register). 0b - The minimum loading point is disabled. 1b - The minimum loading point is enabled.	0
1	CNTMAX	Maximum Loading Point Enable Selects the maximum loading point to PWM synchronization (”Synchronization Points”). If CNTMAX is 1, the selected loading point is when the FTM counter reaches its maximum value (MOD register). 0b - The maximum loading point is disabled. 1b - The maximum loading point is enabled.	0
2	REINIT	FTM Counter Reinitialization by Synchronization Determines if the FTM counter is reinitialized when the selected trigger for the synchronization is detected (”FTM counter synchronization”). The REINIT bit configures the synchronization when SYNCMODE is zero. 0b - FTM counter continues to count normally. 1b - FTM counter is updated with its initial value when the selected trigger is detected.	0
3	SYNCHOM	Output Mask Synchronization Selects when the OUTMASK register is updated with the value of its buffer. 0b - OUTMASK register is updated with the value of its buffer in all rising edges of the FTM input clock. 1b - OUTMASK register is updated with the value of its buffer only by the PWM synchronization.	0
4	TRIG0	PWM Synchronization Hardware Trigger 0 Enables hardware trigger 0 to the PWM synchronization. Hardware trigger 0 occurs when a rising edge is detected at the trigger 0 input signal. 0b - Trigger is disabled. 1b - Trigger is enabled.	0
5	TRIG1	PWM Synchronization Hardware Trigger 1 Enables hardware trigger 1 to the PWM synchronization. Hardware trigger 1 happens when a rising edge is detected at the trigger 1 input signal. 0b - Trigger is disabled. 1b - Trigger is enabled.	0

Table 398. Synchronization (SYNC, offset 0x58) bit description

Bit	Symbol	Description	Reset value
6	TRIG2	PWM Synchronization Hardware Trigger 2 Enables hardware trigger 2 to the PWM synchronization. Hardware trigger 2 happens when a rising edge is detected at the trigger 2 input signal. 0b - Trigger is disabled. 1b - Trigger is enabled.	0
7	SWSYNC	PWM Synchronization Software Trigger Selects the software trigger as the PWM synchronization trigger. The software trigger happens when a 1 is written to SWSYNC bit. 0b - Software trigger is not selected. 1b - Software trigger is selected.	0
31:8	Reserved	Reserved	0

24.7.10 Initial State For Channels Output (OUTINIT)

Table 399. Initial State For Channels Output (OUTINIT, offset 0x5C) bit description

Bit	Symbol	Description	Reset value
0	CH0OI	Channel 0 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
1	CH1OI	Channel 1 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
2	CH2OI	Channel 2 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
3	CH3OI	Channel 3 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
4	CH4OI	Channel 4 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. NOTE: This field is not supported in every instance. 0b - The initialization value is 0. 1b - The initialization value is 1.	0

Table 399. Initial State For Channels Output (OUTINIT, offset 0x5C) bit description

Bit	Symbol	Description	Reset value
5	CH5OI	Channel 5 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. NOTE: This field is not supported in every instance. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
6	CH6OI	Channel 6 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. NOTE: This field is not supported in every instance. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
7	CH7OI	Channel 7 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. NOTE: This field is not supported in every instance. 0b - The initialization value is 0. 1b - The initialization value is 1.	0
31:8	Reserved	Reserved	0

24.7.11 Output Mask (OUTMASK)

This register provides a mask for each FTM channel. The mask of a channel determines if its output responds, that is, it is masked or not, when a match occurs. This feature is used for BLDC control where the PWM signal is presented to an electric motor at specific times to provide electronic commutation.
Any write to the OUTMASK register, stores the value in its write buffer. The register is updated with the value of its write buffer according to ”PWM synchronization”.

Output Mask bits must not be set for trigger mode.

Table 400. Output Mask (OUTMASK, offset 0x60) bit description

Bit	Symbol	Description	Reset value
0	CH0OM	Channel 0 Output Mask Defines if the channel output is masked or unmasked. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
1	CH1OM	Channel 1 Output Mask Defines if the channel output is masked or unmasked. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
2	CH2OM	Channel 2 Output Mask Defines if the channel output is masked or unmasked. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
3	CH3OM	Channel 3 Output Mask Defines if the channel output is masked or unmasked. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
4	CH4OM	Channel 4 Output Mask Defines if the channel output is masked or unmasked. NOTE: This field is not supported in every instance. The following table includes only supported registers. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
5	CH5OM	Channel 5 Output Mask Defines if the channel output is masked or unmasked. NOTE: This field is not supported in every instance. The following table includes only supported registers. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0

Table 400. Output Mask (OUTMASK, offset 0x60) bit description

Bit	Symbol	Description	Reset value
6	CH6OM	Channel 6 Output Mask Defines if the channel output is masked or unmasked. NOTE: This field is not supported in every instance. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
7	CH7OM	Channel 7 Output Mask Defines if the channel output is masked or unmasked. NOTE: This field is not supported in every instance. 0b - Channel output is not masked. It continues to operate normally. 1b - Channel output is masked. It is forced to its inactive state.	0
31:8	Reserved	Reserved	0

24.7.12 Function For Linked Channels (COMBINE)

This register contains the configuration bits for each pair of channels.

Table 401. Function for Linked Channels (COMBINE, offset 0x64) bit description

Bit	Symbol	Description	Reset value
0	COMBINE0	Combine Channels For n = 0 Used on the selection of the combine mode for channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
1	COMP0	Complement Of Channel (n) For n = 0 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - If the channels (n) and (n+1) are in Combine Mode, the channel (n+1) output is the same as the channel (n) output. If the channel (n+1) is in Output Compare Mode, EPWM or CPWM, the channel (n+1) output is independent from channel (n) output. 1b - The channel (n+1) output is the complement of the channel (n) output.	0
2	DECAPEN0	Dual Edge Capture Mode Enable For n = 0 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1.	0

Table 401. Function for Linked Channels (COMBINE, offset 0x64) bit description

Bit	Symbol	Description	Reset value
3	DECAP0	Dual Edge Capture Mode Captures For n = 0 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0b - The dual edge captures are inactive. 1b - The dual edge captures are active.	0
4	DTEN0	Deadtime Enable For n = 0 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The deadtime insertion in this pair of channels is disabled. 1b - The deadtime insertion in this pair of channels is enabled.	0
5	SYNCEN0	Synchronization Enable For n = 0 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0b - The PWM synchronization in this pair of channels is disabled. 1b - The PWM synchronization in this pair of channels is enabled.	0
6	FAULTEN0	Fault Control Enable For n = 0 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The fault control in this pair of channels is disabled. 1b - The fault control in this pair of channels is enabled.	0
7	Reserved	Reserved	0
8	COMBINE1	Combine Channels For n = 2 Used on the selection of the combine mode for channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
9	COMP1	Complement Of Channel (n) For n = 2 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - If the channels (n) and (n+1) are in Combine Mode, the channel (n+1) output is the same as the channel (n) output. If the channel (n+1) is in Output Compare Mode, EPWM or CPWM, the channel (n+1) output is independent from channel (n) output. 1b - The channel (n+1) output is the complement of the channel (n) output.	0
10	DECAPEN1	Dual Edge Capture Mode Enable For n = 2 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1.	0

Table 401. Function for Linked Channels (COMBINE, offset 0x64) bit description

Bit	Symbol	Description	Reset value
11	DECAP1	Dual Edge Capture Mode Captures For n = 2 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if Dual Edge Capture – One-Shot mode is selected and when the capture of channel (n+1) event is made. 0b - The dual edge captures are inactive. 1b - The dual edge captures are active.	0
12	DTEN1	Deadtime Enable For n = 2 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The deadtime insertion in this pair of channels is disabled. 1b - The deadtime insertion in this pair of channels is enabled.	0
13	SYNCEN1	Synchronization Enable For n = 2 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0b - The PWM synchronization in this pair of channels is disabled. 1b - The PWM synchronization in this pair of channels is enabled.	0
14	FAULTEN1	Fault Control Enable For n = 2 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The fault control in this pair of channels is disabled. 1b - The fault control in this pair of channels is enabled.	0
15	Reserved	Reserved	0
16	COMBINE2	Combine Channels For n = 4 Used on the selection of the combine mode for channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance.	0
17	COMP2	Complement Of Channel (n) For n = 4 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - If the channels (n) and (n+1) are in Combine Mode, the channel (n+1) output is the same as the channel (n) output. If the channel (n+1) is in Output Compare Mode, EPWM or CPWM, the channel (n+1) output is independent from channel (n) output. 1b - The channel (n+1) output is the complement of the channel (n) output.	0

Table 401. Function for Linked Channels (COMBINE, offset 0x64) bit description

Bit	Symbol	Description	Reset value
18	DECAPEN2	Dual Edge Capture Mode Enable For n = 4 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance.	0
19	DECAP2	Dual Edge Capture Mode Captures For n = 4 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. NOTE: This field is not supported in every instance.	0
20	DTEN2	Deadtime Enable For n = 4 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - The deadtime insertion in this pair of channels is disabled. 1b - The deadtime insertion in this pair of channels is enabled.	0
21	SYNCEN2	Synchronization Enable For n = 4 Enables PWM synchronization of registers C(n)V and C(n+1)V. NOTE: This field is not supported in every instance. 0b - The PWM synchronization in this pair of channels is disabled. 1b - The PWM synchronization in this pair of channels is enabled.	0
22	FAULTEN2	Fault Control Enable For n = 4 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - The fault control in this pair of channels is disabled. 1b - The fault control in this pair of channels is enabled.	0
23	Reserved	Reserved	0
24	COMBINE3	Combine Channels For n = 6 Used on the selection of the combine mode for channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance.	0

Table 401. Function for Linked Channels (COMBINE, offset 0x64) bit description

Bit	Symbol	Description	Reset value
25	COMP3	Complement Of Channel (n) for n = 6 In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - If the channels (n) and (n+1) are in Combine Mode, the channel (n+1) output is the same as the channel (n) output. If the channel (n+1) is in Output Compare Mode, EPWM or CPWM, the channel (n+1) output is independent from channel (n) output. 1b - The channel (n+1) output is the complement of the channel (n) output.	0
26	DECAPEN3	Dual Edge Capture Mode Enable For n = 6 Enables the Dual Edge Capture mode in the channels (n) and (n+1). See ”Channel Modes”. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance.	0
27	DECAP3	Dual Edge Capture Mode Captures For n = 6 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. NOTE: This field is not supported in every instance. 0b - The dual edge captures are inactive. 1b - The dual edge captures are active.	0
28	DTEN3	Deadtime Enable For n = 6 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - The deadtime insertion in this pair of channels is disabled. 1b - The deadtime insertion in this pair of channels is enabled.	0

Table 401. Function for Linked Channels (COMBINE, offset 0x64) bit description

Bit	Symbol	Description	Reset value
29	SYNCEN3	Synchronization Enable For n = 6 Enables PWM synchronization of registers C(n)V and C(n+1)V. NOTE: This field is not supported in every instance. 0b - The PWM synchronization in this pair of channels is disabled. 1b - The PWM synchronization in this pair of channels is enabled.	0
30	FAULTEN3	Fault Control Enable For n = 6 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - The fault control in this pair of channels is disabled. 1b - The fault control in this pair of channels is enabled.	0
31	Reserved	Reserved	0

24.7.13 Deadtime Configuration (DEADTIME)

This register selects the deadtime prescaler and value for all pair of channels.

Table 402. Deadtime Configuration (DEADTIME, offset 0x68) bit description

Bit	Symbol	Description	Reset value
5:0	DTVAL	Deadtime Value Selects the deadtime value. Deadtime insert value = (DTPS × DTVAL). This field is write protected. It can be written only when MODE[WPDIS] = 1.	0
7:6	DTPS	Deadtime Prescaler Value Selects the division factor of the FTM input clock. This prescaled clock is used by the deadtime counter. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0xb - Divide the FTM input clock by 1. 10b - Divide the FTM input clock by 4. 11b - Divide the FTM input clock by 16.	0
31:8	Reserved	Reserved	0

24.7.14 FTM External Trigger (EXTTRIG)

This register:

- Indicates when the external trigger was generated
- Enables the generation of a trigger when the FTM counter is equal to its initial value
- Selects which channels are used in the generation of the external trigger

Table 403. FTM External Trigger (EXTTRIG, offset 0x06C) bit description

Bit	Symbol	Description	Reset value
0	CH2TRIG	Channel 2 External Trigger Enable Enables the generation of the external trigger when FTM counter = C2V. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
1	CH3TRIG	Channel 3 External Trigger Enable Enables the generation of the external trigger when FTM counter = C3V. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
2	CH4TRIG	Channel 4 External Trigger Enable Enables the generation of the external trigger when FTM counter = C4V. NOTE: This field is not supported in every instance. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
3	CH5TRIG	Channel 5 External Trigger Enable Enables the generation of the external trigger when FTM counter = C5V. NOTE: This field is not supported in every instance. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
4	CH0TRIG	Channel 0 External Trigger Enable Enables the generation of the external trigger when FTM counter = C0V. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
5	CH1TRIG	Channel 1 External Trigger Enable Enables the generation of the external trigger when FTM counter = C1V. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
6	INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. 0b - The generation of initialization trigger is disabled. 1b - The generation of initialization trigger is enabled.	0

Table 403. FTM External Trigger (EXTTRIG, offset 0x06C) bit description

Bit	Symbol	Description	Reset value
7	TRIGF	Channel Trigger Flag Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0b - No channel trigger was generated. 1b - A channel trigger was generated.	0
8	CH6TRIG	Channel 6 External Trigger Enable Enables the generation of the external trigger when FTM counter = C6V. NOTE: This field is not supported in every instance. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
9	CH7TRIG	Channel 7 External Trigger Enable Enables the generation of the external trigger when FTM counter = C7V. NOTE: This field is not supported in every instance. 0b - The generation of this external trigger is disabled. 1b - The generation of this external trigger is enabled.	0
31:10	Reserved	Reserved	0

24.7.15 Channels Polarity (POL)

This register defines the output polarity of the FTM channels.

Note: The channel safe value is the value of its POL bit.The channel safe value is driven on the channel output when the fault control is enabled and a fault condition is detected.

Table 404. Channels Polarity (POL, offset 0x70) bit description

Bit	Symbol	Description	Reset value
0	POL0	Channel 0 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
1	POL1	Channel 1 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
2	POL2	Channel 2 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
3	POL3	Channel 3 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
4	POL4	Channel 4 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
5	POL5	Channel 5 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. The following table includes only supported registers. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0

Table 404. Channels Polarity (POL, offset 0x70) bit description

Bit	Symbol	Description	Reset value
6	POL6	Channel 6 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. The following table includes only supported registers. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
7	POL7	Channel 7 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. NOTE: This field is not supported in every instance. The following table includes only supported registers. 0b - The channel polarity is active high. 1b - The channel polarity is active low.	0
31:8	Reserved	Reserved	0

24.7.16 Fault Mode Status (FMS)

This register contains:

- the write protection enable bit
- the fault detection flags
- the logic OR of the enabled fault inputs

Table 405. Fault Mode Status (FMS, offset 0x74) bit description

Bit	Symbol	Description	Reset value
0	FAULTF0	Fault Detection Flag 0 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF0 by reading the FMS register while FAULTF0 is set and then writing a 0 to FAULTF0 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF0 has no effect. FAULTF0 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF0 remains set after the clearing sequence is completed for the earlier fault condition. 0b - No fault condition was detected at the fault input. 1b - A fault condition was detected at the fault input.	0
1	FAULTF1	Fault Detection Flag 1 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF1 by reading the FMS register while FAULTF1 is set and then writing a 0 to FAULTF1 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF1 has no effect. FAULTF1 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF1 remains set after the clearing sequence is completed for the earlier fault condition. 0b - No fault condition was detected at the fault input. 1b - A fault condition was detected at the fault input.	0
2	FAULTF2	Fault Detection Flag 2 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF2 by reading the FMS register while FAULTF2 is set and then writing a 0 to FAULTF2 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF2 has no effect. FAULTF2 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF2 remains set after the clearing sequence is completed for the earlier fault condition. 0b - No fault condition was detected at the fault input. 1b - A fault condition was detected at the fault input.	0

Table 405. Fault Mode Status (FMS, offset 0x74) bit description

Bit	Symbol	Description	Reset value
3	FAULTF3	Fault Detection Flag 3 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF3 by reading the FMS register while FAULTF3 is set and then writing a 0 to FAULTF3 while there is no existing fault condition at the corresponding fault input. Writing a 1 to FAULTF3 has no effect. FAULTF3 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF3 remains set after the clearing sequence is completed for the earlier fault condition. 0b - No fault condition was detected at the fault input. 1b - A fault condition was detected at the fault input.	0
4	Reserved	Reserved	0
5	FAULTFIN	Fault Inputs Represents the logic OR of the enabled fault inputs after their filter (if their filter is enabled) when fault control is enabled. 0b - The logic OR of the enabled fault inputs is 0. 1b - The logic OR of the enabled fault inputs is 1.	0
6	WPEN	Write Protection Enable The WPEN bit is the negation of the WPDIS bit. WPEN is set when 1 is written to it. WPEN is cleared when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPEN has no effect. 0b - Write protection is disabled. Write protected bits can be written. 1b - Write protection is enabled. Write protected bits cannot be written.	0
7	FAULTF	Fault Detection Flag Represents the logic OR of the FAULTF bit of each enabled fault input. Clear FAULTF by reading the FMS register while FAULTF is set and then writing a 0 to FAULTF while there is no existing fault condition at the enabled fault inputs. Writing a 1 to FAULTF has no effect. If another fault condition is detected in an enabled fault input before the clearing sequence is completed, the sequence is reset so FAULTF remains set after the clearing sequence is completed for the earlier fault condition. FAULTF is also cleared when FAULTF bit of each enabled fault input is cleared. 0b - No fault condition was detected. 1b - A fault condition was detected.	0
31:8	Reserved	Reserved	0

24.7.17 Input Capture Filter Control (FILTER)

This register selects the filter value for the inputs of channels. Channels 4, 5, 6 and 7 do not have an input filter.

Note: Writing to the FILTER register has immediate effect and must be done only when the channels 0, 1, 2, and 3 are not in input modes. Failure to do this could result in a missing valid signal.

Table 406. Input Capture Filter Control (FILTER, offset 0x78) bit description

Bit	Symbol	Description	Reset value
3:0	CH0FVAL	Channel 0 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.	0
7:4	CH1FVAL	Channel1 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.	0
11:8	CH2FVAL	Channel 2 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.	0
15:12	CH3FVAL	Channel 3 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.	0
31:16	Reserved	Reserved	0

24.7.18 Fault Control (FLTCTRL)

This register contains:

- the state of channels output when a fault event happens
- the enable for each fault input
- the filter enable for each fault input
- the filter value for enabled fault inputs and with filter

Table 407. Fault Control (FLTCTRL, offset 0x07C) bit description

Bit	Symbol	Description	Reset value
0	FAULT0EN	Fault Input 0 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input is disabled. 1b - Fault input is enabled.	0
1	FAULT1EN	Fault Input 1 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input is disabled. 1b - Fault input is enabled.	0
2	FAULT2EN	Fault Input 2 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input is disabled. 1b - Fault input is enabled.	0
3	FAULT3EN	Fault Input 3 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input is disabled. 1b - Fault input is enabled.	0
4	FFLTR0EN	Fault Input 0 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input filter is disabled. 1b - Fault input filter is enabled.	0
5	FFLTR1EN	Fault Input 1 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input filter is disabled. 1b - Fault input filter is enabled.	0
6	FFLTR2EN	Fault Input 2 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input filter is disabled. 1b - Fault input filter is enabled.	0

Table 407. Fault Control (FLTCTRL, offset 0x07C) bit description

Bit	Symbol	Description	Reset value
7	FFLTR3EN	Fault Input 3 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Fault input filter is disabled. 1b - Fault input filter is enabled.	0
11:8	FFVAL	Fault Input Filter Selects the filter value for the fault inputs. The fault filter is disabled when the value is zero. NOTE: Writing to this field has immediate effect and must be done only when the fault control or all fault inputs are disabled. Failure to do this could result in a missing fault detection.	0
14:12	Reserved	Reserved	0
15	FSTATE	Fault output state This configuration allows to put the FTM outputs tri-stated when a fault event is ongoing. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - FTM outputs will be placed into safe values when fault events in ongoing (defined by POL bits). 1b - FTM outputs will be tri-stated when fault event is ongoing	0
31:16	Reserved	Reserved	0

24.7.19 Quadrature Decoder Control And Status (QDCTRL)

This register has the control and status bits for the Quadrature Decoder mode.

Table 408. Quadrature Decoder Control And Status (QDCTRL, offset 0x80) bit description

Bit	Symbol	Description	Reset value
0	QUADEN	Quadrature Decoder Mode Enable Enables the Quadrature Decoder mode. In this mode, the phase A and B input signals control the FTM counter direction. The Quadrature Decoder mode has precedence over the other modes. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - Quadrature Decoder mode is disabled. 1b - Quadrature Decoder mode is enabled.	0
1	TOFDIR	Timer Overflow Direction In Quadrature Decoder Mode Indicates if the TOF bit was set on the top or the bottom of counting. 0b - TOF bit was set on the bottom of counting. There was an FTM counter decrement and FTM counter changes from its minimum value (CNTIN register) to its maximum value (MOD register). 1b - TOF bit was set on the top of counting. There was an FTM counter increment and FTM counter changes from its maximum value (MOD register) to its minimum value (CNTIN register).	0
2	QUADIR	FTM Counter Direction In Quadrature Decoder Mode Indicates the counting direction. 0b - Counting direction is decreasing (FTM counter decrement). 1b - Counting direction is increasing (FTM counter increment).	0
3	QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in the Quadrature Decoder mode. 0b - Phase A and phase B encoding mode. 1b - Count and direction encoding mode.	0
4	PHBPOL	Phase B Input Polarity Selects the polarity for the quadrature decoder phase B input. 0b - Normal polarity. Phase B input signal is not inverted before identifying the rising and falling edges of this signal. 1b - Inverted polarity. Phase B input signal is inverted before identifying the rising and falling edges of this signal.	0
5	PHAPOL	Phase A Input Polarity Selects the polarity for the quadrature decoder phase A input. 0b - Normal polarity. Phase A input signal is not inverted before identifying the rising and falling edges of this signal. 1b - Inverted polarity. Phase A input signal is inverted before identifying the rising and falling edges of this signal.	0

Table 408. Quadrature Decoder Control And Status (QDCTRL, offset 0x80) bit description

Bit	Symbol	Description	Reset value
6	PHBFLTREN	Phase B Input Filter Enable Enables the filter for the quadrature decoder phase B input. The filter value for the phase B input is defined by the CH1FVAL field of FILTER. The phase B filter is also disabled when CH1FVAL is zero. NOTE: This field is not supported in every instance. 0b - Phase B input filter is disabled. 1b - Phase B input filter is enabled.	0
7	PHAFLTREN	Phase A Input Filter Enable Enables the filter for the quadrature decoder phase A input. The filter value for the phase A input is defined by the CH0FVAL field of FILTER. The phase A filter is also disabled when CH0FVAL is zero. NOTE: This field is not supported in every instance. 0b - Phase A input filter is disabled. 1b - Phase A input filter is enabled.	0
31:8	Reserved	Reserved	0

24.7.20 Configuration (CONF)

This register selects the frequency of the reload opportunities, the FTM behavior in Debug mode, the use of an external global time base, and the global time base signal generation.

This register also controls if initialization trigger should be generated when a reload point is reached.

Table 409. Configuration (CONF, offset 0x84) bit description

Bit	Symbol	Description	Reset value
4:0	LDFQ	Frequency of the Reload Opportunities The LDFQ[4:0] bits define the number of enabled reload opportunities should happen until an enabled reload opportunity becomes a reload point. See ”Reload Points”. LDFQ = 0: All reload opportunities are reload points. LDFQ = 1: There is a reload point each 2 reload opportunities. LDFQ = 2: There is a reload point each 3 reload opportunities. LDFQ = 3: There is a reload point each 4 reload opportunities. This pattern continues up to a maximum of 32.	0
5	Reserved	Reserved	0
7:6	BDMMODE	Debug Mode Selects the FTM behavior in Debug mode. See ”Debug mode”.	0
8	Reserved	Reserved	0
9	GTBEEN	Global Time Base Enable Configures the FTM to use an external global time base signal that is generated by another FTM. 0b - Use of an external global time base is disabled. 1b - Use of an external global time base is enabled.	0

Table 409. Configuration (CONF, offset 0x84) bit description

Bit	Symbol	Description	Reset value
10	GTBEOUT	Global Time Base Output Enables the global time base signal generation to other FTMs. 0b - A global time base signal generation is disabled. 1b - A global time base signal generation is enabled.	0
11	ITRIGR	Initialization trigger on Reload Point This bit controls whether an initialization trigger is generated when a reload point configured by PWMLOAD register is reached considering the FTM_CONF[LDFQ] settings. 0b - Initialization trigger is generated on counter wrap events. 1b - Initialization trigger is generated when a reload point is reached.	0
31:12	Reserved	Reserved	0

24.7.21 FTM Fault Input Polarity (FLTPOL)

This register defines the fault inputs polarity.

Table 410. FTM Fault Input Polarity (FLTPOL, offset 0x88) bit description

Bit	Symbol	Description	Reset value
0	FLT0POL	Fault Input 0 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The fault input polarity is active high. A 1 at the fault input indicates a fault. 1b - The fault input polarity is active low. A 0 at the fault input indicates a fault.	0
1	FLT1POL	Fault Input 1 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The fault input polarity is active high. A 1 at the fault input indicates a fault. 1b - The fault input polarity is active low. A 0 at the fault input indicates a fault.	0

Table 410. FTM Fault Input Polarity (FLTPOL, offset 0x88) bit description

Bit	Symbol	Description	Reset value
2	FLT2POL	Fault Input 2 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The fault input polarity is active high. A 1 at the fault input indicates a fault. 1b - The fault input polarity is active low. A 0 at the fault input indicates a fault.	0
3	FLT3POL	Fault Input 3 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0b - The fault input polarity is active high. A 1 at the fault input indicates a fault. 1b - The fault input polarity is active low. A 0 at the fault input indicates a fault.	0
31:4	Reserved	Reserved	0

24.7.22 Synchronization Configuration (SYNCONF)

This register selects the PWM synchronization configuration, SWOCTRL, INVCTRL and CNTIN registers synchronization, if FTM clears the TRIGj bit, where j = 0, 1, 2, when the hardware trigger j is detected.

Table 411. Synchronization Configuration (SYNCONF, offset 0x08C) bit description

Bit	Symbol	Description	Reset value
0	HWTRIGMODE	Hardware Trigger Mode 0b - FTM clears the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2. 1b - FTM does not clear the TRIGj bit when the hardware trigger j is detected, where j = 0, 1,2.	0
1	Reserved	Reserved	0
2	CNTINC	CNTIN Register Synchronization 0b - CNTIN register is updated with its buffer value at all rising edges of FTM input clock. 1b - CNTIN register is updated with its buffer value by the PWM synchronization.	0
3	Reserved	Reserved	0
4	INVC	INVCTRL Register Synchronization 0b - INVCTRL register is updated with its buffer value at all rising edges of FTM input clock. 1b - INVCTRL register is updated with its buffer value by the PWM synchronization.	0

Table 411. Synchronization Configuration (SYNCONF, offset 0x08C) bit description

Bit	Symbol	Description	Reset value
5	SWOC	SWOCTRL Register Synchronization 0b - SWOCTRL register is updated with its buffer value at all rising edges of FTM input clock. 1b - SWOCTRL register is updated with its buffer value by the PWM synchronization.	0
6	Reserved	Reserved	0
7	SYNCMODE	Synchronization Mode Selects the PWM Synchronization mode. 0b - Legacy PWM synchronization is selected. 1b - Enhanced PWM synchronization is selected.	0
8	SWRSTCNT	FTM counter synchronization is activated by the software trigger 0b - The software trigger does not activate the FTM counter synchronization. 1b - The software trigger activates the FTM counter synchronization.	0
9	SWWRBUF	MOD, HCR, CNTIN, and CV registers synchronization is activated by the software trigger 0b - The software trigger does not activate MOD, HCR, CNTIN, and CV registers synchronization. 1b - The software trigger activates MOD, HCR, CNTIN, and CV registers synchronization.	0
10	SWOM	Output mask synchronization is activated by the software trigger 0b - The software trigger does not activate the OUTMASK register synchronization. 1b - The software trigger activates the OUTMASK register synchronization.	0
11	SWINC	Inverting control synchronization is activated by the software trigger 0b - The software trigger does not activate the INVCTRL register synchronization. 1b - The software trigger activates the INVCTRL register synchronization.	0
12	SWSOC	Software output control synchronization is activated by the software trigger 0b - The software trigger does not activate the SWOCTRL register synchronization. 1b - The software trigger activates the SWOCTRL register synchronization.	0
15:13	Reserved	Reserved	0
16	HWRSTCNT	FTM counter synchronization is activated by a hardware trigger 0b - A hardware trigger does not activate the FTM counter synchronization. 1b - A hardware trigger activates the FTM counter synchronization.	0

Table 411. Synchronization Configuration (SYNCONF, offset 0x08C) bit description

Bit	Symbol	Description	Reset value
17	HWWRBUF	MOD, HCR, CNTIN, and CV registers synchronization is activated by a hardware trigger 0b - A hardware trigger does not activate MOD, HCR, CNTIN, and CV registers synchronization. 1b - A hardware trigger activates MOD, HCR, CNTIN, and CV registers synchronization.	0
18	HWOM	Output mask synchronization is activated by a hardware trigger 0b - A hardware trigger does not activate the OUTMASK register synchronization. 1b - A hardware trigger activates the OUTMASK register synchronization.	0
19	HWINVC	Inverting control synchronization is activated by a hardware trigger 0b - A hardware trigger does not activate the INVCTRL register synchronization. 1b - A hardware trigger activates the INVCTRL register synchronization.	0
20	HWSOC	Software output control synchronization is activated by a hardware trigger 0b - A hardware trigger does not activate the SWOCTRL register synchronization. 1b - A hardware trigger activates the SWOCTRL register synchronization.	0
31:21	Reserved	Reserved	0

24.7.23 FTM Inverting Control (INVCTRL)

This register controls when the channel (n) output becomes the channel (n+1) output, and channel (n+1) output becomes the channel (n) output. Each INVmEN bit enables the inverting operation for the corresponding pair channels m.

This register has a write buffer. The INVmEN bit is updated by the INVCTRL register synchronization.

Table 412. FTM Inverting Control (INVCTRL, offset 0x90) bit description

Bit	Symbol	Description	Reset value
0	INV0EN	Pair Channels 0 Inverting Enable 0b - Inverting is disabled. 1b - Inverting is enabled.	0
1	INV1EN	Pair Channels 1 Inverting Enable 0b - Inverting is disabled. 1b - Inverting is enabled.	0

Table 412. FTM Inverting Control (INVCTRL, offset 0x90) bit description

Bit	Symbol	Description	Reset value
2	INV2EN	Pair Channels 2 Inverting Enable NOTE: This field is not supported in every instance. 0b - Inverting is disabled. 1b - Inverting is enabled.	0
3	INV3EN	Pair Channels 3 Inverting Enable NOTE: This field is not supported in every instance. 0b - Inverting is disabled. 1b - Inverting is enabled.	0
31:4	Reserved	Reserved	0

24.7.24 FTM Software Output Control (SWOCTRL)

This register enables software control of channel (n) output and defines the value forced to the channel (n) output:

- The CH(n)OC bits enable the control of the corresponding channel (n) output by software.
- The CH(n)OCV bits select the value that is forced at the corresponding channel (n) output.

This register has a write buffer. The fields are updated by the SWOCTRL register synchronization.

Table 413. FTM Software Output Control (SWOCTRL, offset 0x94) bit description

Bit	Symbol	Description	Reset value
0	CH0OC	Channel 0 Software Output Control Enable 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
1	CH1OC	Channel 1 Software Output Control Enable 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
2	CH2OC	Channel 2 Software Output Control Enable 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
3	CH3OC	Channel 3 Software Output Control Enable 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
4	CH4OC	Channel 4 Software Output Control Enable NOTE: This field is not supported in every instance. 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0

Table 413. FTM Software Output Control (SWOCTRL, offset 0x94) bit description

Bit	Symbol	Description	Reset value
5	CH5OC	Channel 5 Software Output Control Enable NOTE: This field is not supported in every instance. 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
6	CH6OC	Channel 6 Software Output Control Enable NOTE: This field is not supported in every instance. 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
7	CH7OC	Channel 7 Software Output Control Enable NOTE: This field is not supported in every instance. 0b - The channel output is not affected by software output control. 1b - The channel output is affected by software output control.	0
8	CH0OCV	Channel 0 Software Output Control Value 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
9	CH1OCV	Channel 1 Software Output Control Value 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
10	CH2OCV	Channel 2 Software Output Control Value 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
11	CH3OCV	Channel 3 Software Output Control Value 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
12	CH4OCV	Channel 4 Software Output Control Value NOTE: This field is not supported in every instance. 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
13	CH5OCV	Channel 5 Software Output Control Value NOTE: This field is not supported in every instance. 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
14	CH6OCV	Channel 6 Software Output Control Value NOTE: This field is not supported in every instance. 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
15	CH7OCV	Channel 7 Software Output Control Value NOTE: This field is not supported in every instance. 0b - The software output control forces 0 to the channel output. 1b - The software output control forces 1 to the channel output.	0
31:16	Reserved	Reserved	0

24.7.25 FTM PWM Load (PWMLOAD)

Enables the reload of the MOD, HCR, CNTIN, C(n)V, and C(n+1)V registers with the values of their write buffers when the FTM counter changes from the MOD register value to its next value or when a channel (j) match occurs. A match occurs for channel (j) when FTM counter = C(j)V. A reload can also occurs when FTM counter = HCR register at a half cycle match. This register also controls the local and global load mechanisms.

Table 414. FTM PWM Load (PWMLOAD, offset 0x98) bit description

Bit	Symbol	Description	Reset value
0	CH0SEL	Channel 0 Select 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
1	CH1SEL	Channel 1 Select 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
2	CH2SEL	Channel 2 Select 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
3	CH3SEL	Channel 3 Select 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
4	CH4SEL	Channel 4 Select NOTE: This field is not supported in every instance. 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
5	CH5SEL	Channel 5 Select NOTE: This field is not supported in every instance. 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
6	CH6SEL	Channel 6 Select NOTE: This field is not supported in every instance. 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
7	CH7SEL	Channel 7 Select NOTE: This field is not supported in every instance. 0b - Channel match is not included as a reload opportunity. 1b - Channel match is included as a reload opportunity.	0
8	HCSEL	Half Cycle Select This bit enables the half cycle match as a reload opportunity. A half cycle is defined by when the FTM counter matches the HCR register. 0b - Half cycle reload is disabled and it is not considered as a reload opportunity. 1b - Half cycle reload is enabled and it is considered as a reload opportunity.	0

Table 414. FTM PWM Load (PWMLOAD, offset 0x98) bit description

Bit	Symbol	Description	Reset value
9	LDOK	Load Enable Enables the loading of the MOD, CNTIN, HCR and CV registers with the values of their buffers. The LDOK bit can also be set by the Global Load mechanism if GLEN bit is enabled. 0b - Loading updated values is disabled. 1b - Loading updated values is enabled.	0
10	GLEN	Global Load Enable This bit enables the global load mechanism implemented by GLDOK. If GLEN bit is set, then an external event on the FTM global load input sets the LDOK bit. The clear of the LDOK bit is done by CPU writes '0' to the bit. 0b - Global Load OK disabled. 1b - Global Load OK enabled. A pulse event on the module global load input sets the LDOK bit.	0
11	GLDOK	Global Load OK This bit controls the global load mechanism. It generates a pulse at FTM module global load output with one FTM clock cycle width, which is used to set LDOK bits of FTM and other modules (including other FTMs). This bit is self-cleared and read value is always zero. The global load mechanism depends on SoC specific information. Refer to FTM SoC specific information to more details. 0b - No action. 1b - LDOK bit is set.	0
31:12	Reserved	Reserved	0

24.7.26 Half Cycle Register (HCR)

The Half Cycle Register contains the match value for FTM half cycle reload feature. After FTM counter reaches this value, a reload opportunity is generated if FTM_PWMLOAD[HCSEL] is enabled.

Writing to the HCR register latches the value into a buffer. The HCR register is updated with the value of its write buffer according to ”Registers updated from write buffers”.

Table 415. Half Cycle Register (HCR, offset 0x09C) bit description

Bit	Symbol	Description	Reset value
15:0	HCVAL	Half Cycle Value	0
31:16	-	Reserved	-

24.7.27 Mirror of Modulo Value (MOD_MIRROR)

This register contains the integer and fractional modulo value for the FTM counter.

Table 416. Mirror of Modulo Value (MOD_MIRROR, offset 0x200) bit description

Bit	Symbol	Description	Reset value
10:0	-	Reserved	-
15:11	FRACMOD	Modulo Fractional Value The modulo fractional value is used in the PWM period dithering. This value is added to an internal accumulator at the end of each PWM period. Writes to the field FRACMOD are done on its write buffer. The FRACMOD is updated with its write buffer value according to ”Registers updated from write buffers”. If FTMEN = 0, a write to SC register resets manually this write coherency mechanism.	0
31:16	MOD	Mirror of the Modulo Integer Value See the field MOD of the register MOD.	0

24.7.28 Mirror of Channel (n) Match Value (C0V_MIRROR - C7V_MIRROR)

This register contains the integer and fractional value of the channel (n) match.

Note: Each module instance supports a different number of registers.

Table 417. (CaV_MIRROR, offset 204h + (a x 4h)) bit description

Bit	Symbol	Description	Reset value
10:0	-	Reserved	-
15:11	FRACVAL	Channel (n) Match Fractional Value The channel (n) match fractional value is used in the PWM edge dithering. This value is added to the channel (n) internal accumulator at the end of each PWM period. Writes to the field FRACVAL are done on its write buffer. The FRACVAL is updated with its write buffer value according to ”Registers updated from write buffers”. If FTMEN = 0, a write to CnSC register resets manually this write coherency mechanism.	0
31:16	VAL	Mirror of the Channel (n) Match Integer Value See the field VAL of the register CnV.	0

24.8 Functional description

24.8.1 Clock source

The FTM has only one clock domain: the FTM input clock.

24.8.1.1 Counter clock source

The CLKS[1:0] bits select one of three possible clock sources for the FTM counter or disable the FTM counter. After any chip reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The external clock passes through a synchronizer clocked by the FTM input clock to assure that counter transitions are properly aligned to FTM input clock transitions.Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must not exceed 1/4 of the FTM input clock frequency.

24.8.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter. The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and FTM counter.

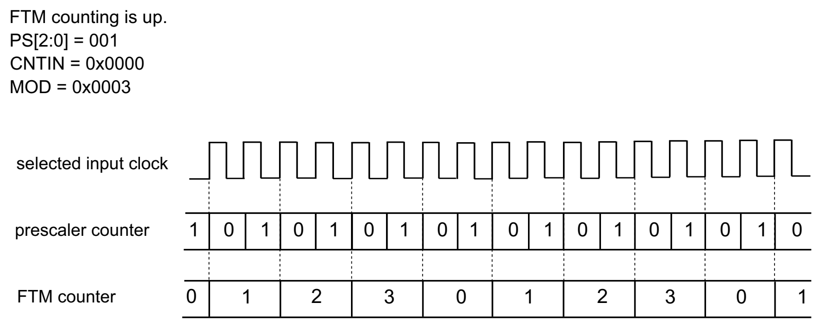


Fig 46. Example of the prescaler counter

24.8.3 Counter

The FTM has a 16-bit counter that is used by the channels either for input or output modes. The FTM counter clock is the selected clock divided by the prescaler.

The FTM counter has these modes of operation:

- ”Up counting”
- ”Up-down counting”
- ”Quadrature Decoder Mode”

24.8.3.1 Up counting

Up counting is selected when:

- QUADEN = 0, and
- CPWMS = 0

CNTIN defines the starting value of the count and MOD defines the final value of the count, see the following figure. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with the value of CNTIN.

The FTM period when using up counting is (MOD – CNTIN + 0x0001) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to CNTIN.

A counter event happens at the same time of TOF bit set when the FTM counter changes from MOD to CNTIN. See "Counter events" for more details.

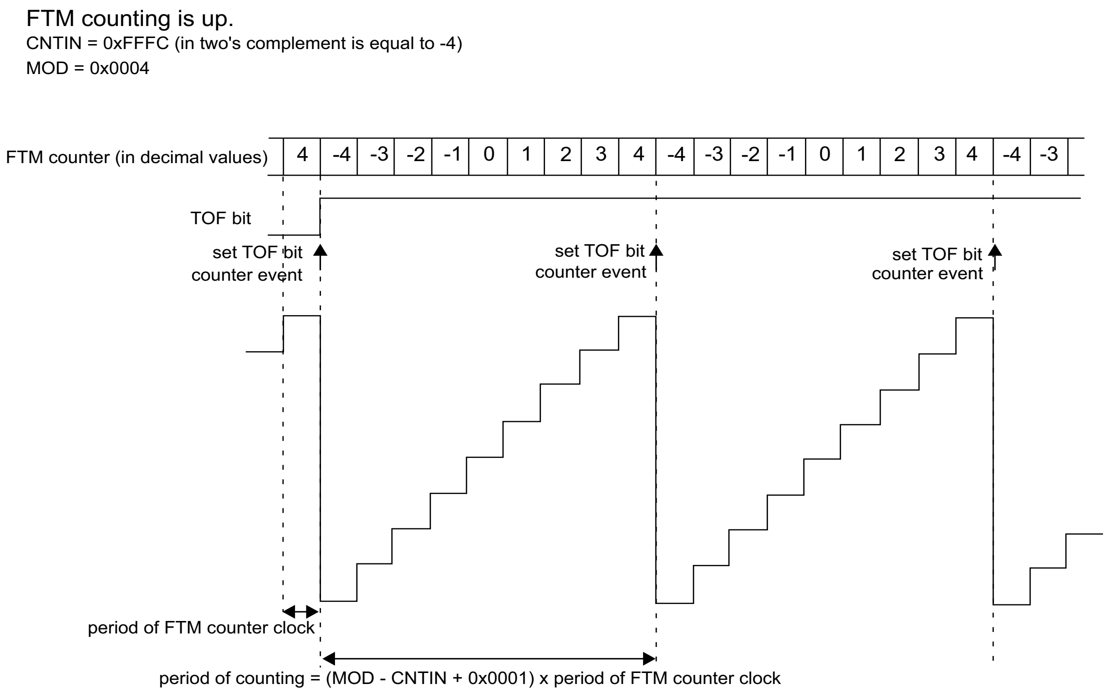

FTM counting is up.
CNTIN = 0xFFFC (in two's complement is equal to -4)
MOD = 0x0004
FTM counter (in decimal values)
4 -4 -3 -2 -1 0 1 2 3 4 -4 -3 -2 -1 0 1 2 3 4 -4 -3
TOF bit
set TOF bit
counter event
set TOF bit
counter event
set TOF bit
counter event
period of FTM counter clock
period of counting = (MOD - CNTIN + 0x0001) x period of FTM counter clock

Fig 47. Example of FTM up and signed counting

Table 418. FTM counting based on CNTIN value

When	Then
CNTIN = 0x0000	The FTM counting is equivalent to TPM up counting, that is, up and unsigned counting. See the following figure.
CNTIN[15] = 1	The initial value of the FTM counter is a negative number in two's complement, so the FTM counting is up and signed.
CNTIN[15] = 0 and CNTIN ≠ 0x0000	The initial value of the FTM counter is a positive number, so the FTM counting is up and unsigned.

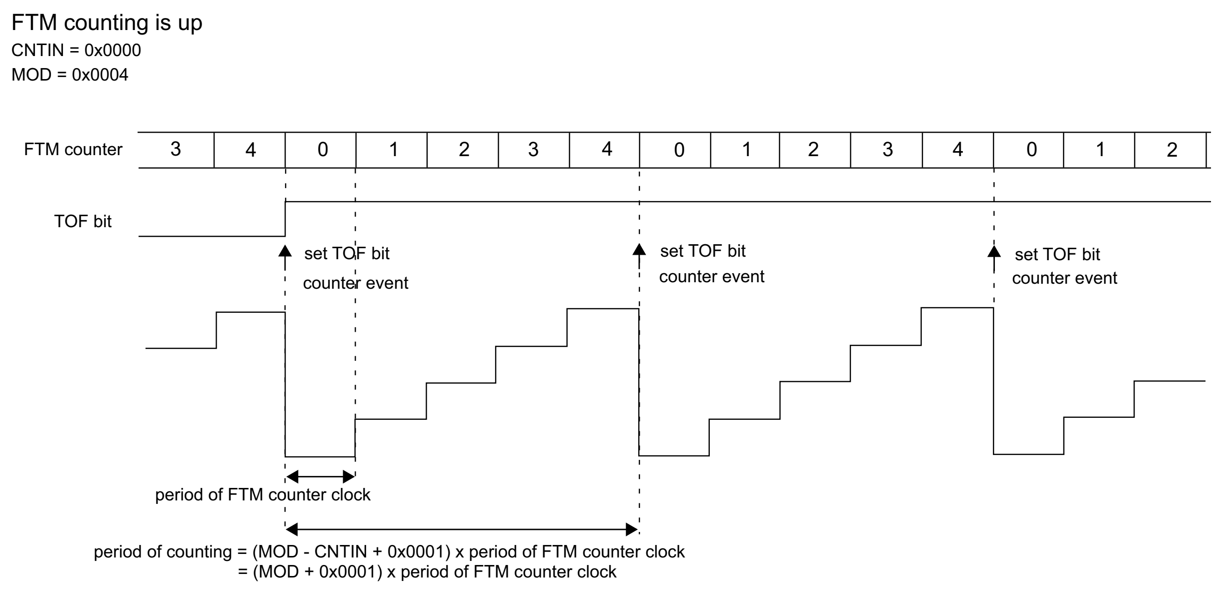


Fig 48. Example of FTM up counting with CNTIN = 0x0000

Note:

- FTM operation is only valid when the value of the CNTIN register is less than the value of the MOD register, either in the unsigned counting or signed counting. It is the responsibility of the software to ensure that the values in the CNTIN and MOD registers meet this requirement. Any values of CNTIN and MOD that do not satisfy this criteria can result in unpredictable behavior.
- MOD = CNTIN is a redundant condition. In this case, the FTM counter is always equal to MOD and the TOF bit is set in each rising edge of the FTM counter clock.
- When MOD = 0x0000, CNTIN = 0x0000, for example after reset, and FTMEN = 1, the FTM counter remains stopped at 0x0000 until a non-zero value is written into the MOD or CNTIN registers.
- Setting CNTIN to be greater than the value of MOD is not recommended as this unusual setting may make the FTM operation difficult to comprehend. However, there is no restriction on this configuration, and an example is shown in the following Figure 49 “Example of up counting when the value of CNTIN is greater than the value of MOD”

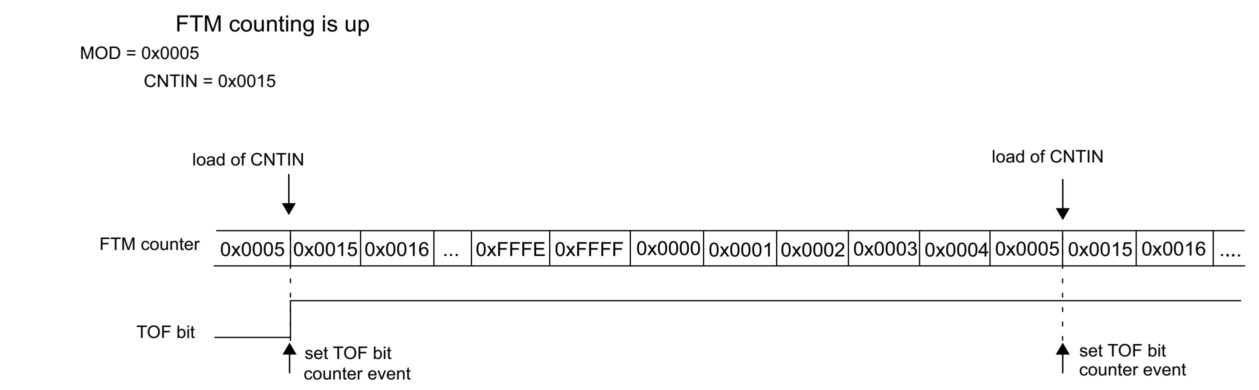


Fig 49. Example of up counting when the value of CNTIN is greater than the value of MOD

24.8.3.2 Up-down counting

Up-down counting is selected when:

- QUADEN = 0, and
- CPWMS = 1

CNTIN defines the starting value of the count and MOD defines the final value of the count. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to the value of CNTIN and the up-down counting restarts.

The FTM period when using up-down counting is 2 × (MOD – CNTIN) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to (MOD – 1). If (CNTIN = 0x0000), the FTM counting is equivalent to TPM up-down counting, that is, up-down and unsigned counting. See the following figure.

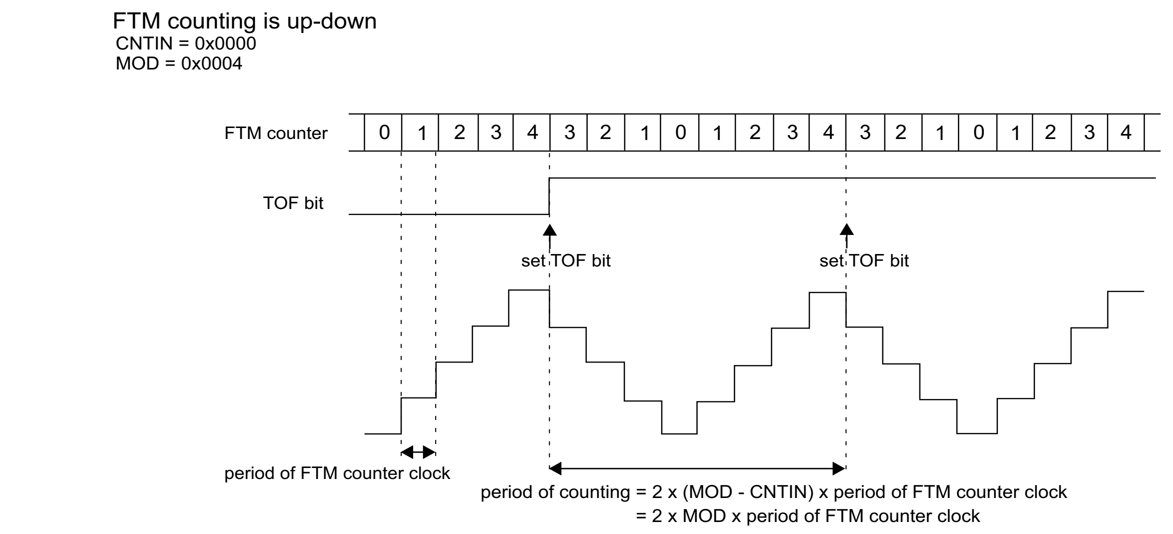

FTM counting is up-down
CNTIN = 0x0000
MOD = 0x0004
FTM counter
0 1 2 3 4 3 2 1 0 1 2 3 4 3 2 1 0 1 2 3 4
TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = 2 x (MOD - CNTIN) x period of FTM counter clock
= 2 x MOD x period of FTM counter clock

Fig 50. Example of up down counting when CNTIN = 0x0000

Note: When CNTIN is different from zero in the up-down counting, a valid CPWM signal is generated:

- if CnV > CNTIN, or
- if CnV = 0 or if CnV[15] = 1. In this case, 0% CPWM is generated.

The figure below shows the possible counter events when in up-down counting mode. See ”Counter events” for more details.

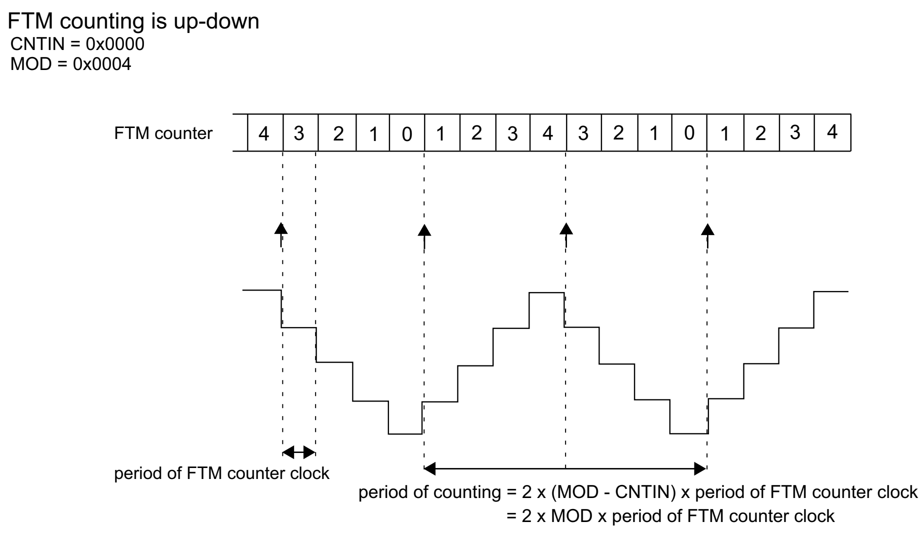

FTM counting is up-down
CNTIN = 0x0000
MOD = 0x0004
FTM counter
4 3 2 1 0 1 2 3 4 3 2 1 0 1 2 3 4
period of FTM counter clock
period of counting = 2 x (MOD - CNTIN) x period of FTM counter clock
= 2 x MOD x period of FTM counter clock

Fig 51. Example of counter events in up-down counting mode when CNTIN = 0x0000

24.8.3.3 Free running counter

If (FTMEN = 0) and (MOD = 0x0000 or MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF

and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000. See the following figure.

A counter event occurs at the same time of TOF bit set when the FTM counter changes from 0xFFFF to 0x0000. See ”Counter events” for more details.

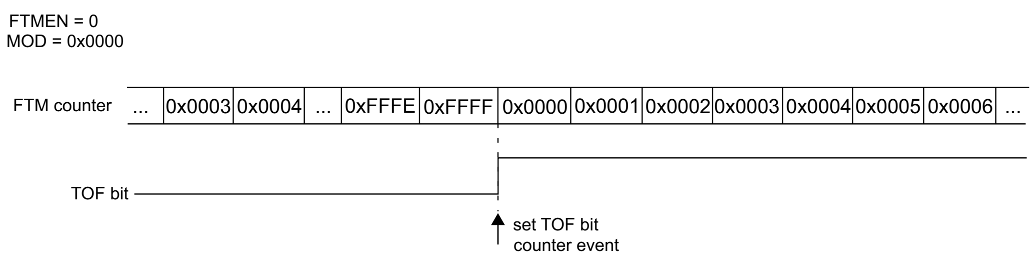

FTMEN = 0
MOD = 0x0000
FTM counter
... 0x0003 0x0004 ... 0xFFFE 0xFFFF 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 ...
TOF bit
set TOF bit
counter event

Fig 52. Example when FTM counter is free running

The FTM counter is also a free running counter when:

- FTMEN = 1
- QUADEN = 0

- CPWMS = 0
- CNTIN = 0x0000, and
- MOD = 0xFFFF

24.8.3.4 Counter reset

Any one of the following cases resets the FTM counter to the value in the CNTIN register and the channels output to its initial value, except for channels in Output Compare mode.

- Any write to CNT.
- ”FTM counter synchronization”.
- A channel in Input Capture mode with ICRST = 1 (”FTM Counter Reset in Input Capture Mode”).

Note that reseting the counter also generates a counter event. See ”Counter events” for more details.

24.8.3.5 Counter events

Counter events can be used as reload opportunities to FTM register sychronization mechanism. See ”Reload Points” for more details. There are some possible counter events depending on the counter mode. Please see the table below for more details.

Table 419. FTM counter events

When	Then
FTM counter is in up counting mode or freerunning	• A counter event happens at the same time of TOF set when the FTM counter changes from MOD to CNTIN (counter wrap). Figure at ”Up counting” shows the counter event generation. • When in freerunning, there is a counter event when FTM counter changes from 0xFFFF to 0x0000. Figure at ”Free running counter” shows the counter event generation.
FTM counter is in up-down counting mode	• In up-down counting mode, there are two possible counter events when FTM counter turns from down to up counting and when counter turns from up to down counting. User can select which point will be used to generate the counter event. Figure at Up-down counting shows the possible counter events.
FTM counter is reseted (see ”Counter reset”) or a value different from zero is written at CLKS field	• In up-counting mode, all counter reset events or a write in the CLKS with a value different from zero generates a counter event. • In up-down counting mode, counter reset events only generates a counter event if the minimum load point when FTM counter turns from down to up counting is configured. A write in the CLKS with a value different from zero always generates a counter event in up-down counting mode.

24.8.4 Channel Modes

The following table shows the channel modes selection.

Table 420. Channel Modes Selection

When	Then					
X	X	X	XX	00	Pin not used for FTM—revert the channel pin to general purpose I/O or other peripheral control	
0	0	0	00	01	Input Capture	Capture on Rising Edge Only
				10		Capture on Falling Edge Only
				11		Capture on Rising or Falling Edge
			01	01	Output Compare	Toggle Output on match
				10		Clear Output on match
				11		Set Output on match
			1X	10	Edge-Aligned PWM	High-true pulses (clear Output on match)
				X1		Low-true pulses (set Output on match)
		1	XX	10	Center-Aligned PWM	High-true pulses (clear Output on match-up)
				X1		Low-true pulses (set Output on match-up)
	1	0	XX	10	Combine PWM	High-true pulses (set on channel (n) match, and clear on channel (n+1) match)
				X1		Low-true pulses (clear on channel (n) match, and set on channel (n +1) match)
1	0	0	X0	See Table 421 “Dual Edge Capture Mode - Edge Polarity Selection”	Dual Edge Capture	One-Shot Capture mode
			X1			Continuous Capture mode

Table 421. Dual Edge Capture Mode - Edge Polarity Selection

ELSB	ELSA	Channel Port Enable	Detected Edges
0	0	Disabled	No edge
0	1	Enabled	Rising edge
1	0	Enabled	Falling edge
1	1	Enabled	Rising and falling edges

24.8.5 Input Capture mode

The Input Capture mode is selected when:

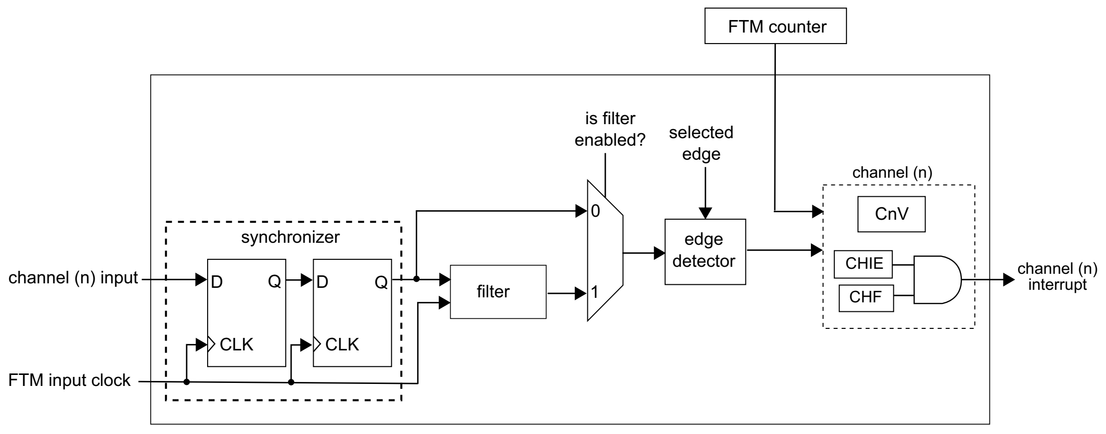
- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0
- MSB:MSA = 0:0, and
- ELSB:ELSA ≠ 0:0

When a selected edge occurs on the channel input, the current value of the FTM counter is captured into the CnV register, at the same time the CHF bit is set and the channel interrupt is generated if enabled by CHIE = 1. See the following figure.

When a channel is configured for input capture, the FTMxCHn pin is an edge-sensitive input. ELSB:ELSA bits determine which edge, falling or rising, triggers input-capture event.

Writes to the CnV register are ignored in input capture mode.

While in Debug mode, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of Debug, is captured into the CnV register and the CHF bit is set.



Note: The filter is only available for the channels 0, 1, 2, and 3 inputs.

Fig 53. Diagram for Input Capture Mode

24.8.5.1 Filter for Input Capture Mode

The filter is only available on channels 0, 1, 2, and 3.

The channel input after being synchronized by FTM input clock (Figure 53 “Diagram for Input Capture Mode”) is the filter input.

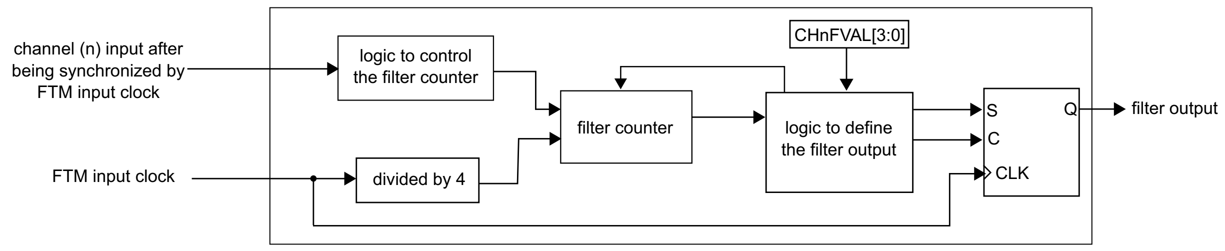

channel (n) input after being synchronized by FTM input clock
FTM input clock
logic to control the filter counter
divided by 4
filter counter
CHnFVAL[3:0]
logic to define the filter output
S
C
CLK
Q
filter output

Fig 54. Channel Input Filter

Note: The maximum frequency for the channel input to be detected correctly is FTM input clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

When there is a state change in the channel input, the counter is reset and starts counting up. As long as the new state is stable on the channel input, the counter continues to increment. When the counter is equal to CHnFVAL[3:0], the new channel input signal value is validated. It is then transmitted as a pulse to the edge detector.

If the opposite edge appears on the channel input signal before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. If a pulse is sampled as a value less than (CHnFVAL[3:0] x 4) consecutive rising edges of FTM input clock, it is regarded as a glitch and is not passed on to the edge detector.

The table below shows the delay that is added by the FTM channel input filter according to its configuration.

Table 422. FTM counting based on CNTIN value

FTM channel input filter	Number of rising edges between the selected edge on channel input and setting CHF bit
• channel does not have the input filter, or • channel input filter is disabled (CHnFVAL[3:0] = 0)	• 3 rising edges of FTM input clock
• channel has the input filter, and • channel input filter is enabled (CHnFVAL[3:0]≠0)	• (4 + 4 × CHnFVAL[3:0]) rising edges of FTM input clock

The following figure illustrates an example of channel input filter.

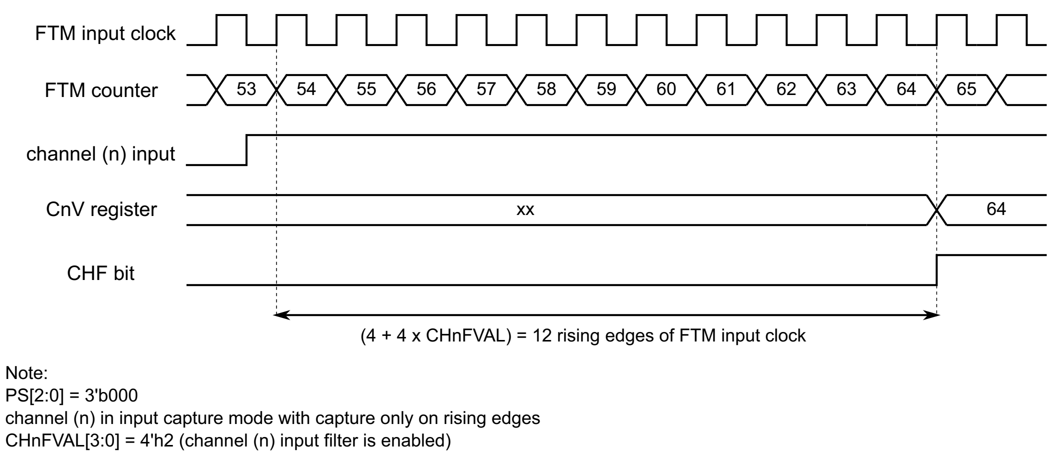

FTM input clock
FTM counter
53
54
55
56
57
58
59
60
61
62
63
64
65
channel (n) input
CnV register
xx
64
CHF bit
(4 + 4 x CHnFVAL) = 12 rising edges of FTM input clock
Note:
PS[2:0] = 3'b000
channel (n) in input capture mode with capture only on rising edges
CHnFVAL[3:0] = 4'h2 (channel (n) input filter is enabled)

Fig 55. Example of Channel Input Filter

24.8.5.2 FTM Counter Reset in Input Capture Mode

If the channel (n) is in input capture mode and CnSC[ICRST = 1], then when the selected input capture event occurs in the channel (n) input signal, the current value of the FTM counter is captured into the CnV register, the CHF bit is set, the channel (n) interrupt is generated (if CHIE = 1) and the FTM counter is reset to the CNTIN register value.

This allows the FTM to measure a period/pulse being applied to the channel (n) input (number of the FTM input clocks) without having to implement a subtraction calculation in software subsequent to the event occurring.

The figure below shows the FTM counter reset when the selected input capture event is detected in a channel in input capture mode with ICRST = 1.

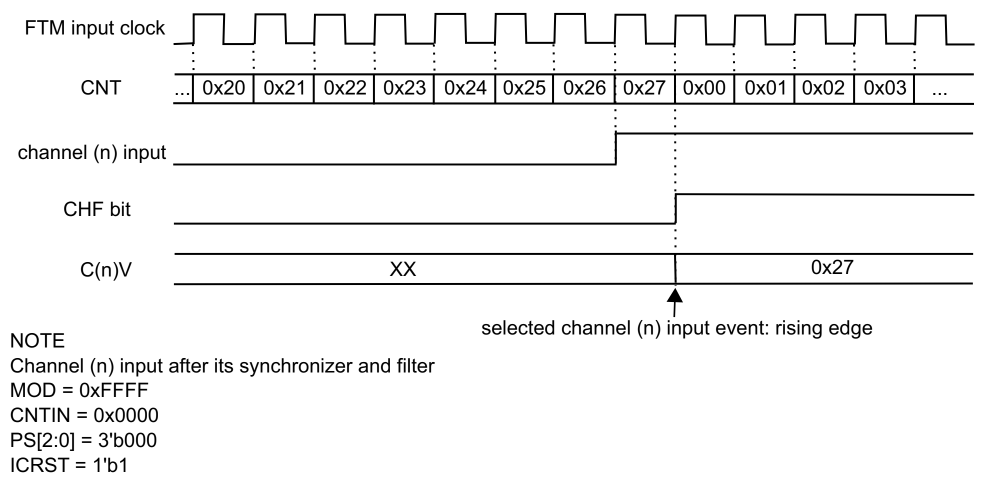


Fig 56. Example of the Input Capture Mode with ICRST = 1

Note

- It is expected that the ICRST bit be set only when the channel is in input capture mode.

- If the FTM counter is reset because the channel is in input capture mode with ICRST = 1, then the prescaler counter ("Prescaler") is also reset.

24.8.6 Output Compare mode

The Output Compare mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSB:MSA = 0:1

In Output Compare mode, the FTM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared, or toggled.

When a channel is initially configured to Toggle mode, the previous value of the channel output is held until the first output compare event occurs.

The CHF bit is set and the channel (n) interrupt is generated if CHIE = 1 at the channel (n) match (FTM counter = CnV).

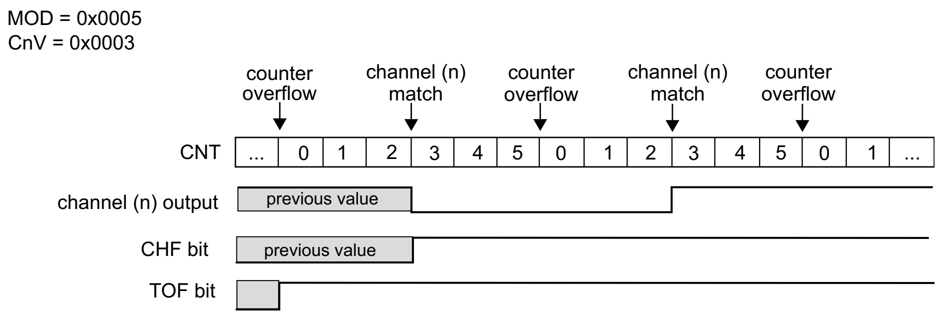


Fig 57. Example of the Output Compare mode when the match toggles the channel output

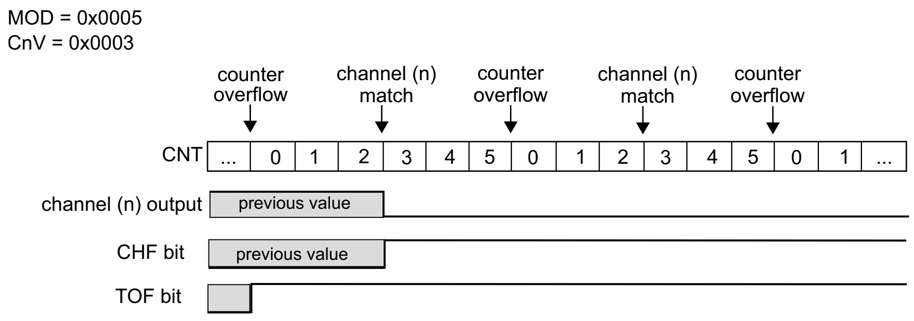


Fig 58. Example of the Output Compare mode when the match clears the channel output

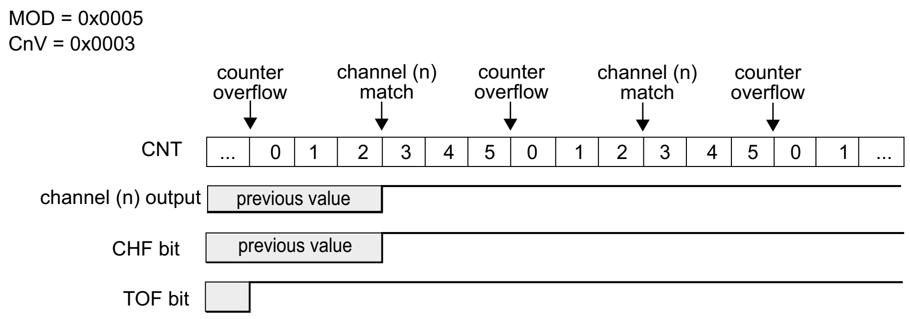


Fig 59. Example of the Output Compare mode when the match sets the channel output

If (ELSB:ELSA = 0:0) when the counter reaches the value in the CnV register, the CHF bit is set and the channel (n) interrupt is generated if CHIE = 1, however the channel (n) output is not modified and controlled by FTM.

24.8.7 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSB = 1

The EPWM period is determined by (MOD - CNTIN + 0x0001) and the pulse width (duty cycle) is determined by (CnV - CNTIN).

The CHF bit is set and the channel (n) interrupt is generated if CHIE = 1 at the channel (n) match (FTM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an FTM.

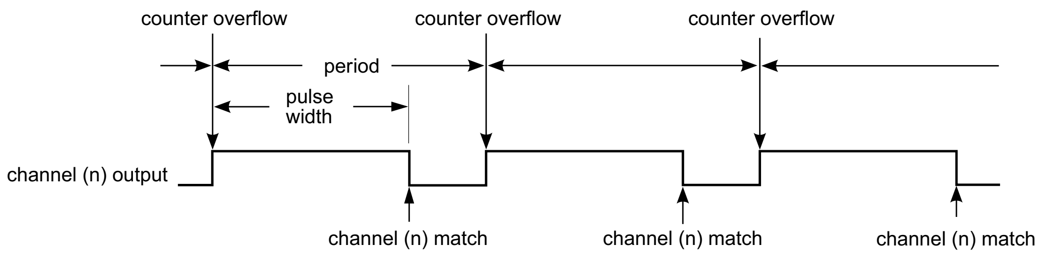


Fig 60. EPWM period and pulse width with ELSB:ELSA = 1:0

If (ELSB:ELSA = 0:0) when the counter reaches the value in the CnV register, the CHF bit is set and the channel (n) interrupt is generated if CHIE = 1, however the channel (n) output is not controlled by FTM.

If (ELSB:ELSA = 1:0), then the channel (n) output is forced high at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced low at the channel (n) match (FTM counter = CnV). See the following figure.

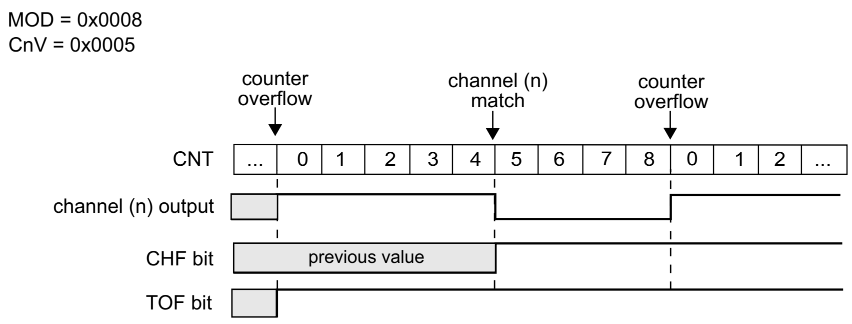


Fig 61. EPWM signal with ELSB:ELSA = 1:0

If (ELSB:ELSA = X:1), then the channel (n) output is forced low at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced high at the channel (n) match (FTM counter = CnV). See the following figure.

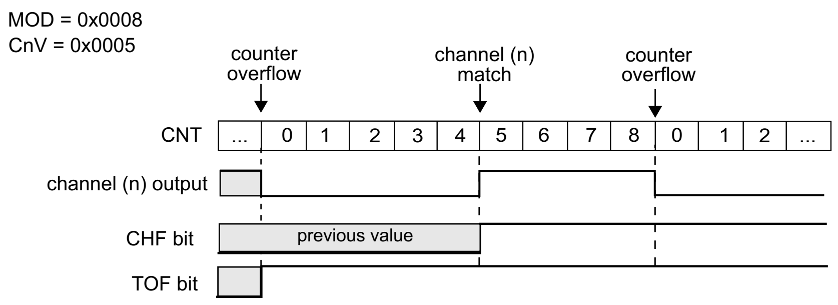


Fig 62. EPWM signal with ELSB:ELSA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal and CHF bit is not set even when there is the channel (n) match.

If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHF bit is not set. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

Note: When CNTIN is different from zero the following EPWM signals can be generated:

- 0% EPWM signal if CnV = CNTIN,
- EPWM signal between 0% and 100% if CNTIN < CnV <= MOD,
- 100% EPWM signal when CNTIN > CnV or CnV > MOD.

24.8.8 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0

- COMBINE = 0, and
- CPWMS = 1

The CPWM pulse width (duty cycle) is determined by 2 × (CnV - CNTIN) and the period is determined by 2 × (MOD - CNTIN). See the following figure. MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

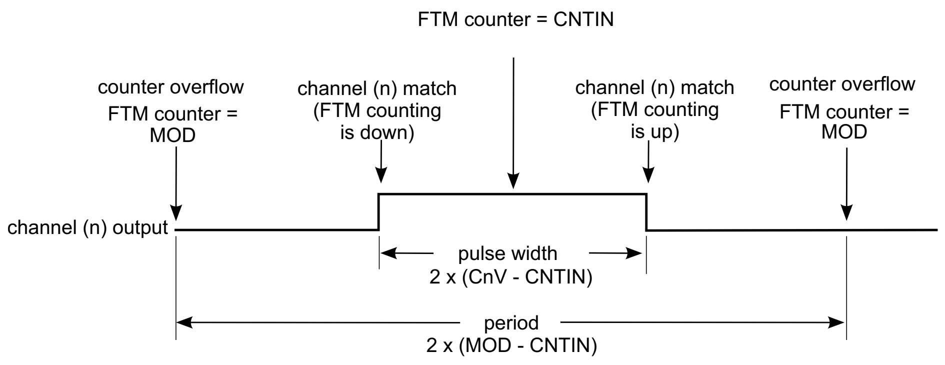


Fig 63. CPWM pulse width

In the CPWM mode, the FTM counter counts up until it reaches MOD and then counts down until it reaches CNTIN.

The CHF bit is set and channel (n) interrupt is generated (if CHIE = 1) at the channel (n) match (FTM counter = CnV) when the FTM counting is down (at the begin of the pulse width) and when the FTM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are aligned with the value of CNTIN.

The other channel modes are not compatible with the up-down counter (CPWMS = 1). Therefore, all FTM channels must be used in CPWM mode when (CPWMS = 1).

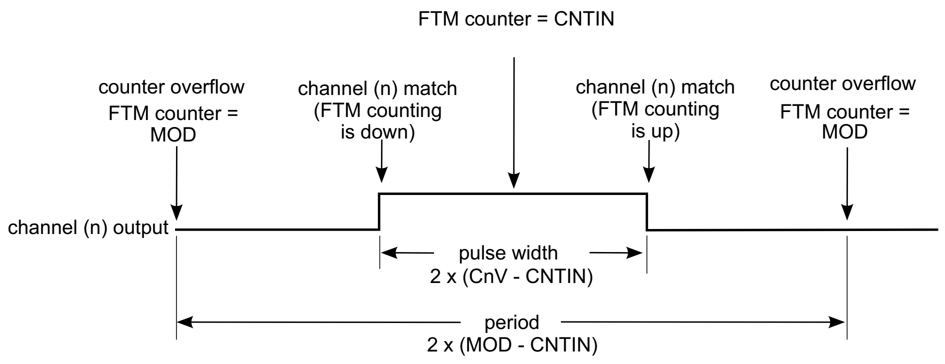


Fig 64. CPWM period and pulse width with ELSA:ELSAA = 1:0

If (ELSB:ELSA = 0:0) when the FTM counter reaches the value in the CnV register, the CHF bit is set and the channel (n) interrupt is generated (if CHIE = 1), however the channel (n) output is not controlled by FTM.

If (ELSB:ELSA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up. See the following figure.

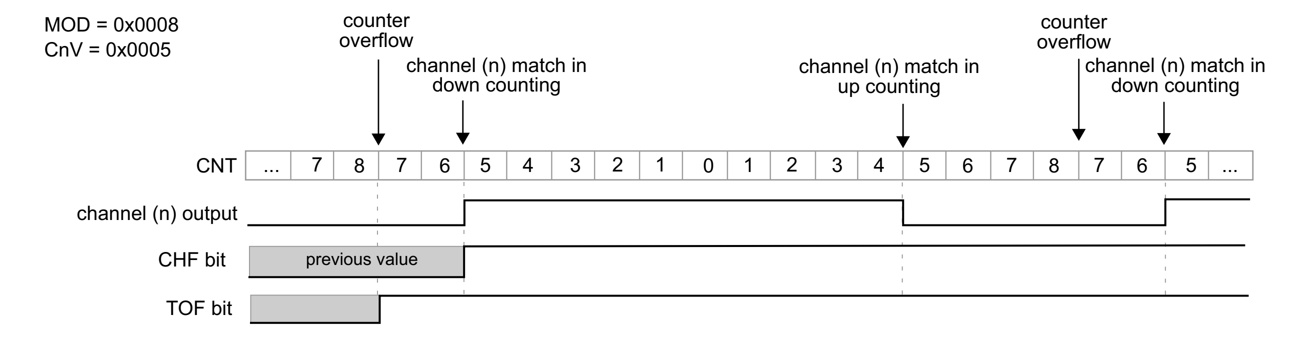


Fig 65. CPWM signal with ELSB:ELSA = 1:0

If (ELSB:ELSA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up. See the following figure.

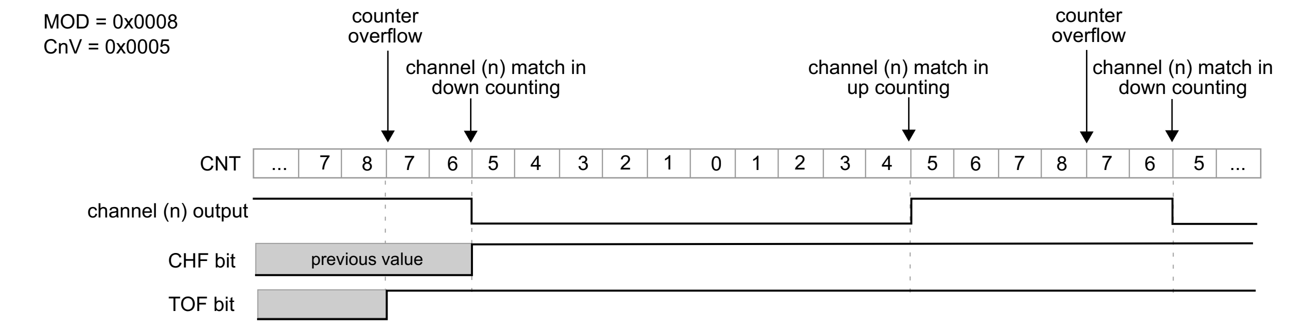


Fig 66. CPWM signal with ELSB:ELSA = X:1

If (CnV = 0x0000) or CnV is a negative value, that is (CnV[15] = 1), then the channel (n) output is a 0% duty cycle CPWM signal and CHF bit is not set even when there is the channel (n) match.

If CnV is a positive value, that is (CnV[15] = 0), (CnV ≥ MOD), and (MOD ≠ 0x0000), then the channel (n) output is a 100% duty cycle CPWM signal and CHF bit is not set even when there is the channel (n) match. This implies that the usable range of periods set by MOD is 0x0001 through 0x7FFE, 0x7FFF if you do not need to generate a 100% duty cycle CPWM signal. This is not a significant limitation because the resulting period is much longer than required for normal applications.

The CPWM mode must not be used when the FTM counter is a free running counter.

24.8.9 Combine mode

The Combine mode is selected when:

- QUADEN = 0
- DECAPEN = 0

- COMBINE = 1, and
- CPWMS = 0

In Combine mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output.

In the Combine mode, the PWM period is determined by (MOD - CNTIN + 0x0001) and the PWM pulse width (duty cycle) is determined by (|C(n+1)V - C(n)V|).

The channel (n) CHF bit is set and its interrupt is generated, if channel (n) CHIE = 1, at the channel (n) match (FTM counter = C(n)V). The channel (n+1) CHF bit is set and its interrupt is generated, if channel (n+1) CHIE = 1, at the channel (n+1) match (FTM counter = C(n+1)V).

If channel (n) ELSB:ELSA = 1:0, then the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n+1)V). It is forced high at the channel (n) match (FTM counter = C(n)V). See the following figure.

If channel (n) ELSB:ELSA = X:1, then the channel (n) output is forced high at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n+1)V). It is forced low at the channel (n) match (FTM counter = C(n)V). See the following figure.

In Combine mode, the channel (n+1) ELSB:ELSA bits are not used in the generation of the channels (n) and (n+1) output. However, if channel (n) ELSB:ELSA = 0:0, then the channel (n) output is not controlled by FTM, and if channel (n+1) ELSB:ELSA = 0:0, then the channel (n+1) output is not controlled by FTM.

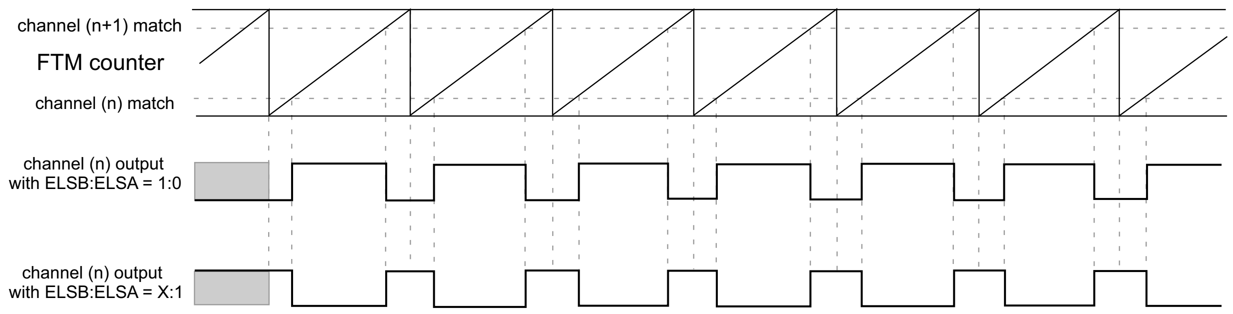

channel (n+1) match
FTM counter
channel (n) match
channel (n) output with ELSB:ELSA = 1:0
channel (n) output with ELSB:ELSA = X:1

Fig 67. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

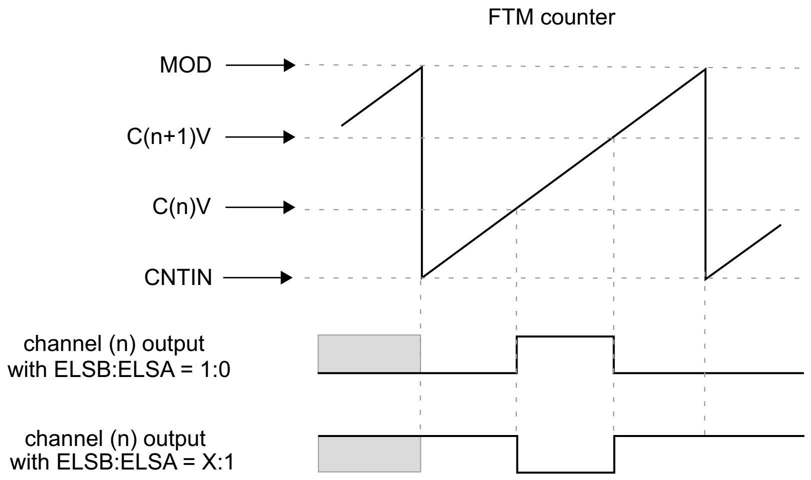


Fig 68. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V < C(n+1)V)

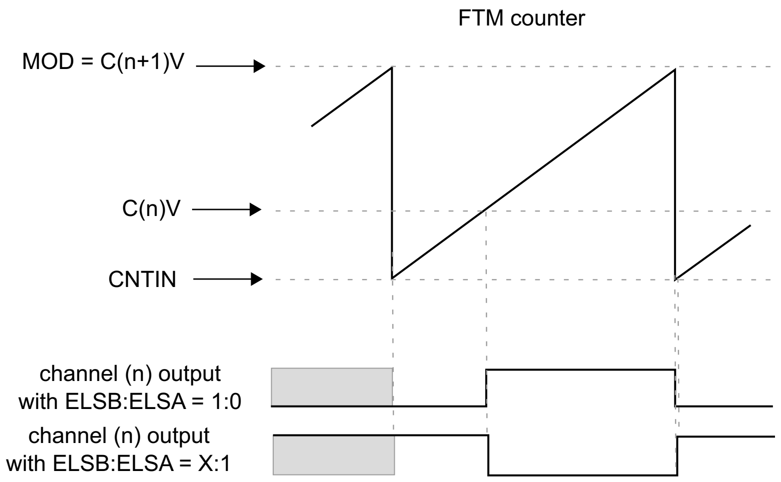


Fig 69. Channel (n) output if (CNTIN < C(n)V < MOD) and C(n+1)V = MOD

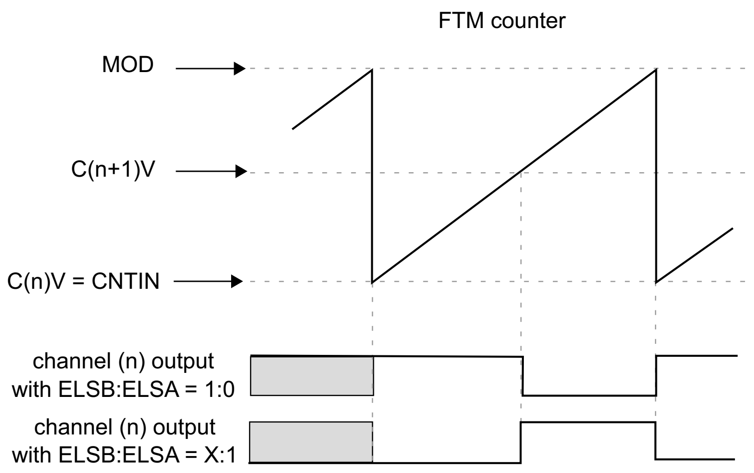


Fig 70. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD)

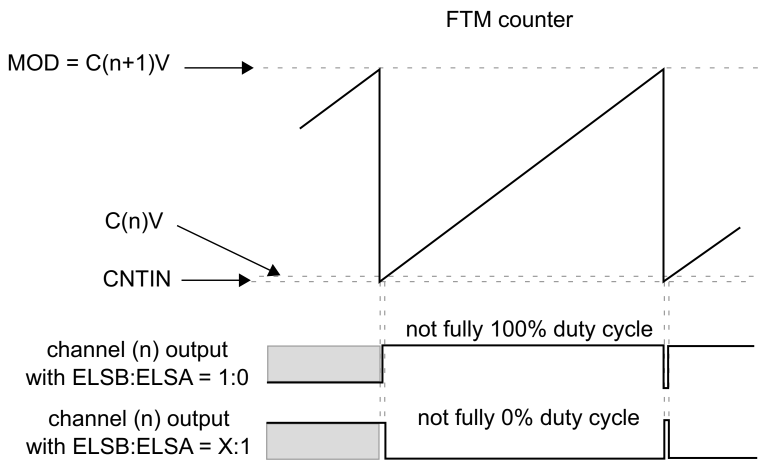


Fig 71. Channel (n) output if (CNTIN < C(n)V < MOD) and C(n)V is Almost Equal to CNTIN) and (C(n+1)V = MOD)

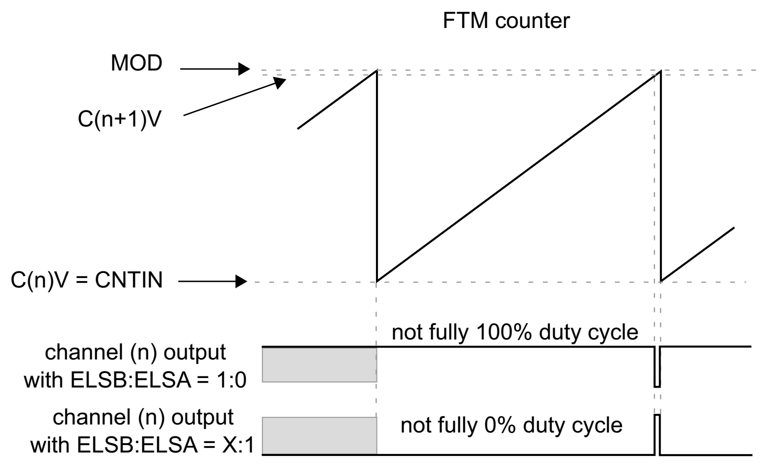


Fig 72. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD) and (C(n+1)V is Almost Equal to MOD)

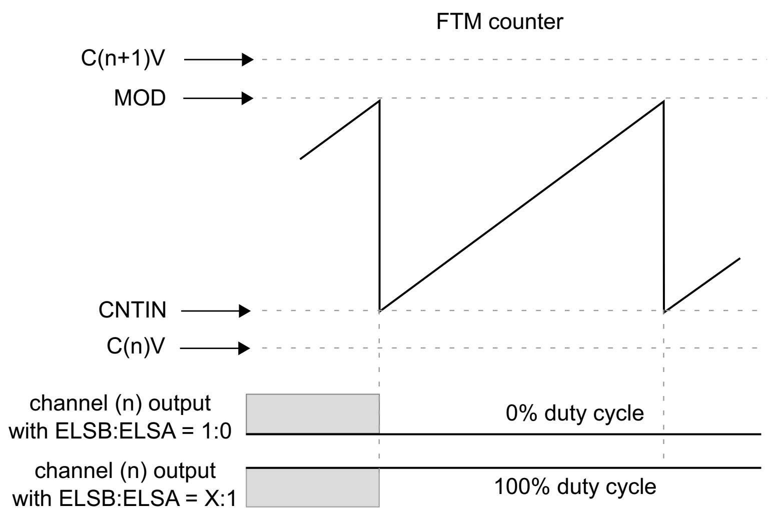


Fig 73. Channel (n) output if C(n)V and C(n+1)V are not between CNTIN and MOD

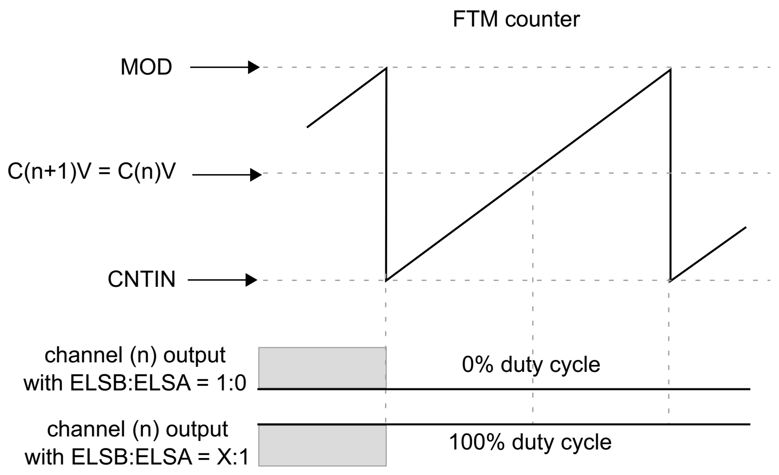


Fig 74. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)v < MOD) and (C(n)V = C(n+1)V)

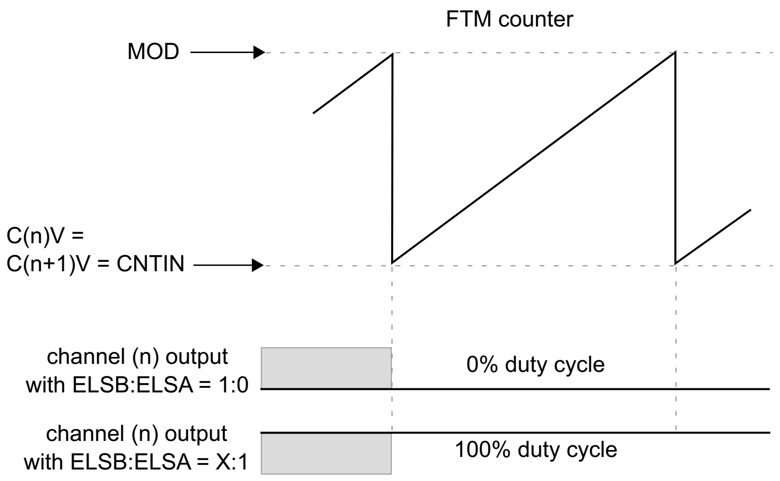


Fig 75. Channel (n) output if (C(n)V = C(n+1)V = CNTIN)

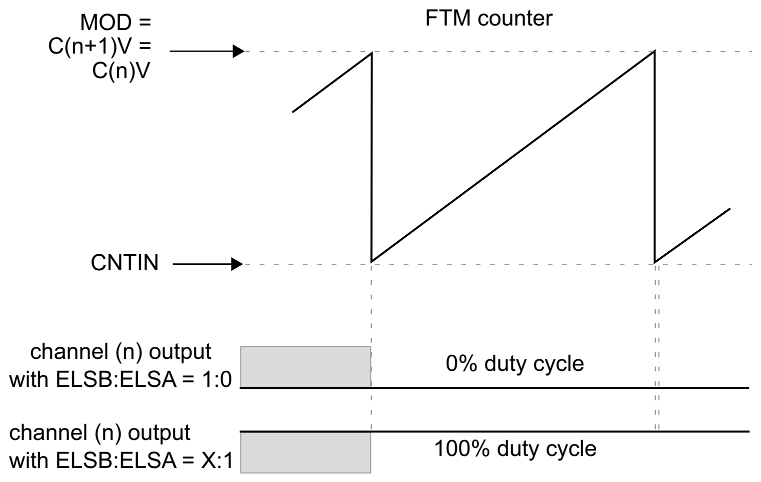


Fig 76. Channel (n) output if (C(n)V = C(n+1)V = MOD)

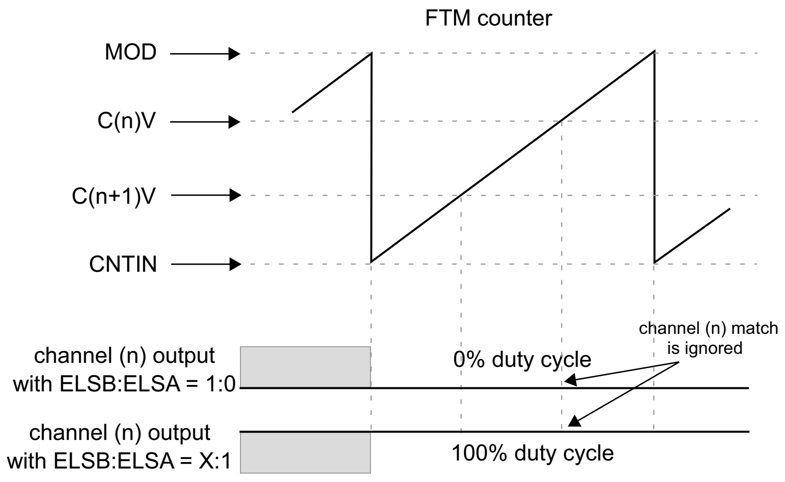


Fig 77. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V > C(n+1)V)

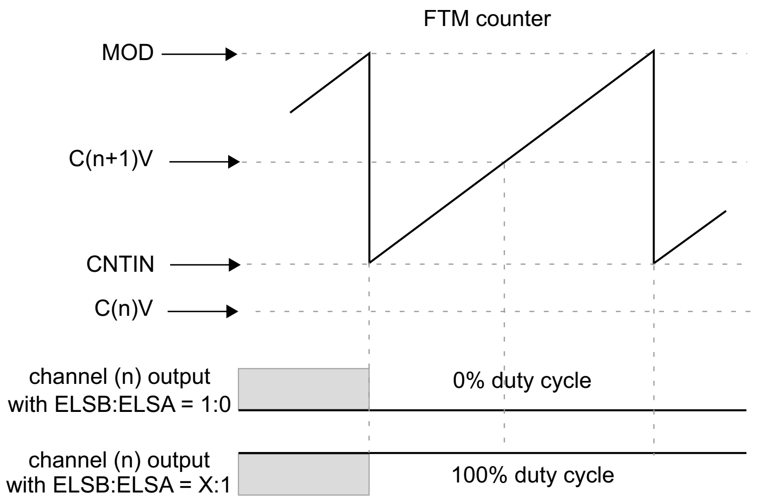


Fig 78. Channel (n) output if (C(n)V < CNTIN) and (CNTIN < C(n+1)V < MOD)

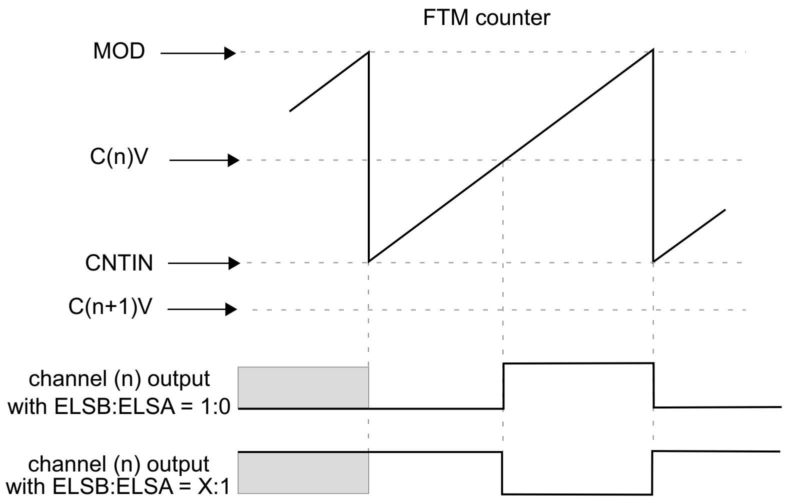


Fig 79. Channel (n) output if (C(n+1)V < CNTIN) and (CNTIN < C(n)V < MOD)

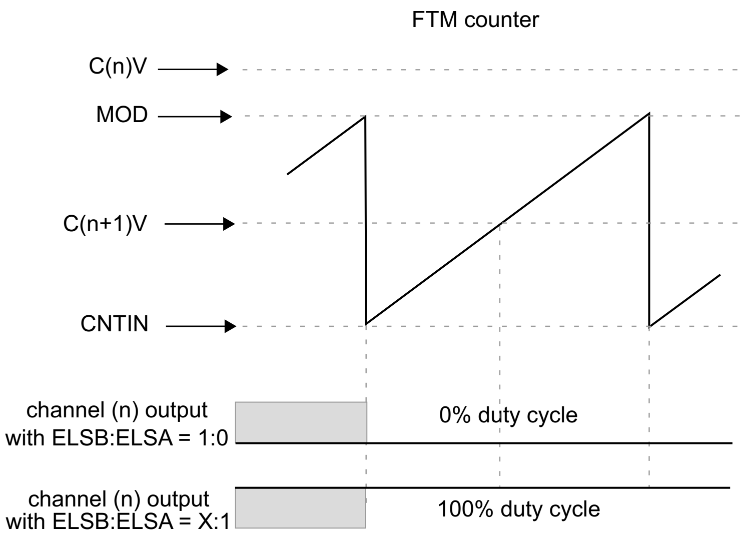


Fig 80. Channel (n) output if (C(n)V > MOD) and (CNTIN < C(n+1)V < MOD)

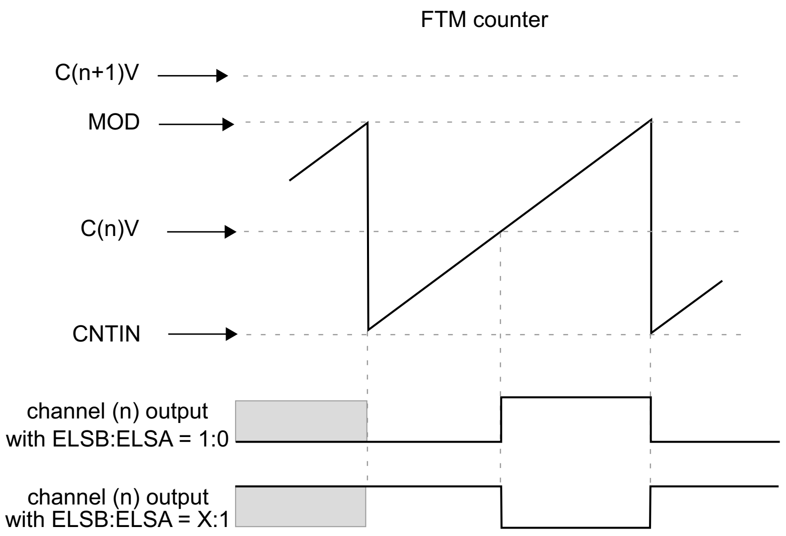


Fig 81. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V < MOD)

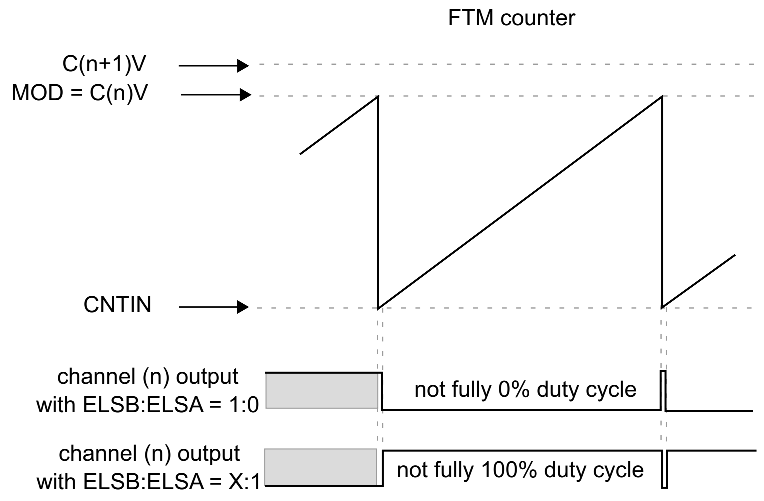

FTM counter
C(n+1)V
MOD = C(n)V
CNTIN
channel (n) output with ELSB:ELSA = 1:0
not fully 0% duty cycle
channel (n) output with ELSB:ELSA = X:1
not fully 100% duty cycle

Fig 82. Channel (n) output if (C(n+1)V > MOD) and (CNTIN <C(n)V = MOD)

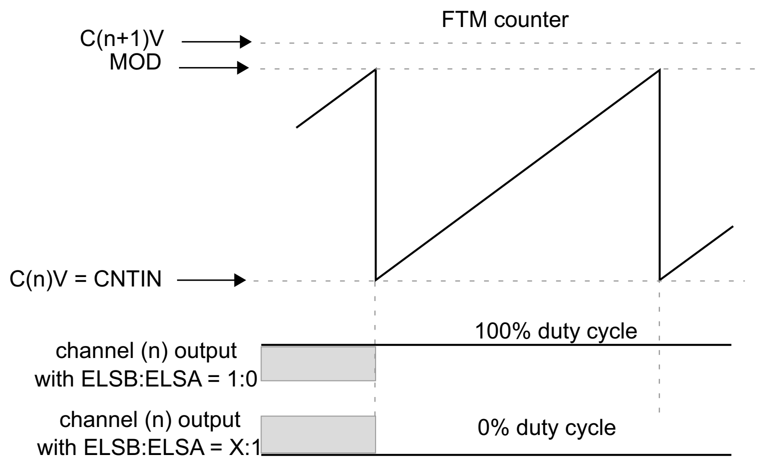

FTM counter
C(n+1)V
MOD
C(n)V = CNTIN
100% duty cycle
channel (n) output with ELSB:ELSA = 1:0
channel (n) output with ELSB:ELSA = X:1
0% duty cycle

Fig 83. Channel (n) output if (C(n)V = CNTIN) and (C(n+1)V > MOD)

24.8.9.1 Asymmetrical PWM

In ”Combine mode”, the PWM first edge (channel (n) match: FTM counter = C(n)V) is independent of the PWM second edge (channel (n+1) match: FTM counter = C(n+1)V).

24.8.10 Complementary Mode

The Complementary mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 1

In Complementary mode, the channel (n+1) output is the inverse of the channel (n) output.

Note: The Complementary Mode is not available on "Output Compare mode".

The channel (n+1) output is the same as the channel (n) output when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 0
- channels (n) and (n+1) are on Combine Mode

The channel (n+1) output is independent from channel (n) output when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 0
- channel (n) is on Output Compare Mode, EPWM or CPWM

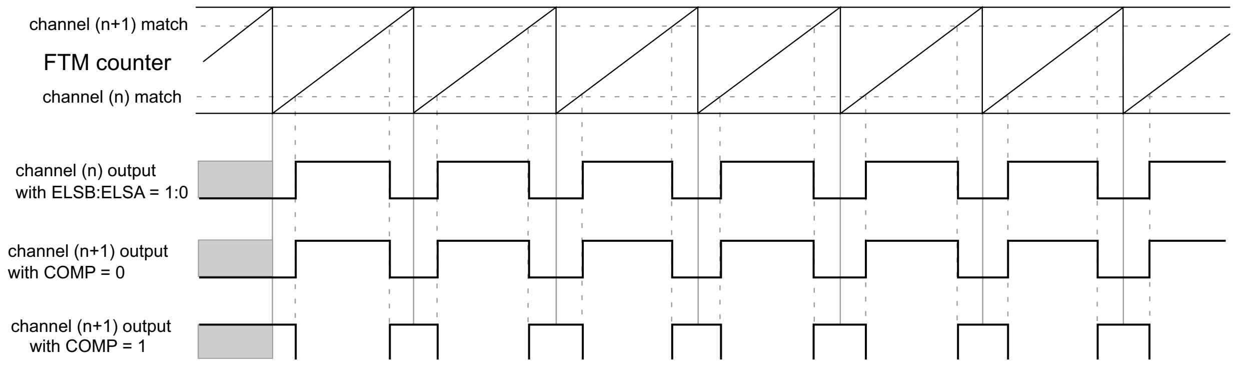

channel (n+1) match
FTM counter
channel (n) match
channel (n) output with ELSB:ELSA = 1:0
channel (n+1) output with COMP = 0
channel (n+1) output with COMP = 1

Fig 84. Channel (n+1) output in Complementary mode with (ELSB:ELSA = 1:0)

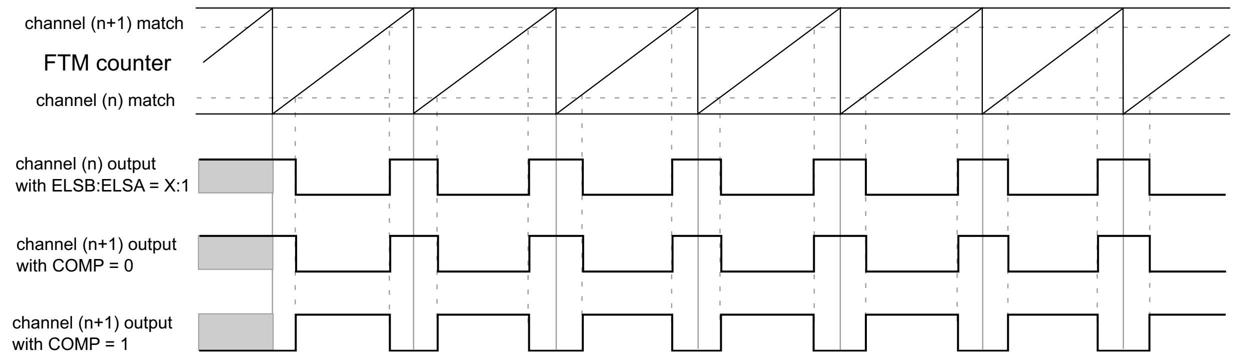

channel (n+1) match
FTM counter
channel (n) match
channel (n) output with ELSB:ELSA = X:1
channel (n+1) output with COMP = 0
channel (n+1) output with COMP = 1

Fig 85. Channel (n+1) output in Complementary mode with (ELSB:ELSA = X:1)

24.8.11 Registers updated from write buffers

24.8.11.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 423. CNTIN register update

When	Then CNTIN register is updated
CLKS[1:0] = 0:0	When CNTIN register is written, independent of FTMEN bit.
• FTMEN = 0, or • CNTINC = 0	At the next FTM input clock after CNTIN was written.
• FTMEN = 1, • SYNCMODE = 1, and • CNTINC = 1	By the ”CNTIN register synchronization”.
• CNTINC = 1, and • LDOK = 1	By the ”Reload Points”.

24.8.11.2 MOD and HCR registers update

The following table describes when MOD or HCR registers are updated:

Table 424. MOD and HCR updates

When	The MOD and HCR is updated
CLKS[1:0] = 0:0	When MOD (or HCR) is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the CPWMS bit, that is: • If the selected mode is not CPWM then MOD (or HCR) is updated after MOD (or HCR) register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM then MOD (or HCR) register is updated after MOD (or HCR) register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	By the ”MOD register synchronization”. HCR follows the same procedure of MOD register in this case.
LDOK = 1	By the ”Reload Points”.

24.8.11.3 CnV register update

The following table describes when CnV register is updated:

Table 425. CnV register update

When	Then CnV register is updated
CLKS[1:0] = 0:0	When CnV register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the selected mode, that is: • If the selected mode is Output Compare, then CnV register is updated on the next FTM counter change, end of the prescaler counting, after CnV register was written. • If the selected mode is EPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, • FTMEN =1	According to the selected mode, that is: • If the selected mode is output compare then CnV register is updated according to the SYNCEN bit. If (SYNCEN = 0) then CnV register is updated after CnV register was written at the next change of the FTM counter, the end of the prescaler counting. If (SYNCEN = 1) then CnV register is updated by the "C(n)V and C(n+1)V register synchronization". • If the selected mode is not output compare and (SYNCEN = 1) then CnV register is updated by the "C(n)V and C(n+1)V register synchronization".
• SYNCEN = 1, and • LDOK = 1	By the "Reload Points".

24.8.12 PWM synchronization

The PWM synchronization provides an opportunity to update the MOD, HCR, CNTIN, CnV, OUTMASK, INVCTRL and SWOCTRL registers with their buffered value and force the FTM counter to the CNTIN register value.

Note: The legacy PWM synchronization (SYNCMODE = 0) is a subset of the enhanced PWM synchronization (SYNCMODE = 1). Thus, only the enhanced PWM synchronization must be used.

24.8.12.1 Hardware trigger

Three hardware trigger signal inputs of the FTM module are enabled when TRIGn = 1, where n = 0, 1 or 2 corresponding to each one of the input signals, respectively. The hardware trigger input n is synchronized by the FTM input clock. The PWM synchronization with hardware trigger is initiated when a rising edge is detected at the enabled hardware trigger inputs.

If (HWTRIGMODE = 0) then the TRIGn bit is cleared when 0 is written to it or when the trigger n event is detected.

In this case, if two or more hardware triggers are enabled (for example, TRIG0 and TRIG1 = 1) and only trigger 1 event occurs, then only TRIG1 bit is cleared. If a trigger n event occurs together with a write setting TRIGn bit, then the synchronization is initiated, but TRIGn bit remains set due to the write operation.

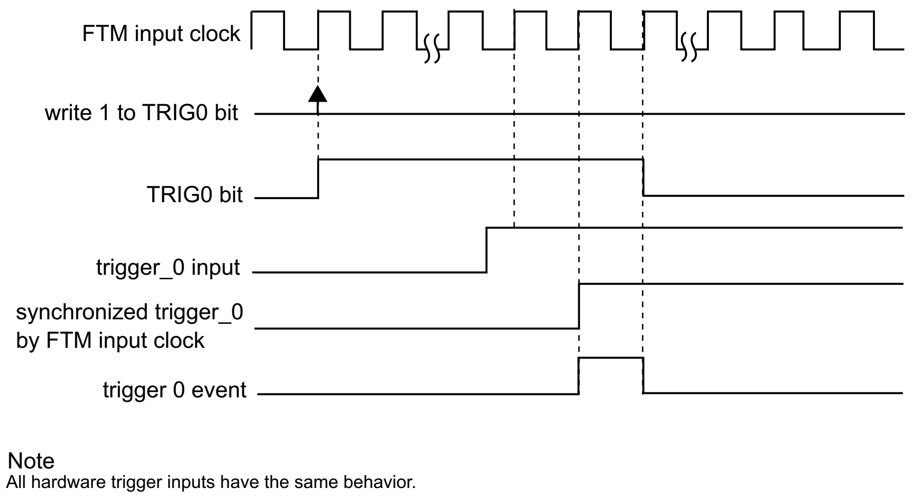


Fig 86. Hardware trigger event with HWTRIGMODE = 0

If HWTRIGMODE = 1, then the TRIGn bit is only cleared when 0 is written to it.

Note: The HWTRIGMODE bit must be 1 only with enhanced PWM synchronization (SYNCMODE = 1).

24.8.12.2 Software trigger

A software trigger event occurs when 1 is written to the SYNC[SWSYNC] bit. The SWSYNC bit is cleared when 0 is written to it or when the PWM synchronization, initiated by the software event, is completed.

If another software trigger event occurs (by writing another 1 to the SWSYNC bit) at the same time the PWM synchronization initiated by the previous software trigger event is ending, a new PWM synchronization is started and the SWSYNC bit remains equal to 1.

If SYNCMODE = 0 then the SWSYNC bit is also cleared by FTM according to PWMSYNC and REINIT bits. In this case if (PWMSYNC = 1) or (PWMSYNC = 0 and REINIT = 0) then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see ”Synchronization Points” and the following figure. If (PWMSYNC = 0) and (REINIT = 1) then SWSYNC bit is cleared when the software trigger event occurs.

If SYNCMODE = 1 then the SWSYNC bit is also cleared by FTM according to the SWRSTCNT bit. If SWRSTCNT = 0 then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see the following figure. If SWRSTCNT = 1 then SWSYNC bit is cleared when the software trigger event occurs.

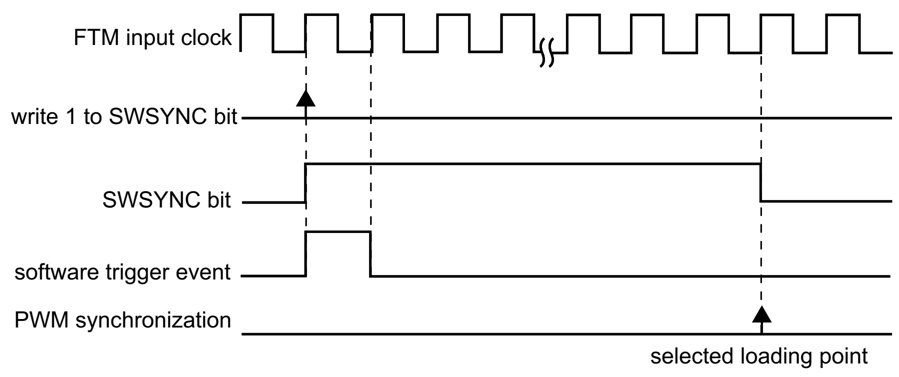


Fig 87. Software trigger event

24.8.12.3 Synchronization Points

The synchronization points are points where the registers can be updated with their write buffer by PWM synchronization. These synchronization points are safe points because guarantee smooth transitions in the generated PWM signals.

In "Up counting", the synchronization points are when the FTM counter changes from MOD to CNTIN. In this case, the synchronization points are enabled if (CNTMIN = 1) or (CNTMAX = 1).

In "Up-down counting", the synchronization points are:

- if (CNTMAX = 1), when the FTM counter changes from (MOD) to (MOD - 1);
- if (CNTMIN = 1), when the FTM counter changes from (CNTIN) to (CNTIN + 1).

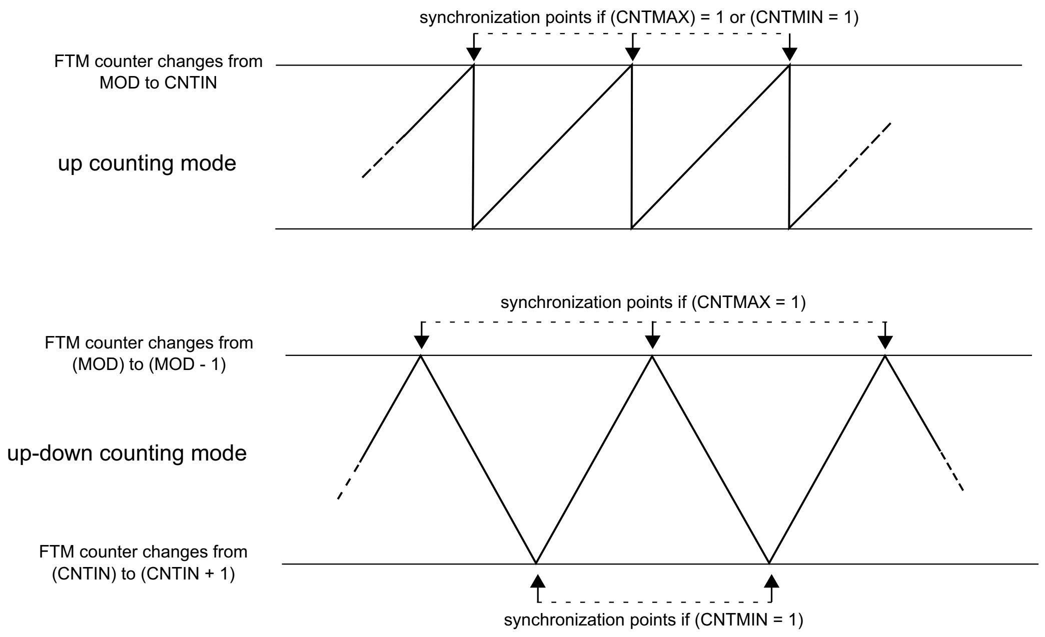

synchronization points if (CNTMAX) = 1 or (CNTMIN = 1)
FTM counter changes from MOD to CNTIN
up counting mode
synchronization points if (CNTMAX = 1)
FTM counter changes from (MOD) to (MOD - 1)
up-down counting mode
FTM counter changes from (CNTIN) to (CNTIN + 1)
synchronization points if (CNTMIN = 1)

Fig 88. Synchronization Points

24.8.12.4 MOD register synchronization

The MOD register synchronization updates the MOD register with its buffer value. This synchronization is enabled if (FTMEN = 1).

The MOD register synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, it is expected that the MOD register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the MOD register synchronization depends on SWWRBUF, SWRSTCNT, HWWRBUF, and HWRSTCNT bits according to this flowchart:

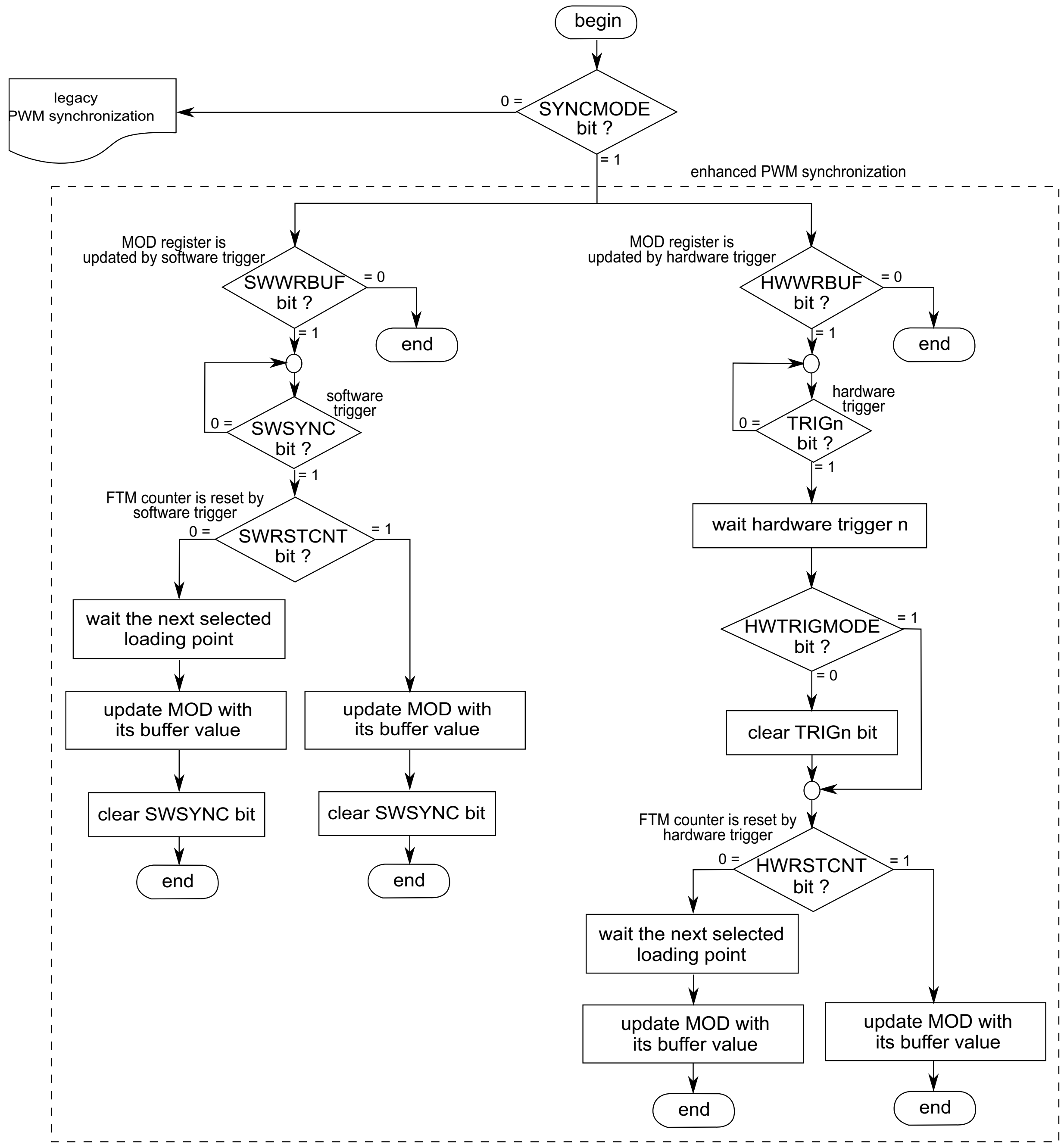

begin
SYNCMODE bit ?
0 =
legacy PWM synchronization
= 1
enhanced PWM synchronization
MOD register is updated by software trigger
SWWRBUF bit ?
= 0
end
= 1
software trigger
0 =
SWSYNC bit ?
= 1
FTM counter is reset by software trigger
0 =
SWRSTCNT bit ?
= 1
wait the next selected loading point
update MOD with its buffer value
clear SWSYNC bit
end
update MOD with its buffer value
clear SWSYNC bit
end
MOD register is updated by hardware trigger
HWWRBUF bit ?
= 0
end
= 1
hardware trigger
0 =
TRIGn bit ?
= 1
wait hardware trigger n
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
FTM counter is reset by hardware trigger
0 =
HWRSTCNT bit ?
= 1
wait the next selected loading point
update MOD with its buffer value
end
update MOD with its buffer value
end

Fig 89. MOD register synchronization flowchart

In the case of legacy PWM synchronization, the MOD register synchronization depends on PWMSYNC and REINIT bits according to the following description.

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 0), then this synchronization is made on the next selected loading point after an enabled trigger event takes place. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the trigger enable bit (TRIGn) is cleared according to "Hardware trigger". Examples with software and hardware triggers follow.

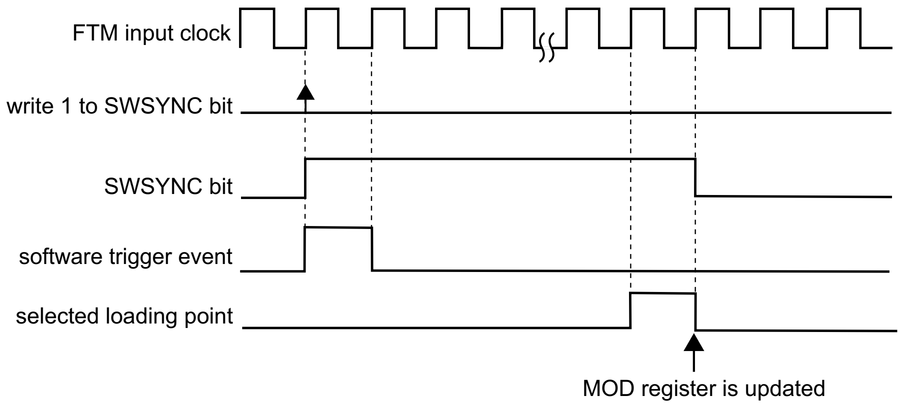


Fig 90. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 0), and software trigger was used

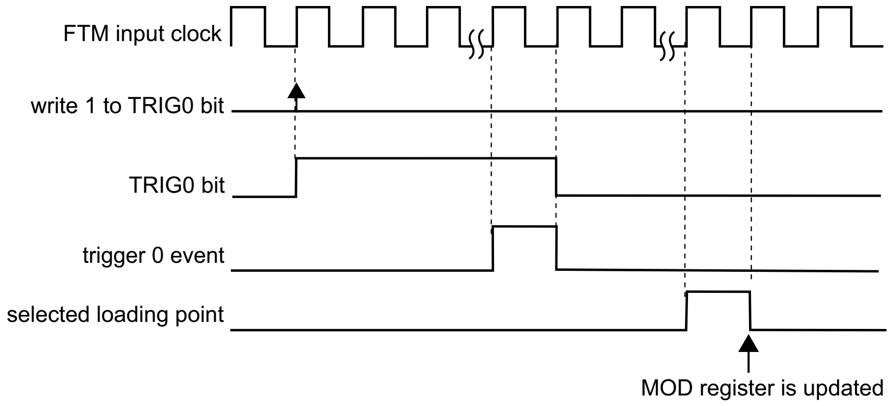


Fig 91. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 0), and hardware trigger was used

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 1), then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to ”Hardware trigger”. Examples with software and hardware triggers follow.

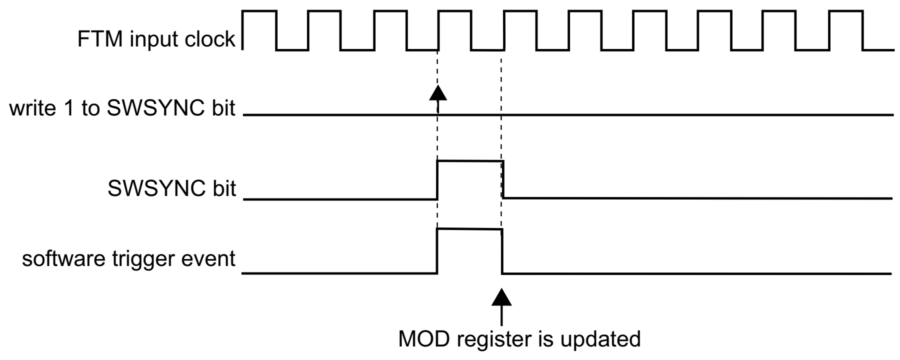


Fig 92. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 1), and software trigger was used

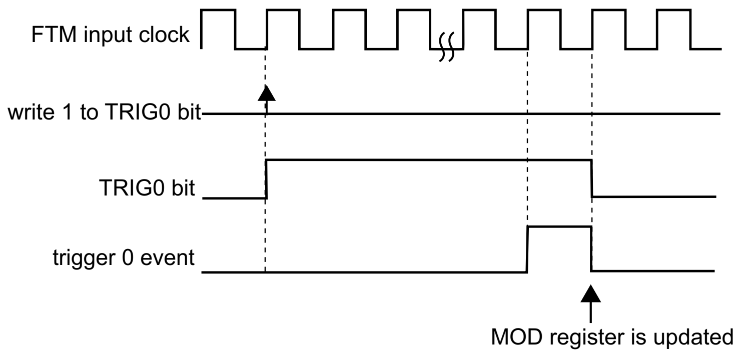


Fig 93. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 1), and hardware trigger was used

If (SYNCMODE = 0) and (PWMSYNC = 1), then this synchronization is made on the next selected loading point after the software trigger event takes place. The SWSYNC bit is cleared on the next selected loading point:

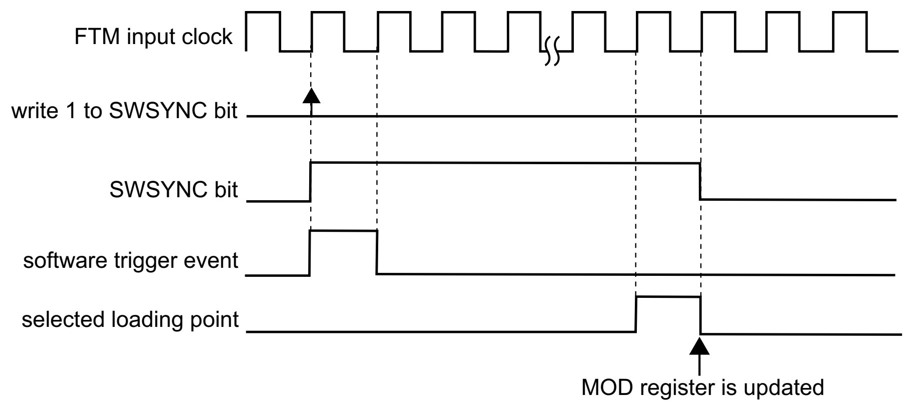


Fig 94. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 1)

24.8.12.5 CNTIN register synchronization

The CNTIN register synchronization updates the CNTIN register with its buffer value.

This synchronization is enabled if (FTMEN = 1), (SYNCMODE = 1), and (CNTINC = 1). The CNTIN register synchronization can be done only by the enhanced PWM synchronization (SYNCMODE = 1). The synchronization mechanism is the same as the MOD register synchronization done by the enhanced PWM synchronization; see ”MOD register synchronization”.

24.8.12.6 C(n)V and C(n+1)V register synchronization

The C(n)V and C(n+1)V registers synchronization updates the C(n)V and C(n+1)V registers with their buffer values.

This synchronization is enabled if (FTMEN = 1) and (SYNCEN = 1). The synchronization mechanism is the same as the ”MOD register synchronization”. However, it is expected that the C(n)V and C(n+1)V registers be synchronized only by the enhanced PWM synchronization (SYNCMODE = 1).

24.8.12.7 OUTMASK register synchronization

The OUTMASK register synchronization updates the OUTMASK register with its buffer value.

The OUTMASK register can be updated at each rising edge of FTM input clock (SYNCHOM = 0), by the enhanced PWM synchronization (SYNCHOM = 1 and SYNCMODE = 1) or by the legacy PWM synchronization (SYNCHOM = 1 and SYNCMODE = 0). However, it is expected that the OUTMASK register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the OUTMASK register synchronization depends on SWOM and HWOM bits. See the following flowchart:

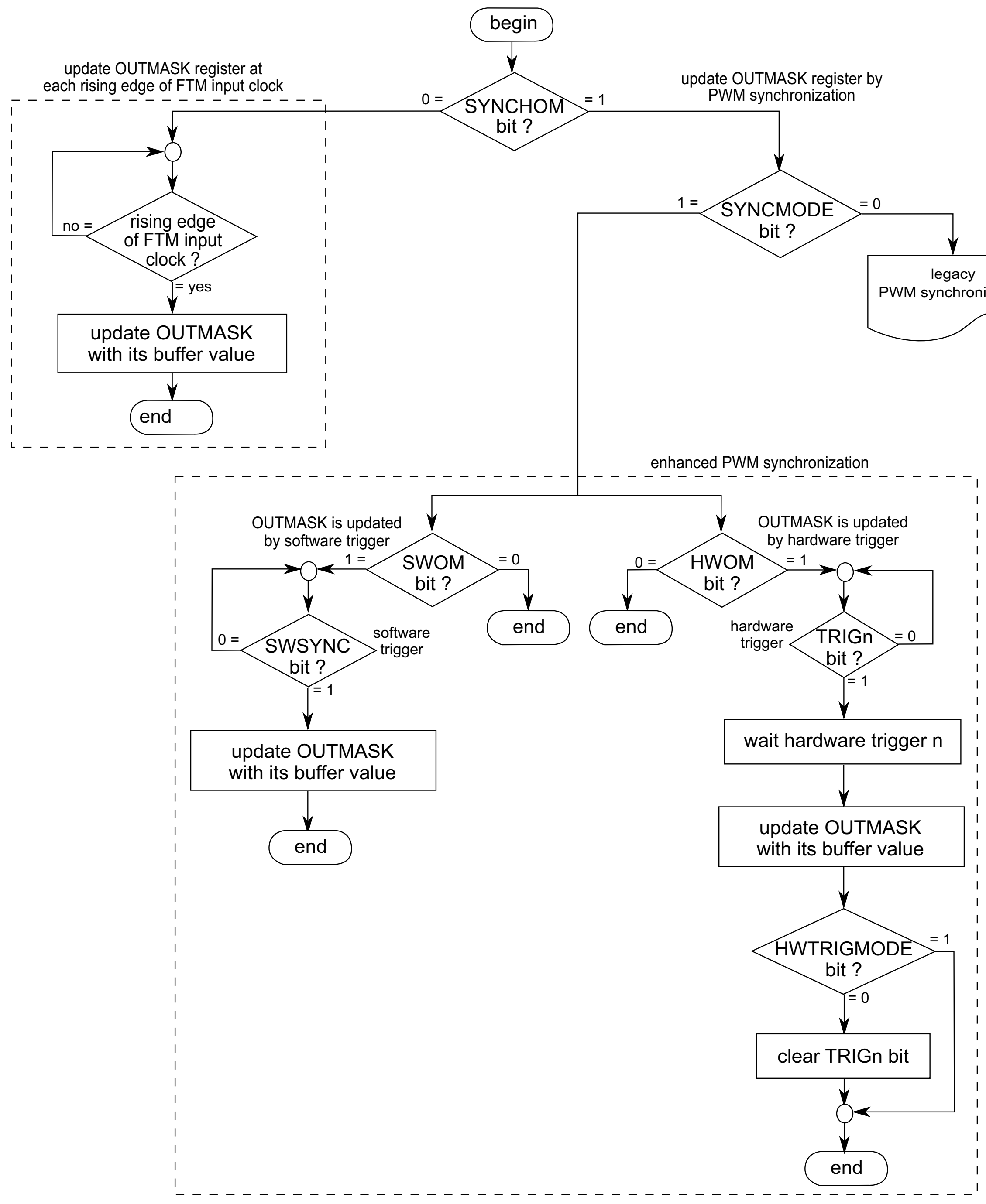

begin
SYNCHOM bit ?
0 =
= 1
update OUTMASK register at each rising edge of FTM input clock
rising edge of FTM input clock ?
no =
= yes
update OUTMASK with its buffer value
end
update OUTMASK register by PWM synchronization
SYNCMODE bit ?
1 =
= 0
legacy PWM synchronization
enhanced PWM synchronization
OUTMASK is updated by software trigger
SWOM bit ?
1 =
= 0
end
SWSYNC bit ?
0 =
= 1
software trigger
update OUTMASK with its buffer value
end
OUTMASK is updated by hardware trigger
HWOM bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update OUTMASK with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Fig 95. OUTMASK register synchronization flowchart

In the case of legacy PWM synchronization, the OUTMASK register synchronization depends on PWMSYNC bit according to the following description.

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 0), then this synchronization is done on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to ”Hardware trigger”. Examples with software and hardware triggers follow.

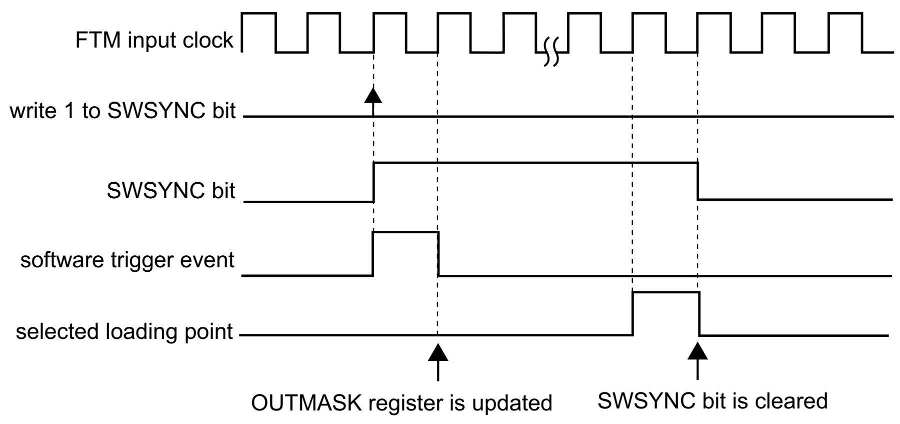


Fig 96. OUTMASK synchronization with (SYNCMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and software trigger was used

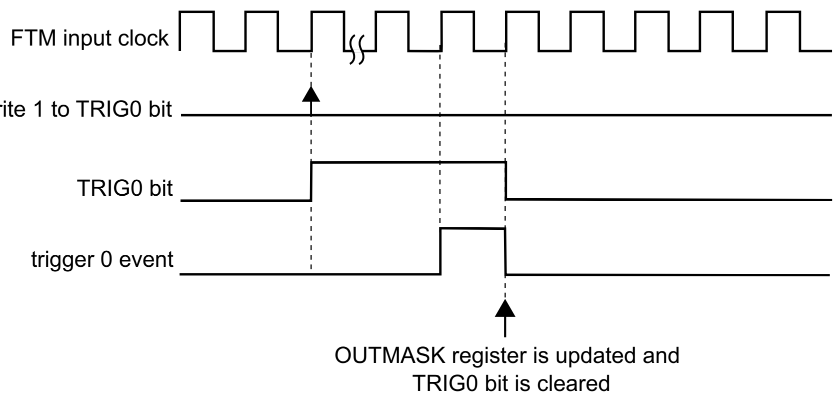


Fig 97. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and hardware trigger was used

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 1), then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to ”Hardware trigger”. An example with a hardware trigger follows.

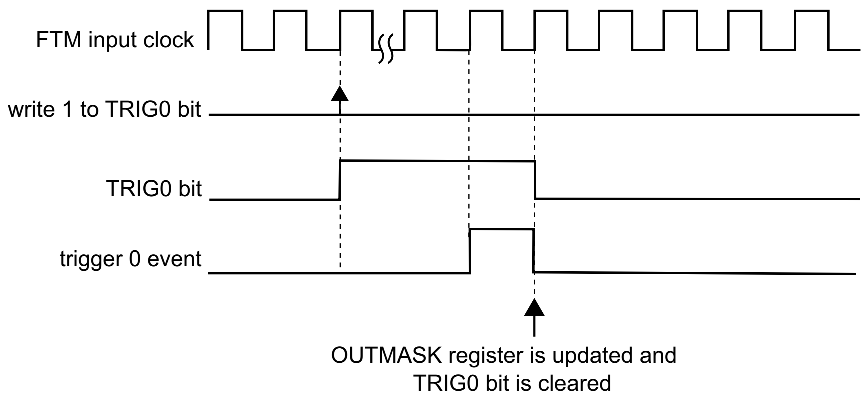


Fig 98. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 1), and hardware trigger was used

24.8.12.8 INVCTRL register synchronization

The INVCTRL register synchronization updates the INVCTRL register with its buffer value.

The INVCTRL register can be updated at each rising edge of FTM input clock (INVC = 0) or by the enhanced PWM synchronization (INVC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the INVCTRL register synchronization depends on SWINVC and HWINVC bits.

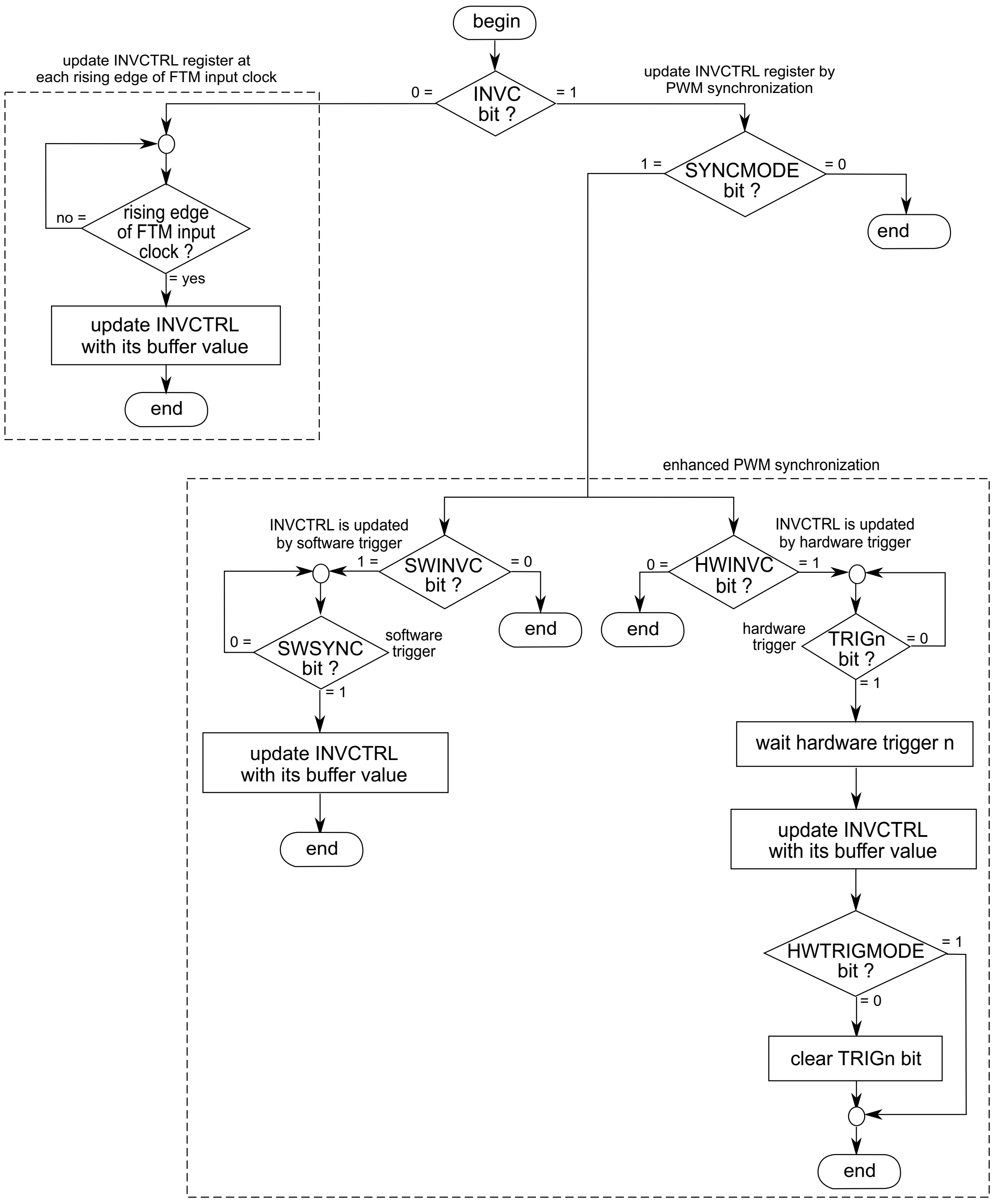

begin
INVC bit ?
0 =
= 1
update INVCTRL register at each rising edge of FTM input clock
update INVCTRL register by PWM synchronization
rising edge of FTM input clock ?
no =
= yes
update INVCTRL with its buffer value
end
SYNCMODE bit ?
1 =
= 0
end
enhanced PWM synchronization
INVCTRL is updated by software trigger
SWINVC bit ?
1 =
= 0
end
SWSYNC bit ?
0 =
software trigger
= 1
update INVCTRL with its buffer value
end
INVCTRL is updated by hardware trigger
HWINVC bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update INVCTRL with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Fig 99. INVCTRL register synchronization flowchart

24.8.12.9 SWOCTRL register synchronization

The SWOCTRL register synchronization updates the SWOCTRL register with its buffer value.

The SWOCTRL register can be updated at each rising edge of FTM input clock (SWOC = 0) or by the enhanced PWM synchronization (SWOC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the SWOCTRL register synchronization depends on SWSOC and HWSOC bits.

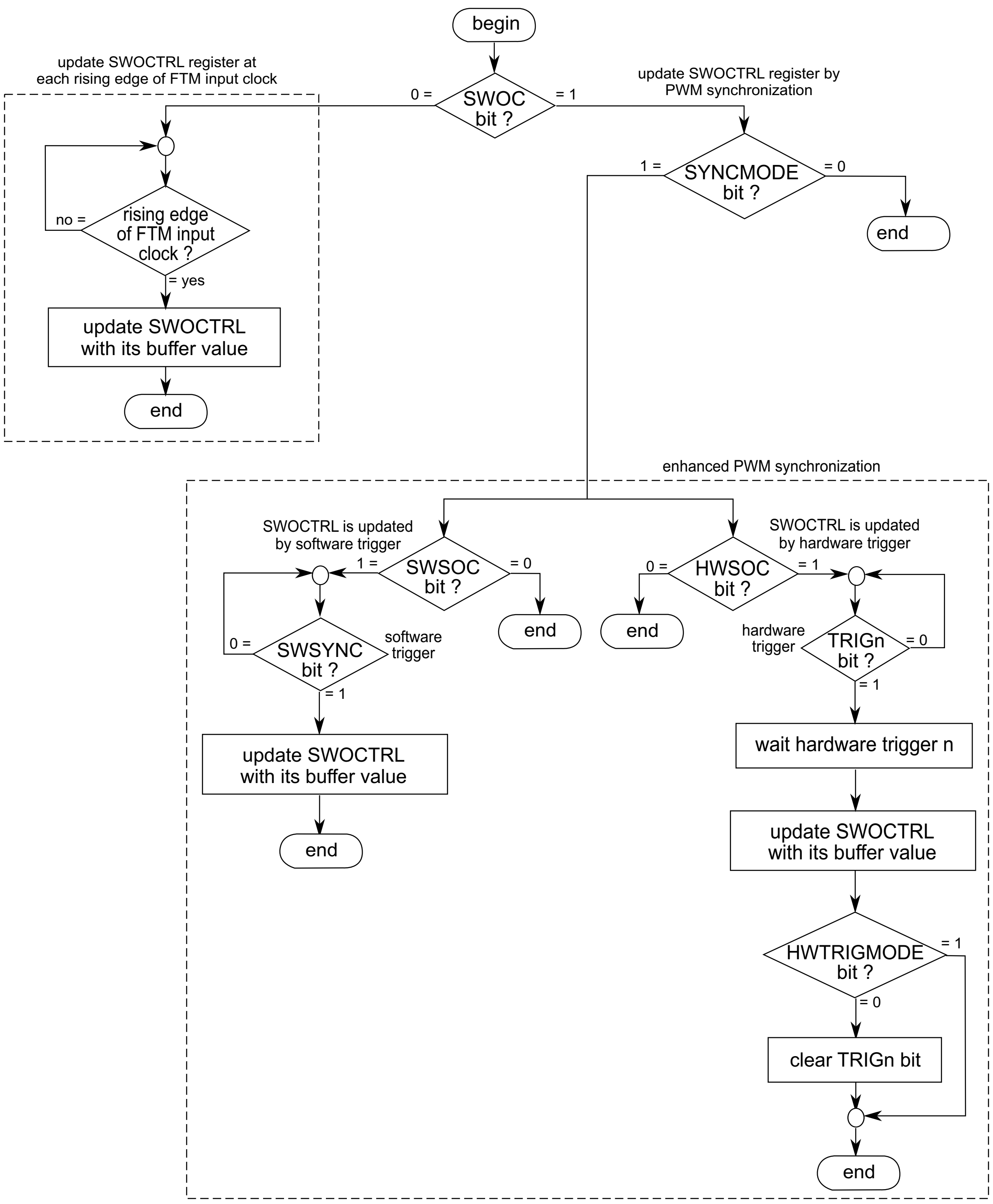

begin
SWOC bit ?
0 =
= 1
update SWOCTRL register at each rising edge of FTM input clock
update SWOCTRL register by PWM synchronization
rising edge of FTM input clock ?
no =
= yes
update SWOCTRL with its buffer value
end
SYNCMODE bit ?
1 =
= 0
end
enhanced PWM synchronization
SWOCTRL is updated by software trigger
SWSOC bit ?
1 =
= 0
end
SWSYNC bit ?
0 =
= 1
software trigger
update SWOCTRL with its buffer value
end
SWOCTRL is updated by hardware trigger
HWSOC bit ?
0 =
= 1
end
TRIGn bit ?
hardware trigger
= 0
= 1
wait hardware trigger n
update SWOCTRL with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Fig 100. SWOCTRL register synchronization flowchart

24.8.12.10 FTM counter synchronization

The FTM counter synchronization is a mechanism that allows the FTM to restart the

PWM generation at a certain point in the PWM period. The channels outputs are forced to their initial value, except for channels in Output Compare mode, and the FTM counter is forced to its initial counting value defined by CNTIN register.

The following figure shows the FTM counter synchronization. Note that after the synchronization event occurs, the channel (n) is set to its initial value and the channel (n +1) is not set to its initial value due to a specific timing of this figure in which the deadtime insertion prevents this channel output from transitioning to 1. If no deadtime insertion is selected, then the channel (n+1) transitions to logical value 1 immediately after the synchronization event occurs.

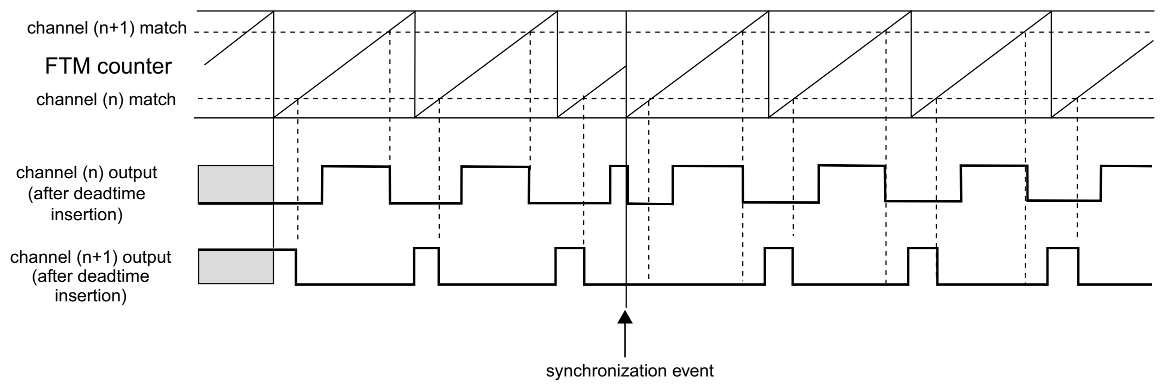

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)
synchronization event

Fig 101. FTM counter synchronization

The FTM counter synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, the FTM counter must be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the FTM counter synchronization depends on SWRSTCNT and HWRSTCNT bits according to the following flowchart.

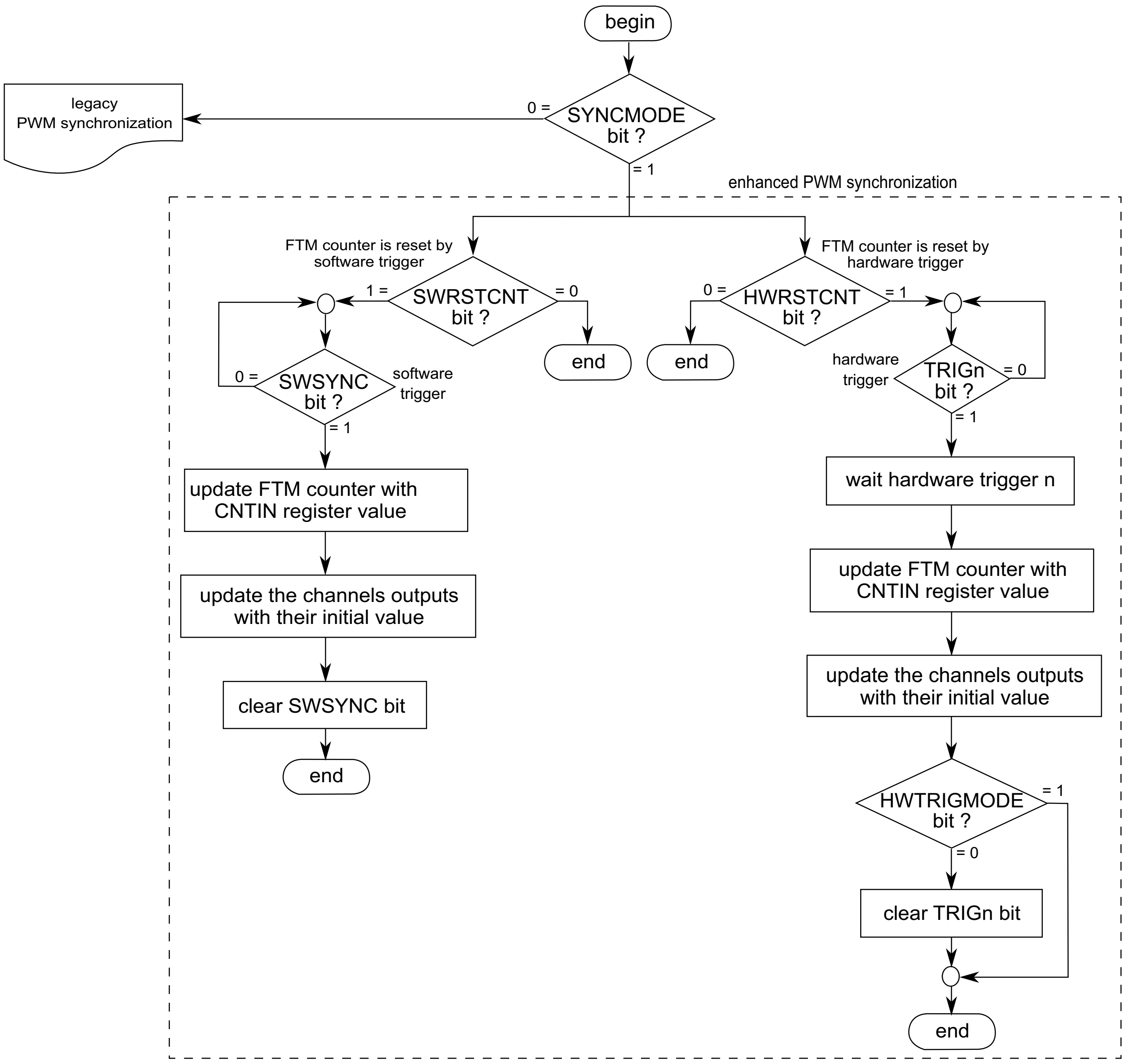

begin
SYNCMODE bit ?
0 =
legacy PWM synchronization
= 1
enhanced PWM synchronization
FTM counter is reset by software trigger
SWRSTCNT bit ?
1 =
= 0
end
SWSYNC bit ?
0 =
software trigger
= 1
update FTM counter with CNTIN register value
update the channels outputs with their initial value
clear SWSYNC bit
end
FTM counter is reset by hardware trigger
HWRSTCNT bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update FTM counter with CNTIN register value
update the channels outputs with their initial value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Fig 102. FTM counter synchronization flowchart

In the case of legacy PWM synchronization, the FTM counter synchronization depends on REINIT and PWMSYNC bits according to the following description.

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 0) then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger then the TRIGn bit is cleared according to "Hardware trigger". Examples with software and hardware triggers follow.

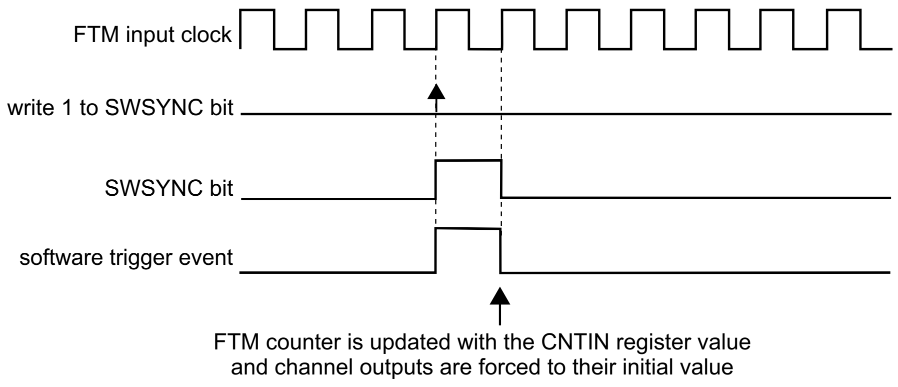

FTM input clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Fig 103. FTM counter synchronization with (SYNCMODE = 0), (REINIT = 1), (PWMSYNC = 0), and software trigger was used

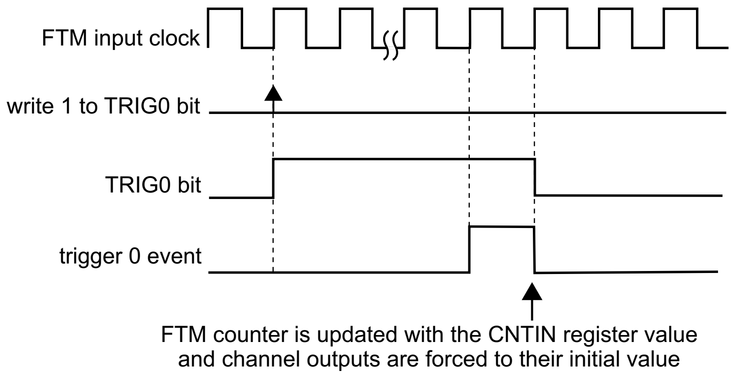

FTM input clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Fig 104. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 0), and hardware trigger was used

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 1) then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to "Hardware trigger".

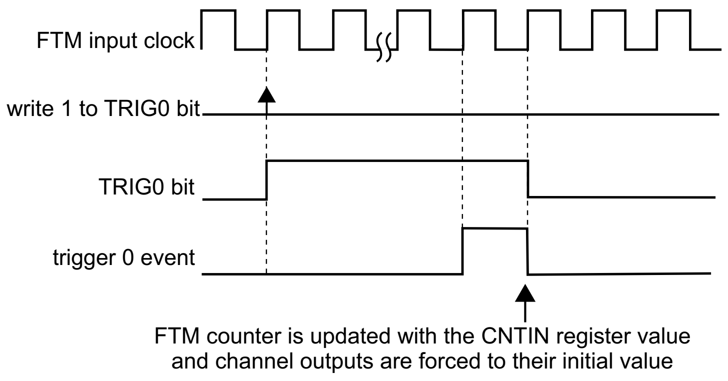

FTM input clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Fig 105. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 1), and hardware trigger was used

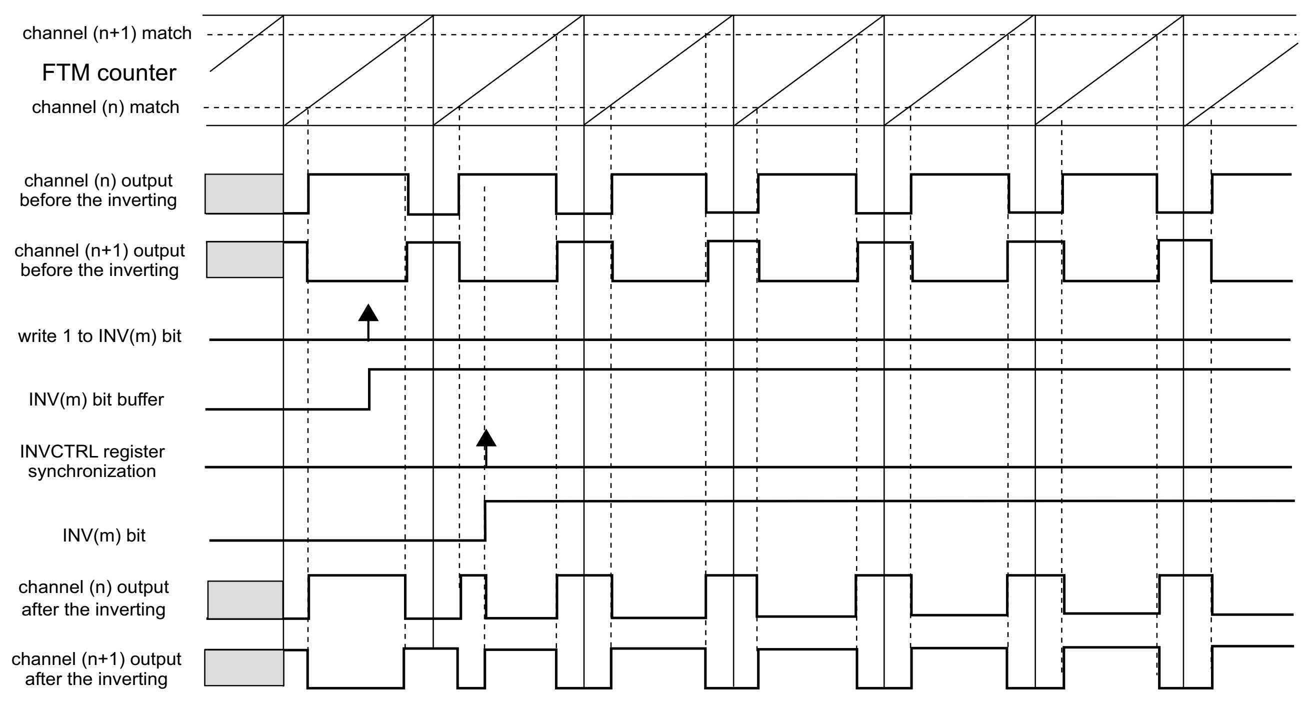
24.8.13 Inverting

The invert functionality swaps the signals between channel (n) and channel (n+1) outputs. The inverting operation is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMP = 1, and
- INVm = 1 (where m represents a channel pair)

The INVm bit in INVCTRL register is updated with its buffer value according to ”INVCTRL register synchronization”.

In combine mode with channel (n) ELSB:ELSA = 1:0, the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN), forced high at the channel (n) match and forced low at the channel (n+1) match. If the inverting is selected, the channel (n) output behavior is changed to force high at the beginning of the PWM period, force low at the channel (n) match and force high at the channel (n+1) match. See the following figure.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output before the inverting
channel (n+1) output before the inverting
write 1 to INV(m) bit
INV(m) bit buffer
INVCTRL register synchronization
INV(m) bit
channel (n) output after the inverting
channel (n+1) output after the inverting

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Fig 106. Channels (n) and (n+1) outputs after the inverting in combine mode with channel (n) ELSB:ELSA = 1:0

Note that the channel (n) ELSB:ELSA bits should be considered because they define the active state of the channels outputs. In combine mode with channel (n) ELSB:ELSA = X:1, the channel (n) output is forced high at the beginning of the period, forced low at the channel (n) match and forced high at the channel (n+1) match. When inverting is selected, the channels (n) and (n+1) present waveforms as shown in the following figure.

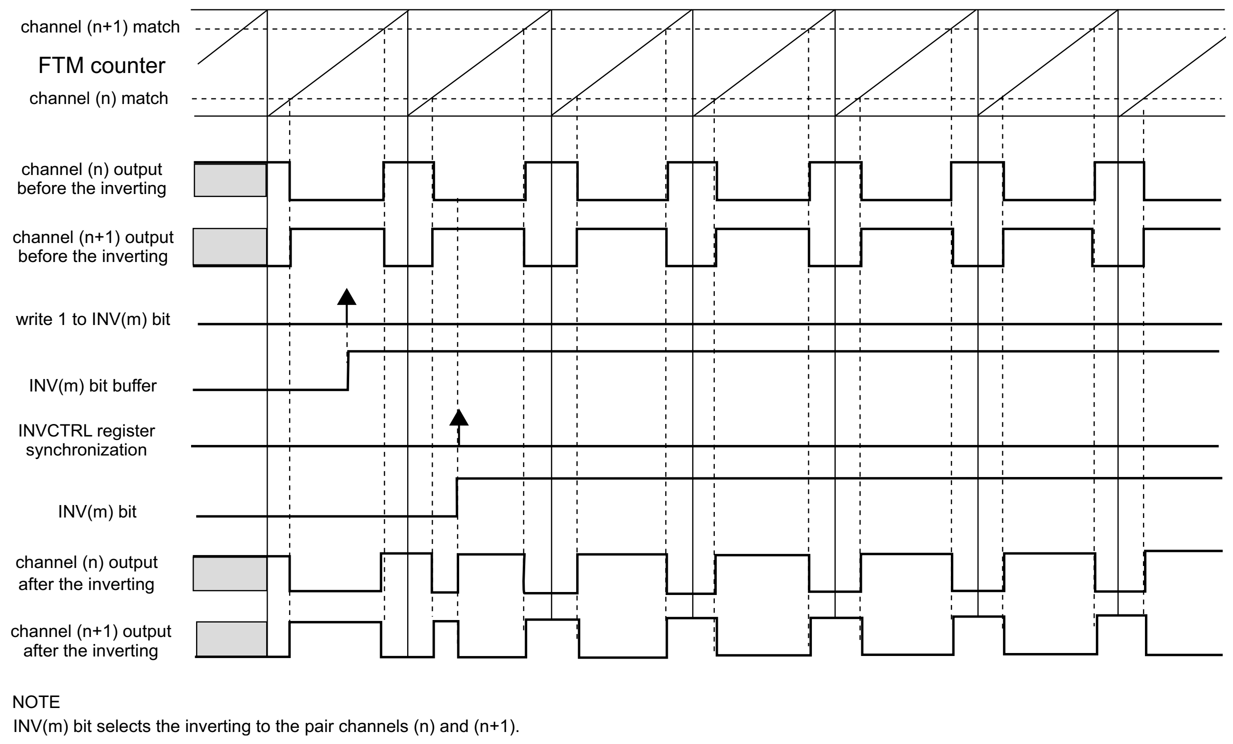

channel (n+1) match
FTM counter
channel (n) match
channel (n) output before the inverting
channel (n+1) output before the inverting
write 1 to INV(m) bit
INV(m) bit buffer
INVCTRL register synchronization
INV(m) bit
channel (n) output after the inverting
channel (n+1) output after the inverting
NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Fig 107. Channels (n) and (n+1) outputs after the inverting in combine mode with channel (n) ELSB:ELSA = X:1

Note: The Inverting is not available in ”Output Compare mode”.

24.8.14 Software Output Control Mode

The software output control forces the channel output according to software defined values at a specific time in the PWM generation.

The software output control is selected when:

- QUADEN = 0
- DECAPEN = 0, and
- CH(n)OC = 1

The CH(n)OC bit enables the software output control for a specific channel output and the CH(n)OCV selects the value that is forced to this channel output.

Both CH(n)OC and CH(n)OCV bits in SWOCTRL register are buffered and updated with their buffer value according to ”SWOCTRL register synchronization”.

The following figure shows the channels (n) and (n+1) outputs signals when the software output control is used. In this case the channels (n) and (n+1) are set to Combine and Complementary mode.

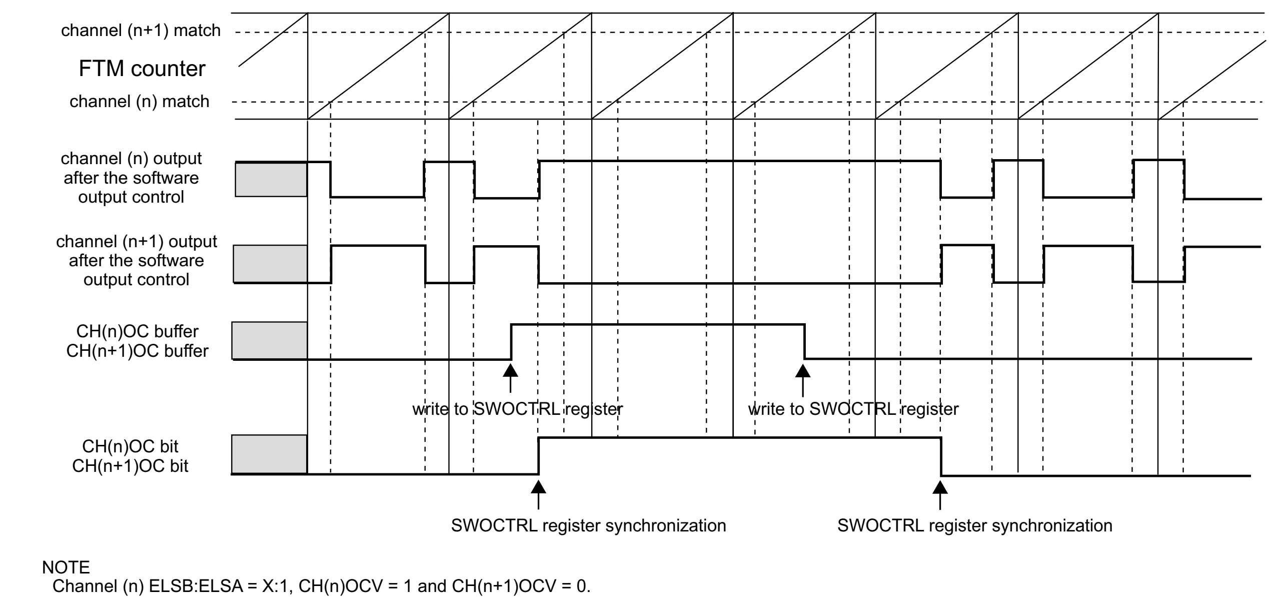

channel (n+1) match
FTM counter
channel (n) match
channel (n) output after the software output control
channel (n+1) output after the software output control
CH(n)OC buffer CH(n+1)OC buffer
write to SWOCTRL register
write to SWOCTRL register
CH(n)OC bit CH(n+1)OC bit
SWOCTRL register synchronization
SWOCTRL register synchronization
NOTE
Channel (n) ELSB:ELSA = X:1, CH(n)OCV = 1 and CH(n+1)OCV = 0.

Fig 108. Example of software output control in Combine and Complementary mode

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is zero.

Table 426. Software ouput control behavior when (COMP = 0)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to one

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is one.

Table 427. Software ouput control behavior when (COMP = 1)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to zero

Note:

- The CH(n)OC and CH(n+1)OC bits should be equal.
- The COMP bit must not be modified when software output control is enabled, that is, CH(n)OC = 1 and/or CH(n+1)OC = 1.

- Software output control has the same behavior with disabled or enabled FTM counter (see the CLKS field description in the Status and Control register).

24.8.15 Deadtime insertion

The deadtime insertion is enabled when (DTEN = 1) and DTVAL[5:0] is non- zero.

DEADTIME register defines the deadtime delay that can be used for all FTM channels. The clock for the DEADTIME delay is the FTM input clock divided by DTPS bits, and the DTVAL[5:0] bits define the deadtime modulo, that is, the number of the deadtime prescaler clocks.

The deadtime delay insertion ensures that no two complementary signals (channels (n) and (n+1)) drive the active state at the same time.

If POL(n) = 0, POL(n+1) = 0, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the low value until the end of the deadtime delay when the channel (n) output is set. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the low value until the end of the deadtime delay when the channel (n+1) output is set. See the following figures.

If POL(n) = 1, POL(n+1) = 1, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the high value until the end of the deadtime delay when the channel (n) output is cleared. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the high value until the end of the deadtime delay when the channel (n+1) output is cleared.

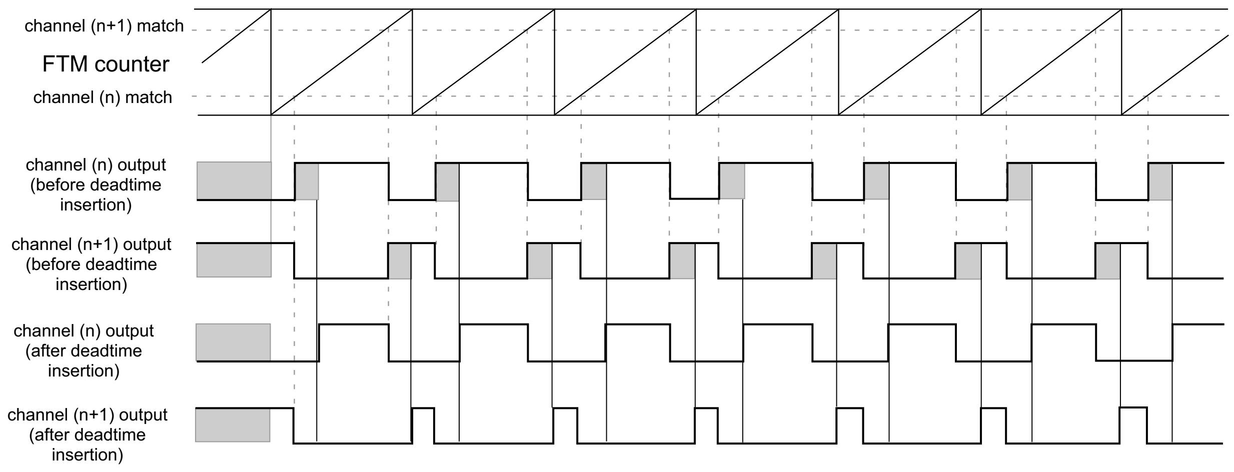

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Fig 109. Deadtime insertion with ELSB:ELSA = 1:0, POL(n) = 0, and POL(n+1) = 0

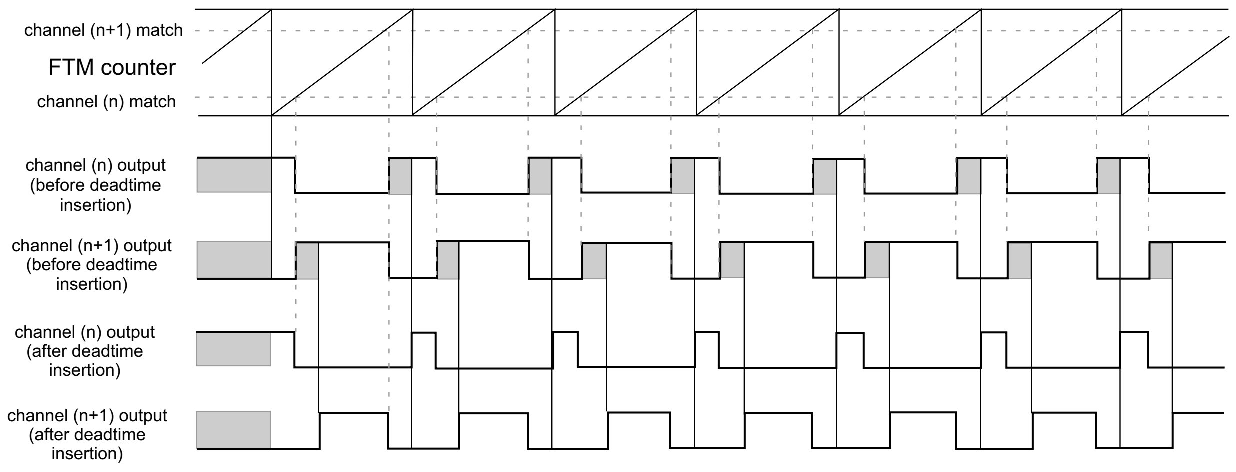

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Fig 110. Deadtime insertion with ELSB:ELSA = X:1, POL(n) = 0, and POL(n+1) = 0

Note:

- The deadtime feature must be used only in Complementary mode.
- The deadtime feature is not available in Output Compare mode.

24.8.15.1 Deadtime insertion corner cases

If (PS[2:0] is cleared), (DTPS[1:0] = 0:0 or DTPS[1:0] = 0:1):

- and the deadtime delay is greater than or equal to the channel (n) duty cycle ((C(n+1)V – C(n)V) × FTM input clock), then the channel (n) output is always the inactive value (POL(n) bit value).
- and the deadtime delay is greater than or equal to the channel (n+1) duty cycle ((MOD – CNTIN + 1 – (C(n+1)V – C(n)V)) × FTM input clock), then the channel (n+1) output is always the inactive value (POL(n+1) bit value).

Although, in most cases the deadtime delay is not comparable to channels (n) and (n+1) duty cycle, the following figures show examples where the deadtime delay is comparable to the duty cycle.

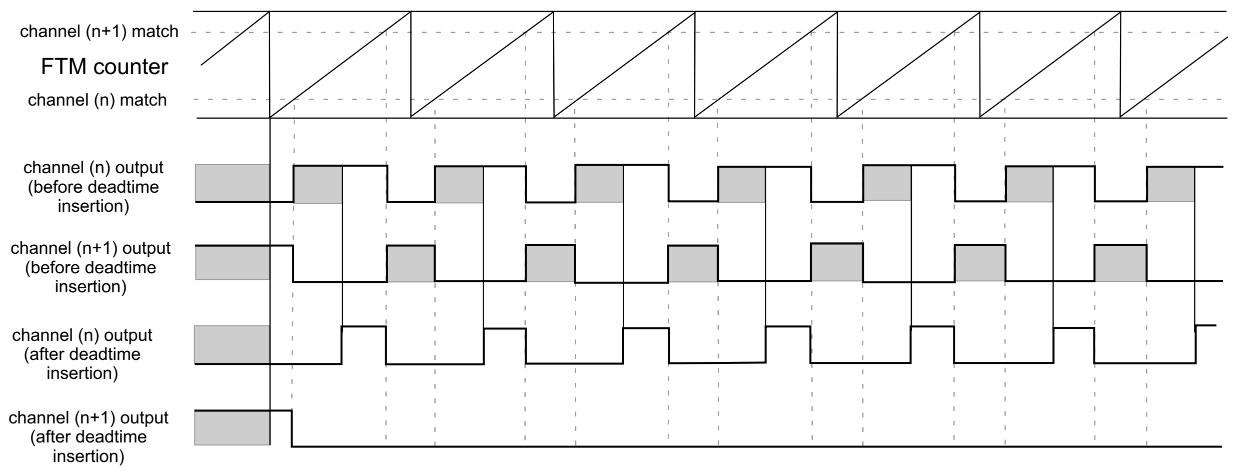

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Fig 111. Example of the deadtime insertion (channel (n) ELSB:ELSA = 1:0, POL(n) = 0, and POL(n+1) = 0 when the deadtime delay is comparable to channel (n+1) duty cycle

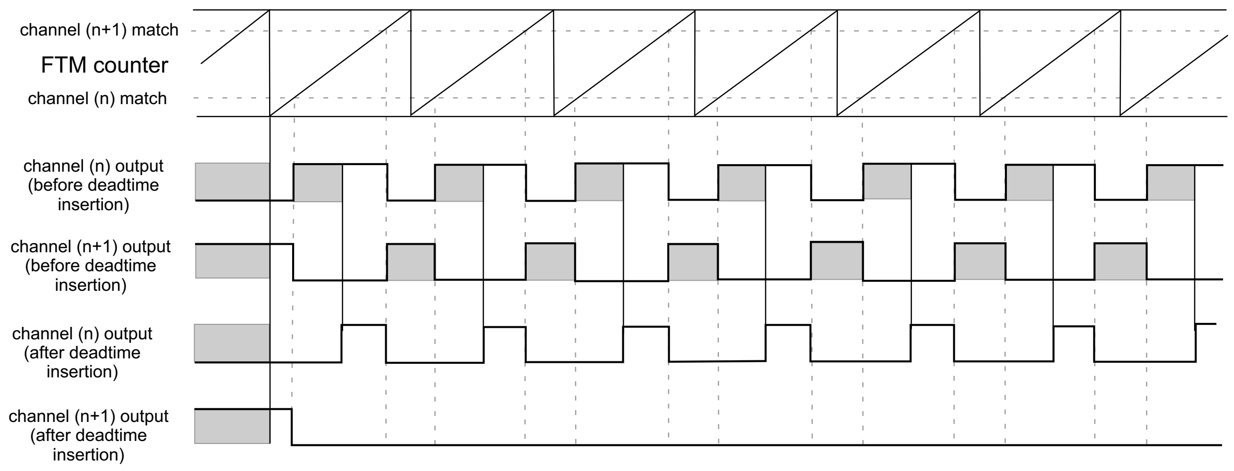

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Fig 112. Example of the deadtime insertion (channel (n) ELSB:ELSA = 1:0, POL(n) = 0, and POL(n+1) = 0 when the deadtime delay is comparable to channels (n) and (n+1) duty cycle

24.8.16 Output mask

The output mask can be used to force channels output to their inactive state through software. For example: to control a BLDC motor.

Any write to the OUTMASK register updates its write buffer. The OUTMASK register is updated with its buffer value by PWM synchronization; see ”OUTMASK register synchronization”.

If CH(n)OM = 1, then the channel (n) output is forced to its inactive state (POLn bit value). If CH(n)OM = 0, then the channel (n) output is unaffected by the output mask. See the following figure.

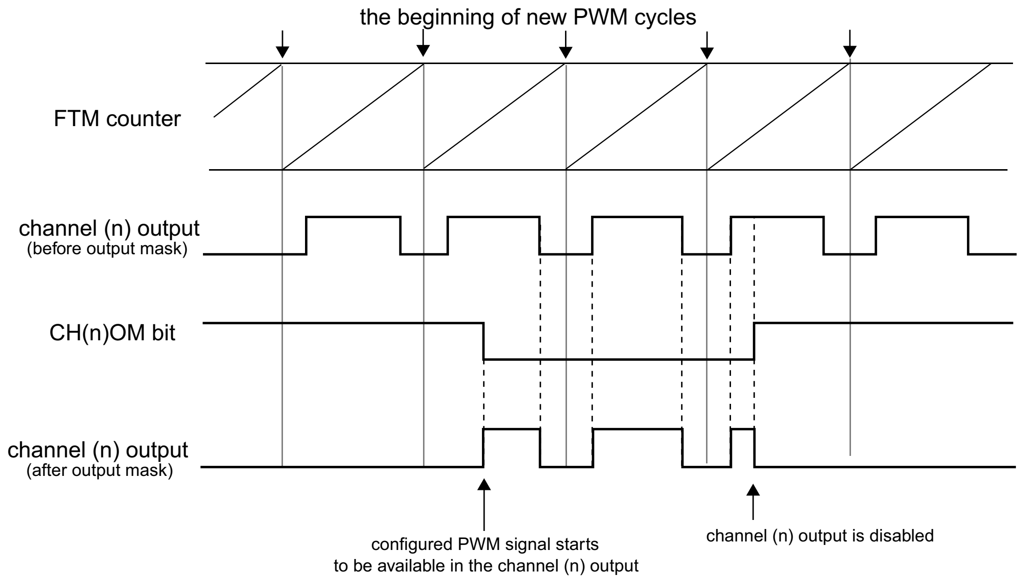

the beginning of new PWM cycles
FTM counter
channel (n) output
(before output mask)
CH(n)OM bit
channel (n) output
(after output mask)
configured PWM signal starts
to be available in the channel (n) output
channel (n) output is disabled

Fig 113. Output mask with POLn = 0

The following table shows the output mask result before the polarity control.

Table 428. Output mask result for channel (n) before the polarity control

	Output Mask Input	Output Mask Result
0	inactive state	inactive state
	active state	active state
1	inactive state	inactive state
	active state	

24.8.17 Fault Control

The fault control is enabled if FAULTM[1:0] ≠ 0:0.

FTM can have up to four fault inputs. FAULTnEN bit (where n = 0, 1, 2, 3) enables the fault input n and FFLTRnEN bit enables the fault input n filter. FFVAL[3:0] bits select the value of the enabled filter in each enabled fault input.

The fault input after being synchronized by FTM input clock is the filter input.

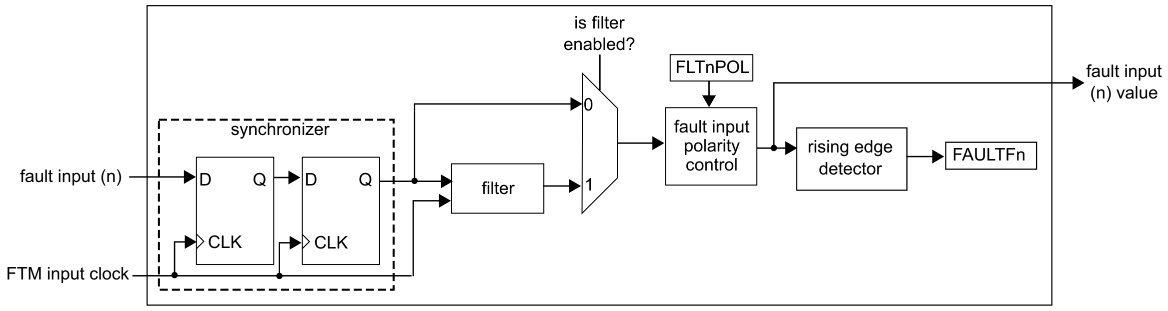


Fig 114. Diagram for Fault Control

When there is a state change in the fault input (n), the counter is reset and starts counting up. As long as the new state is stable on the fault input (n), the counter continues to increment. When the counter is equal to FFVAL[3:0] bits, the new fault input (n) value is validated. It is then transmitted as a pulse to the edge detector.

If the opposite edge appears on the fault input (n) before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. If a pulse is sampled as a value less than FFVAL[3:0] consecutive rising edges of FTM input clock, it is regarded as a glitch and is not passed on to the edge detector.

The table below shows the delay that is added by the FTM fault input filter according to its configuration.

Table 429. FTM Fault Input Filter Delay

FTM fault input filter	Number of rising edges between the selected edge on fault input and forcing the channels outputs to their safe values
• fault input does not have the input filter, or • fault input filter is disabled (FFLTRnEN = 0 or FFVAL[3:0] = 0)	• 3 rising edges of FTM input clock
• fault input has the input filter, and • fault input filter is enabled (FFLTRnEN = 1 and FFVAL[3:0] ≠ 0)	• (4 + FFVAL[3:0]) rising edges of FTM input clock

If the fault control and fault input (n) are enabled, and the selected edge at the fault input (n) is detected, then a fault condition has occurred and the FAULTFn bit is set. The FAULTF bit is the logic OR of FAULTFn[3:0] bits. See the following figure.

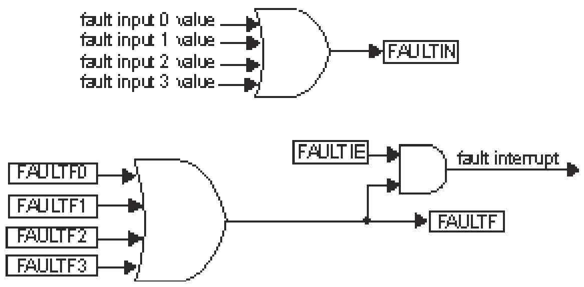


Fig 115. FAULTF and FAULTIN bits and FAULT interrupt

If the fault control is enabled (FAULTM[1:0] ,?0:0), a fault condition has occurred, and (FAULTENj = 1, where j is the pair j of the channels), then the channels (n) and (n+1) outputs are forced to their safe values:

- channel (n) output takes the POL(n) bit value
- channel (n+1) output takes the POL(n+1) bit value

The fault interrupt is generated when (FAULTF = 1) and (FAULTIE = 1). This interrupt request remains set until:

- Software clears the FAULTF bit by reading FAULTF bit as 1 and writing 0 to it
- Software clears the FAULTIE bit
- A reset occurs

24.8.17.1 Automatic fault clearing

If the automatic fault clearing is selected (FAULTM[1:0] = 1:1), then the channels output disabled by fault control is again enabled when the fault input signal (FAULTIN) returns to zero and a new PWM cycle begins. See the following figure.

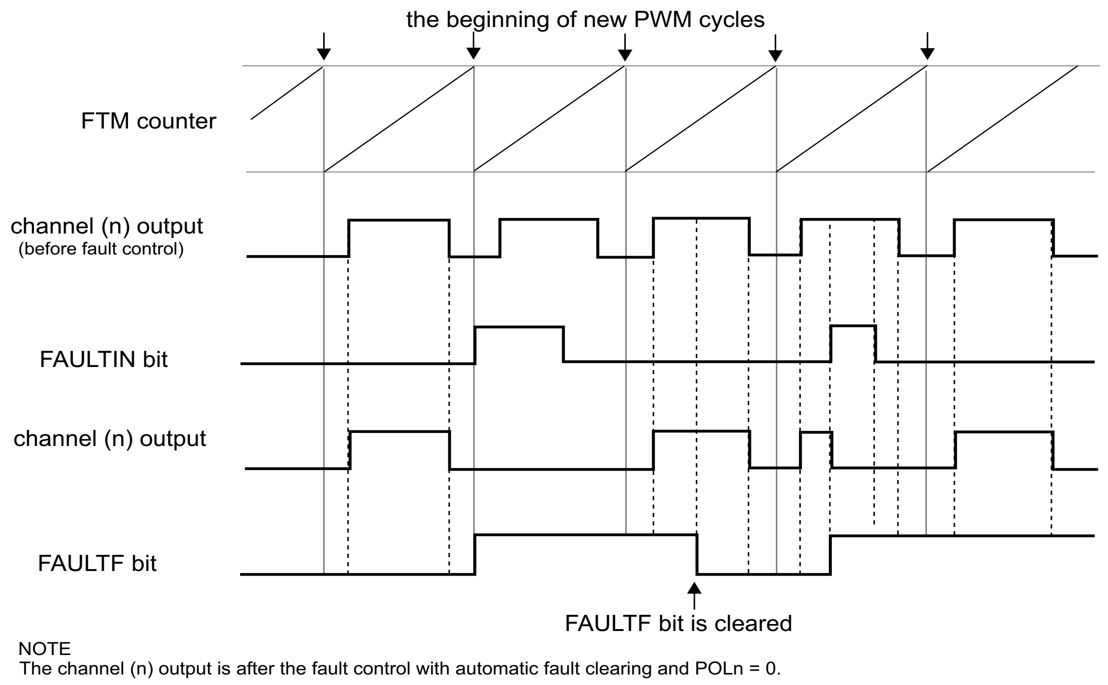
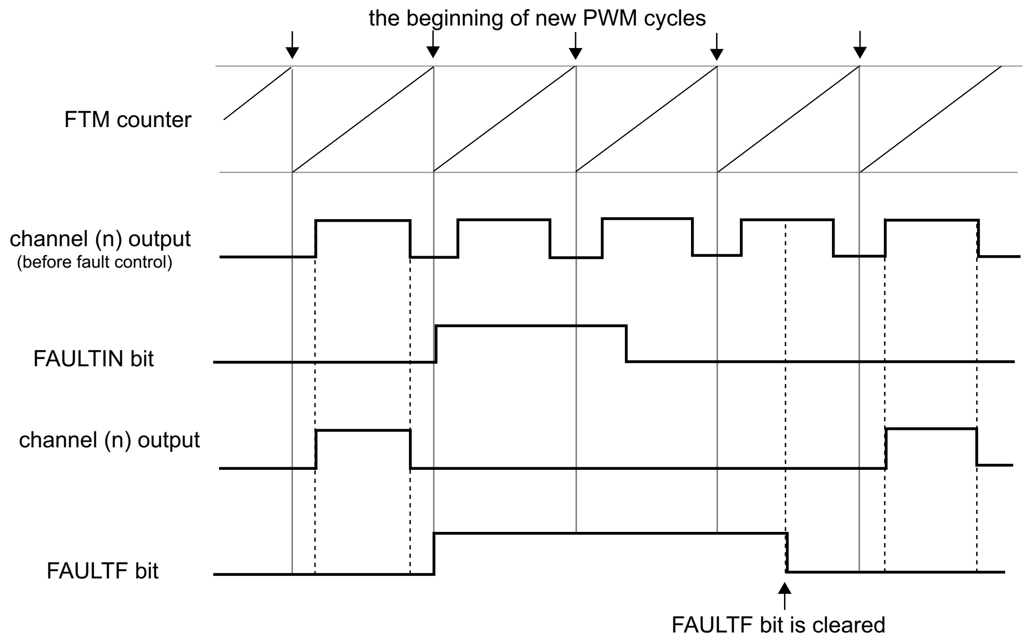

the beginning of new PWM cycles
FTM counter
channel (n) output
(before fault control)
FAULTIN bit
channel (n) output
FAULTF bit
FAULTF bit is cleared
NOTE
The channel (n) output is after the fault control with automatic fault clearing and POLn = 0.

Fig 116. Fault control with automatic fault clearing

24.8.17.2 Manual fault clearing

If the manual fault clearing is selected (FAULTM[1:0] = 0:1 or 1:0), then the channels output disabled by fault control is again enabled when the FAULTF bit is cleared and a new PWM cycle begins. See the following figure.



NOTE
The channel (n) output is after the fault control with manual fault clearing and POLn = 0.

Fig 117. Fault control with manual fault clearing

24.8.17.3 Fault inputs polarity control

The FLTjPOL bit selects the fault input j polarity, where j = 0, 1, 2, 3:

- If FLTjPOL = 0, the fault j input polarity is high, so the logical one at the fault input j indicates a fault.
- If FLTjPOL = 1, the fault j input polarity is low, so the logical zero at the fault input j indicates a fault.

24.8.18 Polarity Control

The POLn bit selects the channel (n) output polarity:

- If POLn = 0, the channel (n) output polarity is high, so the logical one is the active state and the logical zero is the inactive state.
- If POLn = 1, the channel (n) output polarity is low, so the logical zero is the active state and the logical one is the inactive state.

24.8.19 Initialization

The initialization forces the CH(n)OI bit value to the channel (n) output when 1 is written to the INIT bit.

The initialization depends on COMP and DTEN bits. The following table shows the values that channels (n) and (n+1) are forced by initialization when the COMP and DTEN bits are zero.

Table 430. Initialization behavior when (COMP = 0 and DTEN = 0)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	0	is forced to zero	is forced to zero
0	1	is forced to zero	is forced to one
1	0	is forced to one	is forced to zero
1	1	is forced to one	is forced to one

The following table shows the values that channels (n) and (n+1) are forced by initialization when (COMP = 1) or (DTEN = 1).

Table 431. Initialization behavior when (COMP = 1 and DTEN = 1)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	X	is forced to zero	is forced to one
1	X	is forced to one	is forced to zero

Note: The initialization feature must be used only with disabled FTM counter. See the description of the CLKS field in the Status and Control register.

24.8.20 Features Priority

The following figure shows the priority of the features used at the generation of channels (n) and (n+1) outputs signals.

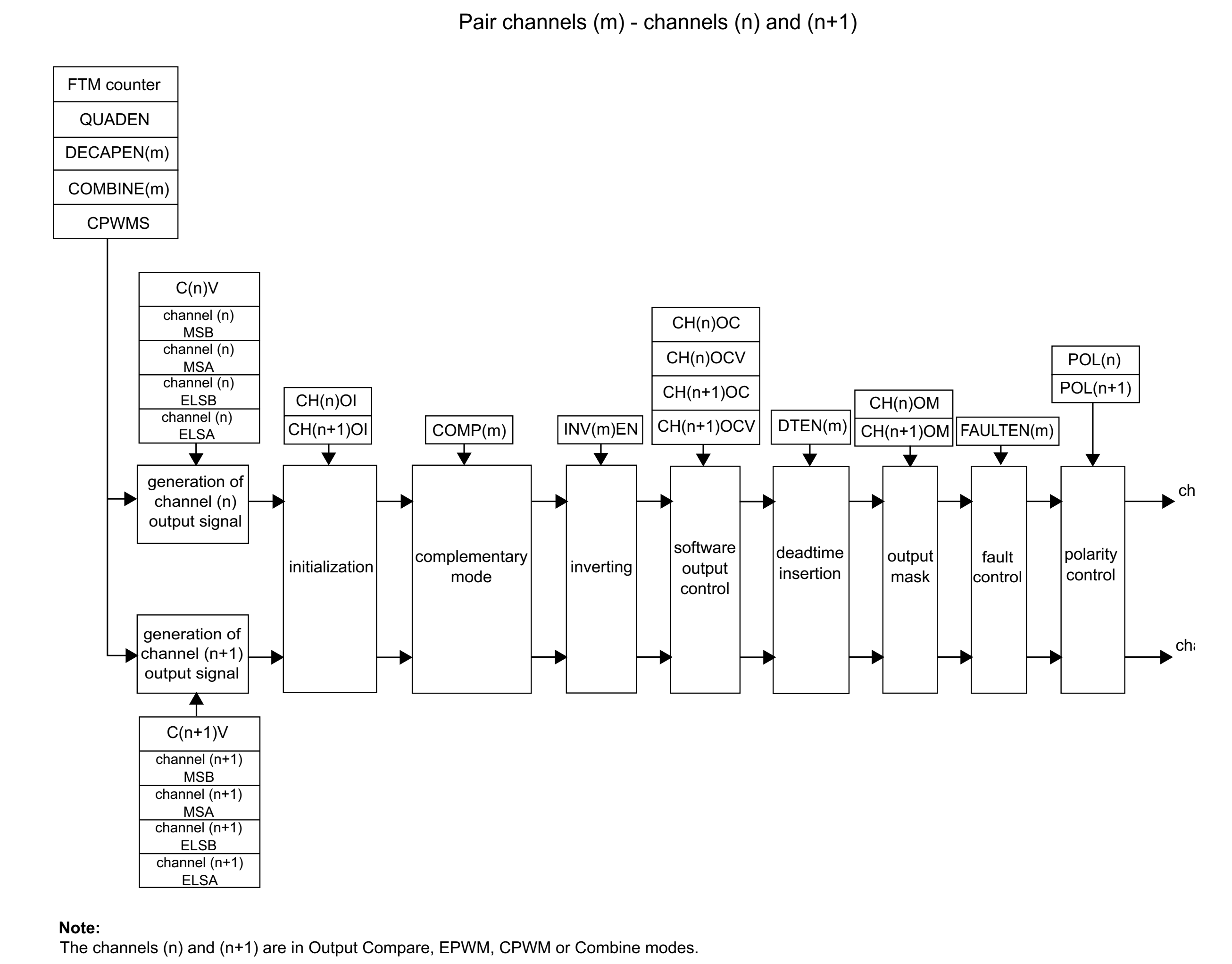


Fig 118. Priority of the features used at the generation of channels (n) and (n+1) output

Note: The ”Initialization” must not be used with ”Inverting”and ”Software Output Control Mode”.

24.8.21 External Trigger

If the CH(n)TRIG bit (register EXTTRIG) is set, where n = 0, 1, 2, 3, 4, 5, 6 or 7, then the FTM generates a trigger when the channel (n) match occurs (FTM counter = C(n)V) at the FTM external trigger output.

The width of a channel (n) trigger is one FTM input clock and the FTM is able to generate multiple triggers in one PWM period. See the figure below.

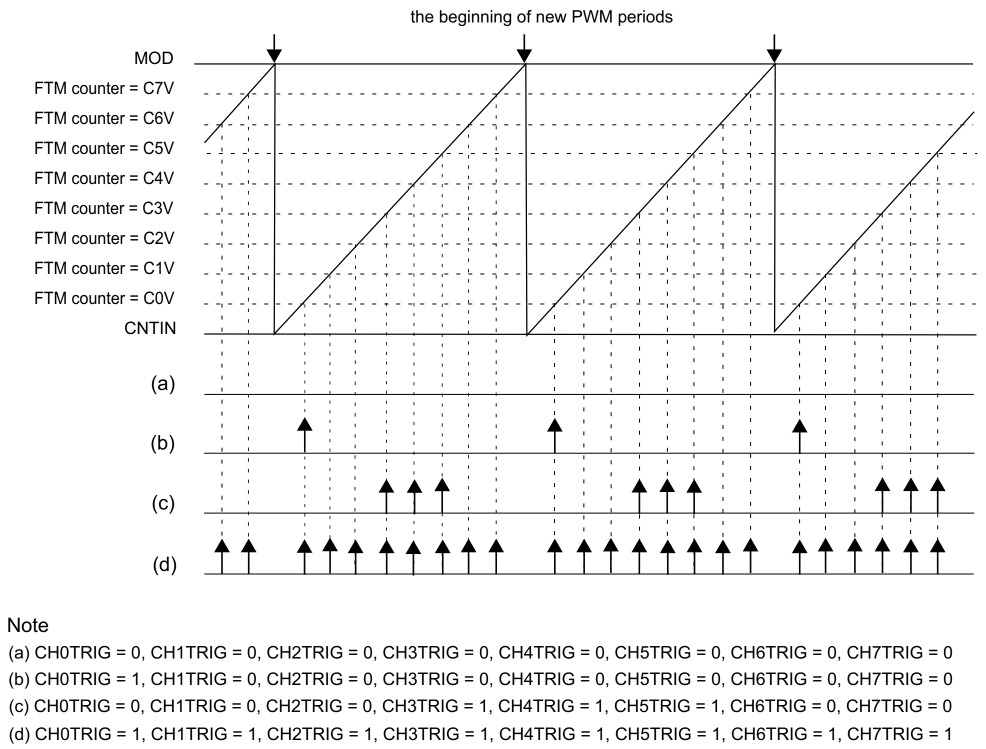

the beginning of new PWM periods
MOD
FTM counter = C7V
FTM counter = C6V
FTM counter = C5V
FTM counter = C4V
FTM counter = C3V
FTM counter = C2V
FTM counter = C1V
FTM counter = C0V
CNTIN
(a)
(b)
(c)
(d)
Note
(a) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0, CH6TRIG = 0, CH7TRIG = 0
(b) CH0TRIG = 1, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0, CH6TRIG = 0, CH7TRIG = 0
(c) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1, CH6TRIG = 0, CH7TRIG = 0
(d) CH0TRIG = 1, CH1TRIG = 1, CH2TRIG = 1, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1, CH6TRIG = 1, CH7TRIG = 1

Fig 119. External Trigger

24.8.22 Initialization Trigger

Initialization trigger allows FTM to generate a trigger in some specific points of FTM counter cycle. This feature is controlled by the bits INITTRIGEN and ITRIGR. The INITTRIGEN bit enables the initialization trigger generation and the ITRIGR bit selects when the initialization trigger is generated.

If INITTRIGEN = 1 and ITRIGR = 1, then the initialization trigger is generated when FTM counter reaches a reload point according to the frequency of the reload opportunities (”Reload Points”).

Note: For this configuration of initialization trigger and in CPWM mode, the bits CNTMAX and CNTMIN select where the initialization trigger is generated.

If INITTRIGEN = 1 and ITRIGR = 0, then the initialization trigger is generated when FTM counter is updated with the CNTIN register value. See the cases below.

1. When FTM counter is updated with CNTIN register value automatically.

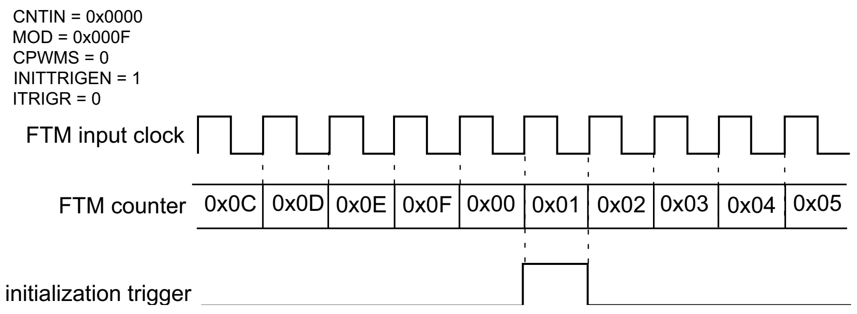


Fig 120. Example of the generation of the initialization trigger in the case 1.

2. When there is a write to CNT register.

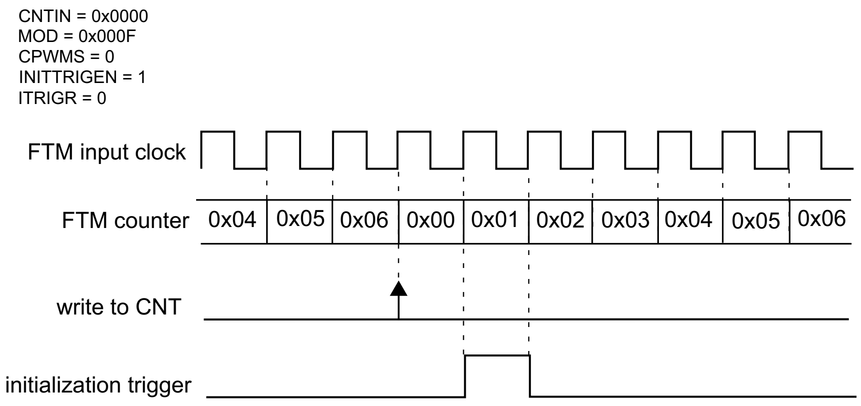


Fig 121. Example of the generation of the initialization trigger in the case 2.

Note: This behavior is not available in CPWM mode.

3. When there is the "FTM counter synchronization".

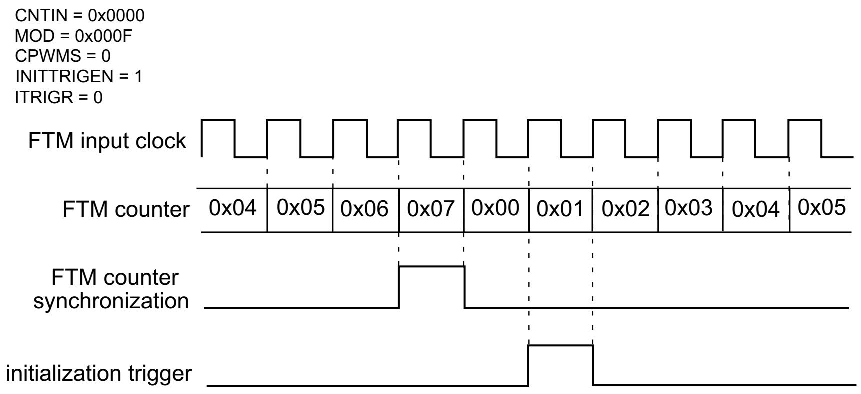


Fig 122. Example of the generation of the initialization trigger in the case 3.

Note: This behavior is not available in CPWM mode

4. If (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits.

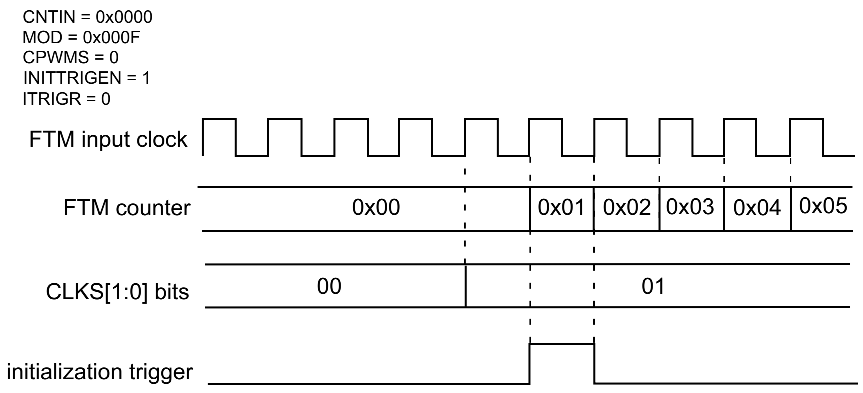


Fig 123. Example of the generation of the initialization trigger in the case 4.

Note: This behavior is not available in CPWM mode

5. If the channel (n) is in Input Capture mode, (ICRST = 1) and the selected input capture event occurs in the channel (n) input.

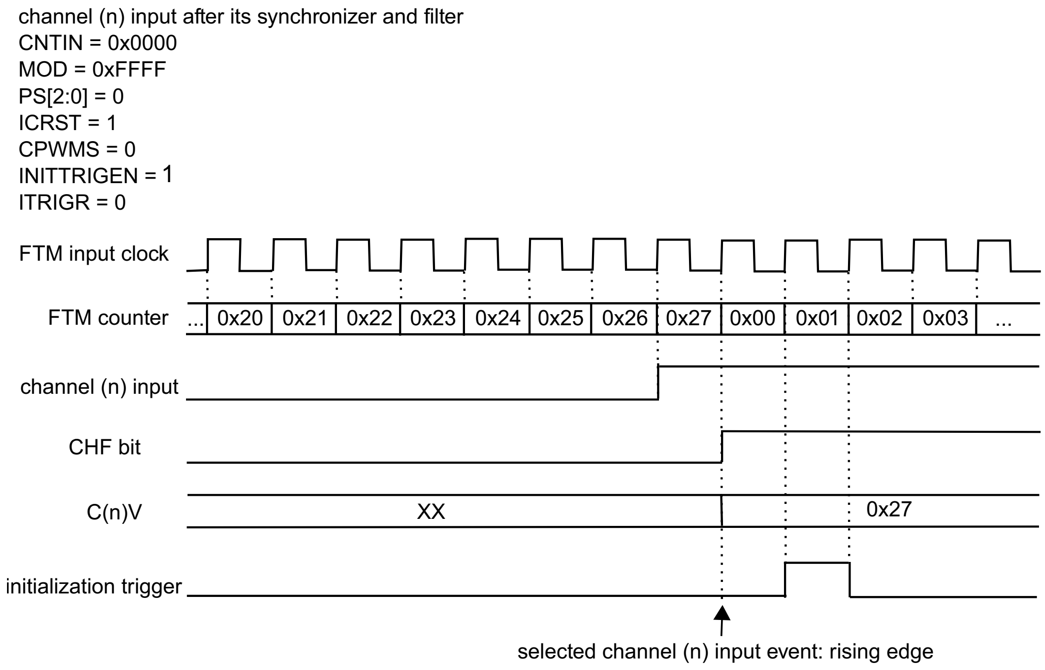


Fig 124. Example of the generation of the initialization trigger in the case 5.

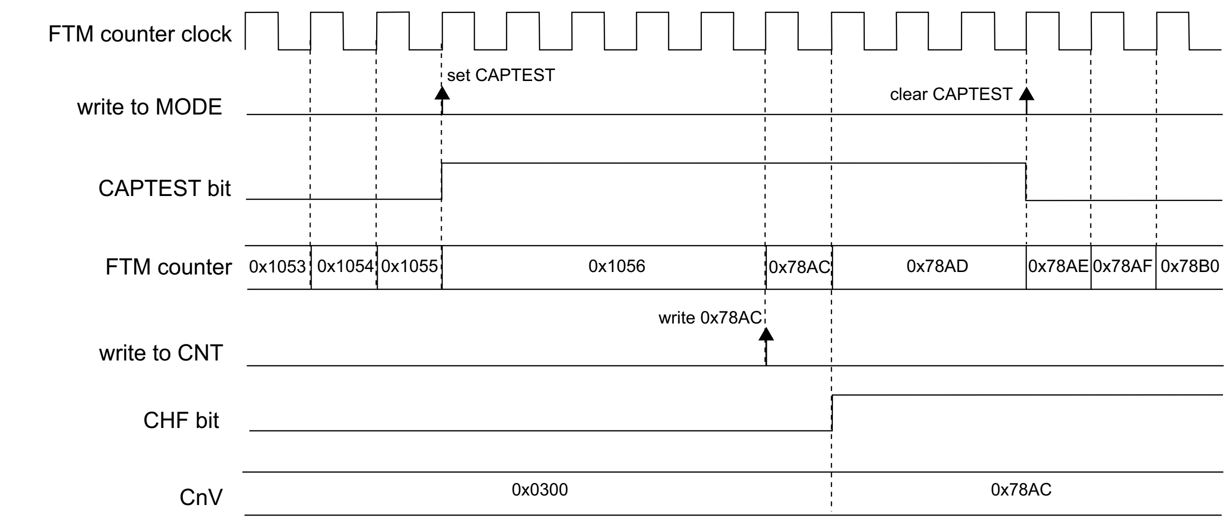
24.8.23 Capture Test Mode

The Capture Test mode allows to test the CnV registers, the FTM counter and the interconnection logic between the FTM counter and CnV registers.

In this test mode, all channels must be configured for "Input Capture mode" and FTM counter must be configured to the "Up counting".

When the Capture Test mode is enabled (CAPTEST = 1), the FTM counter is frozen and any write to CNT register updates directly the FTM counter; see the following figure. After it was written, all CnV registers are updated with the written value to CNT register and CHF bits are set. Therefore, the FTM counter is updated with its next value according to its configuration. Its next value depends on CNTIN, MOD, and the written value to FTM counter.

The next reads of CnV registers return the written value to the FTM counter and the next reads of CNT register return FTM counter next value.



Note:
- FTM counter is in free running
- FTMEN = 1
- FTM channel (n) is in Input Capture Mode

Fig 125. Capture Test Mode

24.8.24 Dual Edge Capture Mode

The dual edge capture mode is enabled if DECAPEN = 1. This mode allows to measure a pulse width or period of the channel (n) input where n = 0, 2, 4 or 6. The channel (n) filter can be enabled for n = 0 or 2.

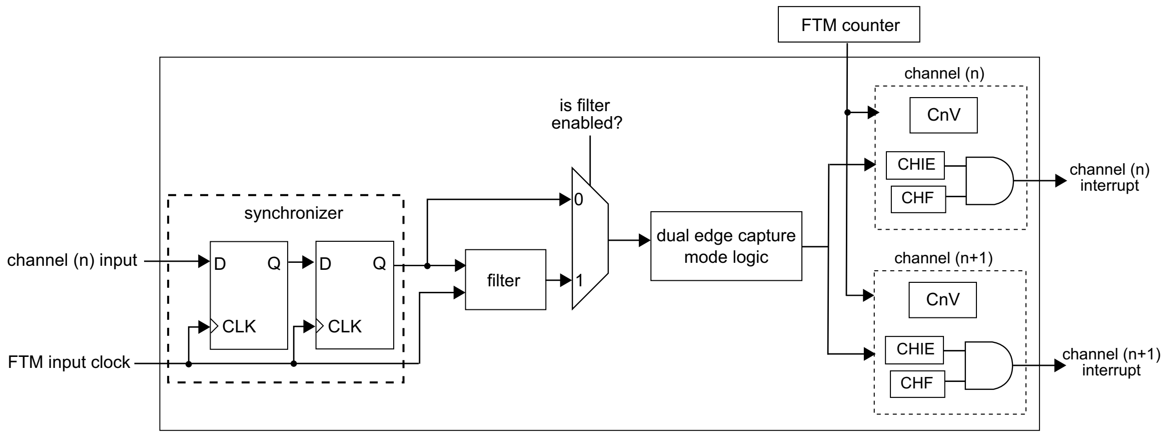


Fig 126. Diagram for Dual Edge Capture Mode

The channel (n) MSA bit defines if the dual edge capture mode is one-shot or continuous.

The channel (n) ELSB:ELSA bits select the edge that is captured by channel (n), and channel (n+1) ELSB:ELSA bits select the edge that is captured by channel (n+1). If both channel (n) ELSB:ELSA and channel (n+1) ELSB:ELSA bits select the same edge, then it is the period measurement. If these bits select different edges, then it is a pulse width measurement.

In the dual edge capture mode, only channel (n) input is used and channel (n+1) input is ignored.

If the selected edge by channel (n) bits is detected at channel (n) input, then channel (n) CHF bit is set and the channel (n) interrupt is generated (if channel (n) CHIE = 1). If the selected edge by channel (n+1) bits is detected at channel (n) input and (channel (n) CHF = 1), then channel (n+1) CHF bit is set and the channel (n+1) interrupt is generated (if channel (n+1) CHIE = 1).

The C(n)V register stores the FTM counter value when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the FTM counter value when the selected edge by channel (n+1) is detected at channel (n) input.

In this mode, a coherency mechanism (for channels (n) and (n+1)) ensures coherent data when the C(n)V and C(n+1)V registers are read. The only requirement is that C(n)V must be read before C(n+1)V.

Note: The dual edge capture mode must be used with channel (n) ELSB:ELSA = 0:1 or 1:0, channel (n+1) ELSB:ELSA = 0:1 or 1:0 and the FTM counter in ”Free running counter”.

24.8.24.1 One-Shot Capture mode

The One-Shot Capture mode is selected when (DECAPEN = 1), and (channel (n) MSA = 0). In this capture mode, only one pair of edges at the channel (n) input is captured. The channel (n) ELSB:ELSA bits select the first edge to be captured, and channel (n+1) ELSB:ELSA bits select the second edge to be captured.

The edge captures are enabled while DECAP bit is set. For each new measurement in One-Shot Capture mode, first the channel (n) CHF and channel (n+1) CHF bits must be cleared, and then the DECAP bit must be set.

In this mode, the DECAP bit is automatically cleared by FTM when the edge selected by channel (n+1) is captured. Therefore, while DECAP bit is set, the one-shot capture is in process. When this bit is cleared, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

Similarly, when the channel (n+1) CHF bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

24.8.24.2 Continuous Capture mode

The Continuous Capture mode is selected when (DECAPEN = 1), and (channel (n) MSA = 1). In this capture mode, the edges at the channel (n) input are captured continuously. The channel (n) ELSB:ELSA bits select the initial edge to be captured, and channel (n+1) ELSB:ELSA bits select the final edge to be captured.

The edge captures are enabled while DECAP bit is set. For the initial use, first the channel (n) CHF and channel (n+1) CHF bits must be cleared, and then DECAP bit must be set to start the continuous measurements.

When the channel (n+1) CHF bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers. The latest captured values are always available in these registers even after the DECAP bit is cleared.

In this mode, it is possible to clear only the channel (n+1) CHF bit. Therefore, when the channel (n+1) CHF bit is set again, the latest captured values are available in C(n)V and C(n+1)V registers.

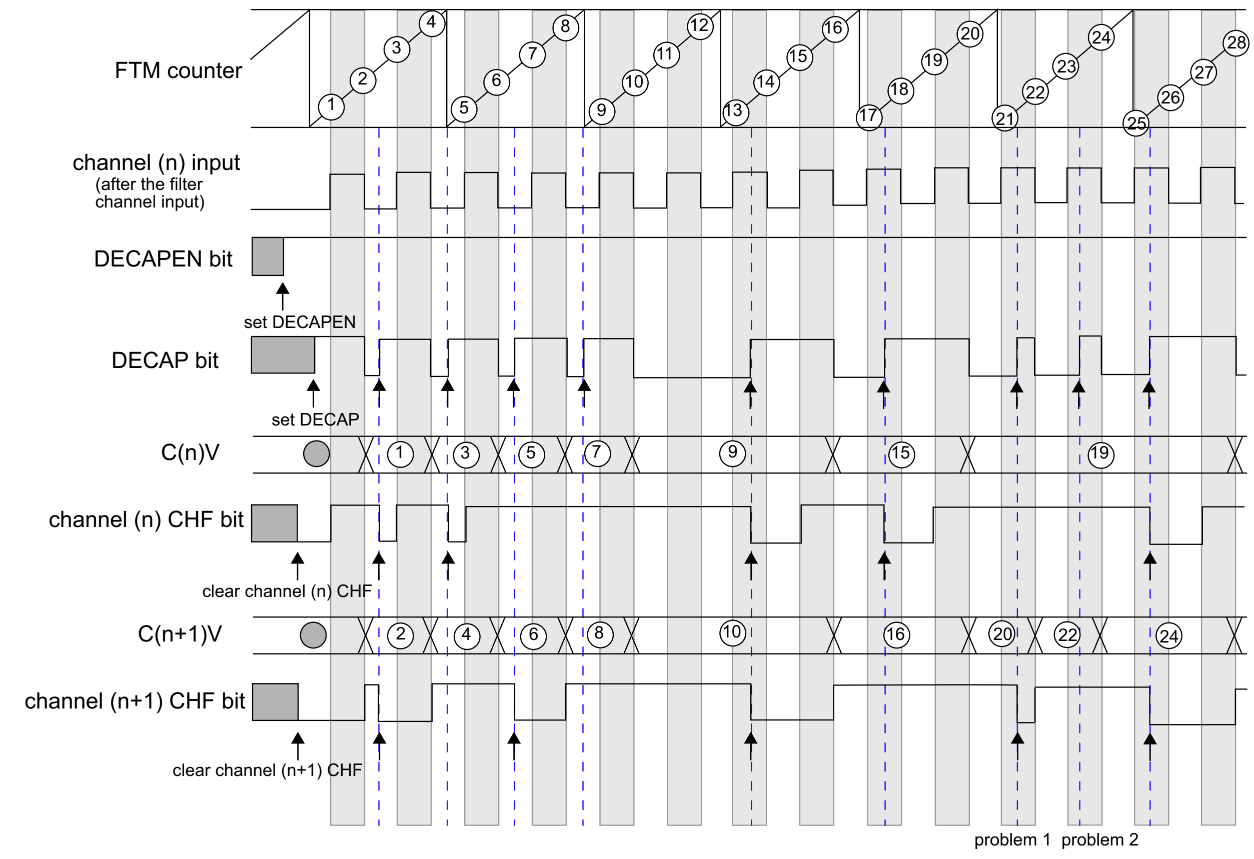
For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the channel (n) CHF and channel (n+1) CHF bits to start new measurements.

24.8.24.3 Pulse width measurement

If the channel (n) is configured to capture rising edges (channel (n) ELSB:ELSA = 0:1) and the channel (n+1) to capture falling edges (channel (n+1) ELSB:ELSA = 1:0), then the positive polarity pulse width is measured. If the channel (n) is configured to capture falling edges (channel (n) ELSB:ELSA = 1:0) and the channel (n+1) to capture rising edges (channel (n+1) ELSB:ELSA = 0:1), then the negative polarity pulse width is measured.

The pulse width measurement can be made in ”One-Shot Capture mode” or ”Continuous Capture mode”.

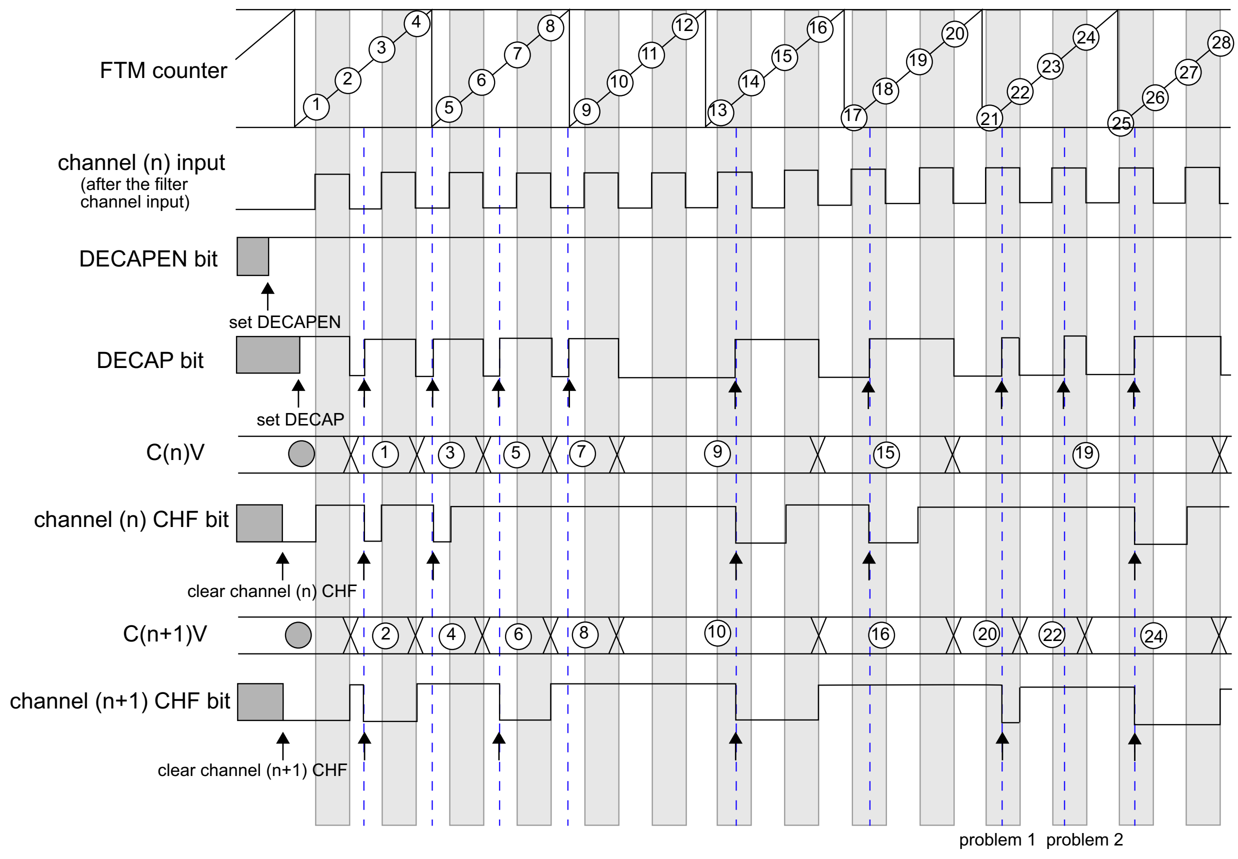
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The channel (n) CHF bit is set when the first edge of this pulse is detected, that is, the edge selected by channel (n) ELSB:ELSA bits. The channel (n+1) CHF bit is set and DECAP bit is cleared when the second edge of this pulse is detected, that is, the edge selected by channel (n+1) ELSB:ELSA bits. Both DECAP and channel (n+1) CHF bits indicate when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear channel (n) CHF, and clear channel (n+1) CHF are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear channel (n) CHF, and clear channel (n+1) CHF.
- Problem 2: channel (n) input = 1, set DECAP, not clear channel (n) CHF, and not clear channel (n+1) CHF.

Fig 127. Dual Edge Capture - One-Shot mode for positive polarity pulse width measurement

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The channel (n) CHF bit is set when the first edge of the positive polarity pulse is detected, that is, the edge selected by channel (n) ELSB:ELSA bits. The channel (n+1) CHF bit is set when the second edge of this pulse is detected, that is, the edge selected by channel (n+1) ELSB:ELSA bits. The channel (n+1) CHF bit indicates when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear channel (n) CHF, and clear channel (n+1) CHF are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear channel (n) CHF, and clear channel (n+1) CHF.
- Problem 2: channel (n) input = 1, set DECAP, not clear channel (n) CHF, and not clear channel (n+1) CHF.

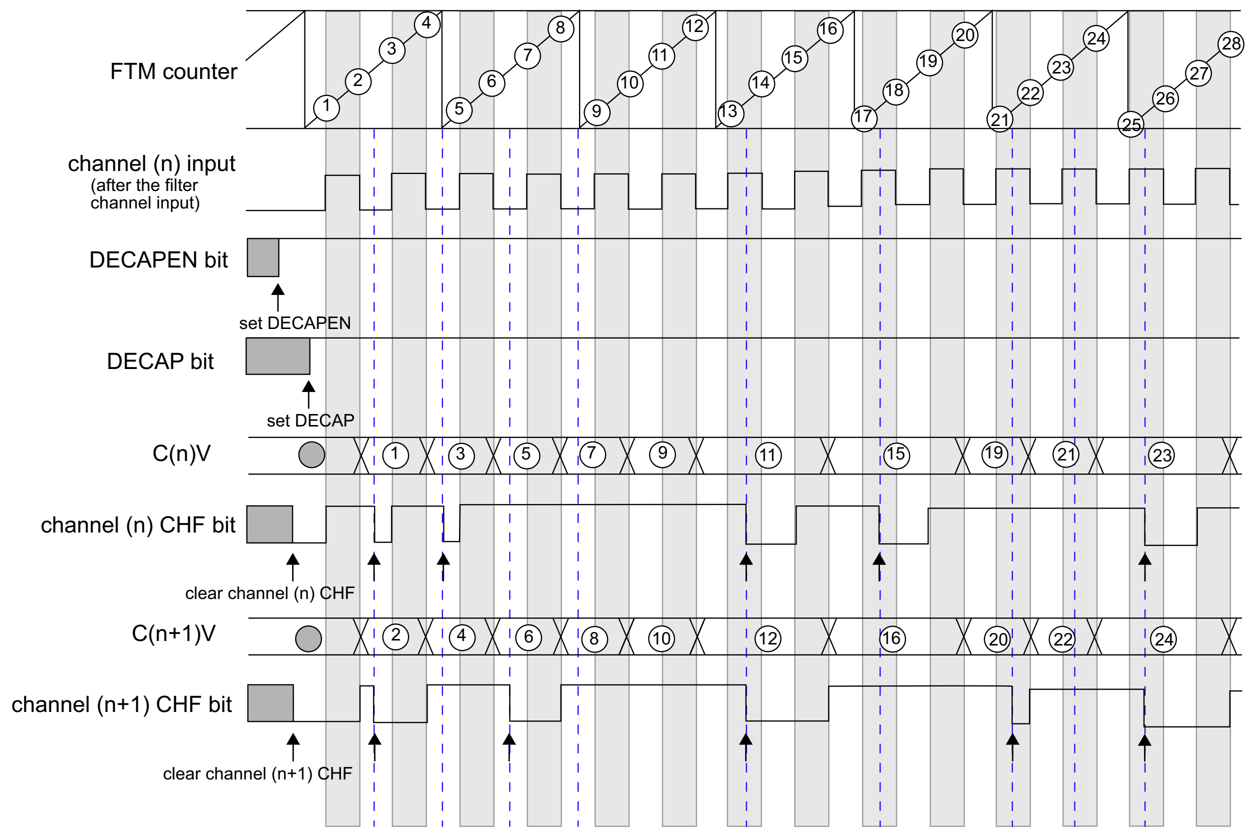
Fig 128. Dual Edge Capture - Continuous mode for positive polarity pulse width measurement

24.8.24.4 Period measurement

If the channels (n) and (n+1) are configured to capture consecutive edges of the same polarity, then the period of the channel (n) input signal is measured. If both channels (n) and (n+1) are configured to capture rising edges (channel (n) ELSB:ELSA = 0:1 and channel (n+1) ELSB:ELSA = 0:1), then the period between two consecutive rising edges is measured. If both channels (n) and (n+1) are configured to capture falling edges (channel (n) ELSB:ELSA = 1:0 and channel (n+1) ELSB:ELSA = 1:0), then the period between two consecutive falling edges is measured.

The period measurement can be made in ”One-Shot Capture mode” or ”Continuous Capture mode”.

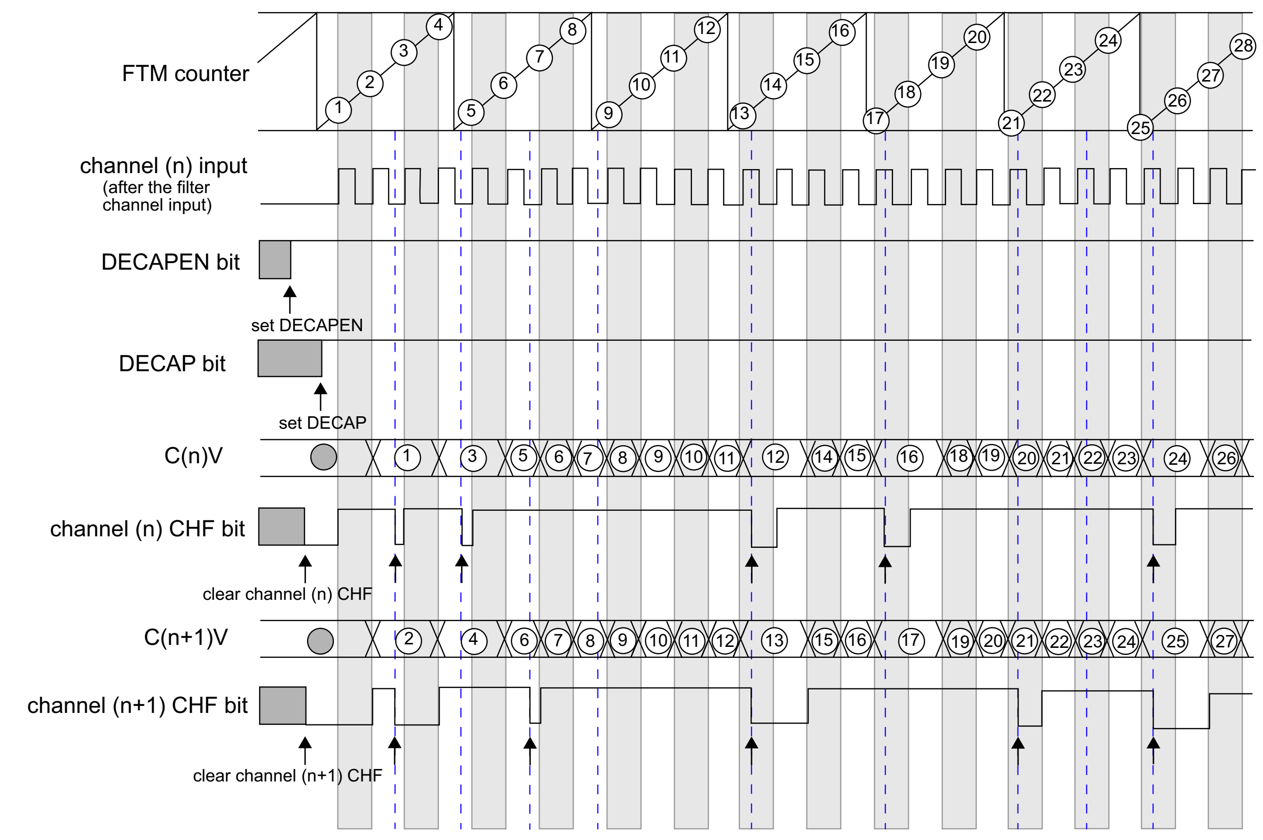
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next period. The channel (n) CHF bit is set when the first rising edge is detected, that is, the edge selected by channel (n) ELSB:ELSA bits. The channel (n+1) CHF bit is set and DECAP bit is cleared when the second rising edge is detected, that is, the edge selected by channel (n+1) ELSB:ELSA bits. Both DECAP and channel (n+1) CHF bits indicate when two selected edges were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear channel (n) CHF, and clear channel (n+1) CHF are made by the user.

Fig 129. Dual Edge Capture - One - Shot Mode to measure of the period between two consecutive rising edges

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The channel (n) CHF bit is set when the first rising edge is detected, that is, the edge selected by channel (n) ELSB:ELSA bits. The channel (n+1) CHF bit is set when the second rising edge is detected, that is, the edge selected by channel (n+1) ELSB:ELSA bits. The channel (n+1) CHF bit indicates when two edges of the period were captured and the C(n)V and C(n+1)V registers are ready for reading.



Note
- The commands set DECAPEN, set DECAP, clear channel (n) CHF, and clear channel (n+1) CHF are made by the user.

Fig 130. Dual Edge Capture - Continuous mode to measure of the period between two consecutive rising edges

24.8.24.5 Read coherency mechanism

The Dual Edge Capture mode implements a read coherency mechanism between the FTM counter value captured in C(n)V and C(n+1)V registers. The read coherency mechanism is illustrated in the following figure. In this example, the channels (n) and (n +1) are in Dual Edge Capture – Continuous mode for positive polarity pulse width measurement. Thus, the channel (n) is configured to capture the FTM counter value when there is a rising edge at channel (n) input signal, and channel (n+1) to capture the FTM counter value when there is a falling edge at channel (n) input signal.

When a rising edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n) capture buffer. The channel (n) capture buffer value is transferred to C(n)V register when a falling edge occurs in the channel (n) input signal. C(n)V register has the FTM counter value when the previous rising edge occurred, and the channel (n) capture buffer has the FTM counter value when the last rising edge occurred.

When a falling edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n+1) capture buffer. The channel (n+1) capture buffer value is transferred to C(n+1)V register when the C(n)V register is read.

In the following figure, the read of C(n)V returns the FTM counter value when the event 1 occurred and the read of C(n+1)V returns the FTM counter value when the event 2 occurred.

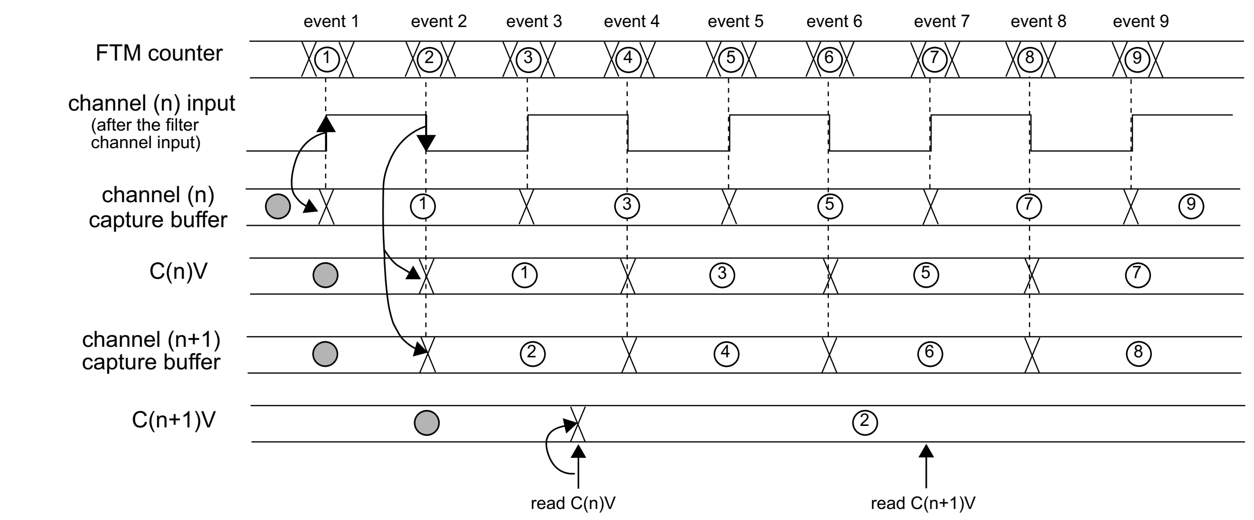


Fig 131. Dual edge Capture mode read coherency mechanism

C(n)V register must be read prior to C(n+1)V register in dual edge capture one-shot and continuous modes for the read coherency mechanism works properly.

24.8.25 Quadrature Decoder Mode

The quadrature decoder mode is enabled if QUADEN = 1. The quadrature decoder mode uses the input signals phase A and B to control the FTM counter increment and decrement.

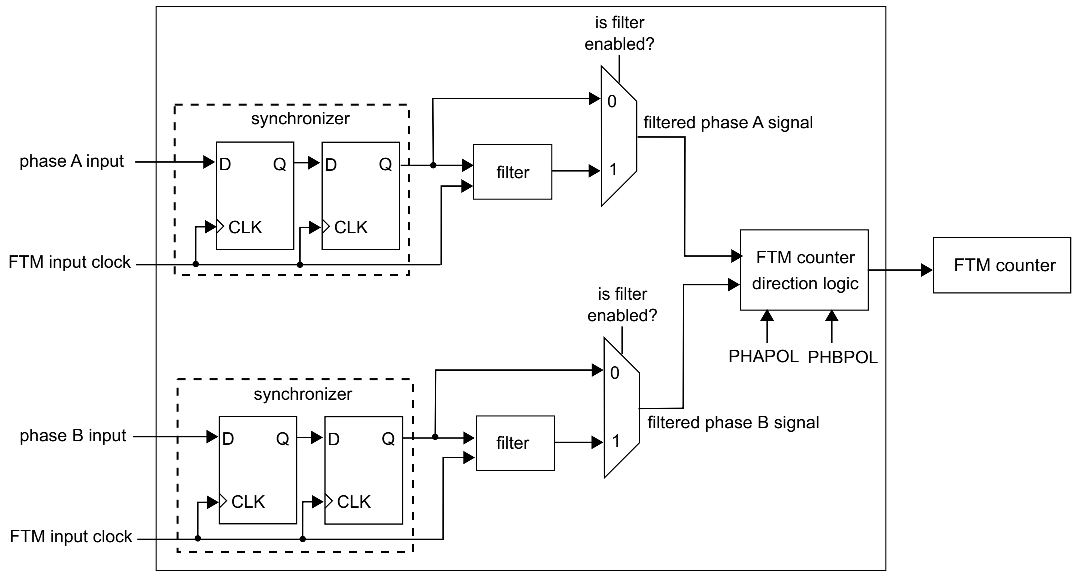


Fig 132. Diagram for Quadrature Decoder

Each one of input signals phase A and B has a filter that is equivalent to the channel input filter ("Filter for Input Capture Mode"). The phase A input filter is enabled by PHAFLTREN bit and its value is defined by CH0FVAL[3:0] bits. The phase B input filter is enabled by PHBFLTREN bit and its value is defined by CH1FVAL[3:0] bits.

Except for CH0FVAL[3:0] and CH1FVAL[3:0] bits, no channel logic is used in quadrature decoder mode.

Note: The FTM counter is clocked by the phase A and B input signals when quadrature decoder mode is enabled. Therefore it is expected that the quadrature decoder mode be used only with the FTM channels in input capture or output compare modes.

Note: An edge at phase A must not occur together an edge at phase B and vice-versa.

The PHAPOL bit selects the polarity of the phase A input, and the PHBPOL bit selects the polarity of the phase B input.

The QUADMODE selects the encoding mode used in the quadrature decoder mode. If QUADMODE = 1, then the count and direction encoding mode is enabled; see the following figure. In this mode, the phase B input value indicates the counting direction, and the phase A input defines the counting rate. The FTM counter is updated when there is a rising edge at phase A input signal.

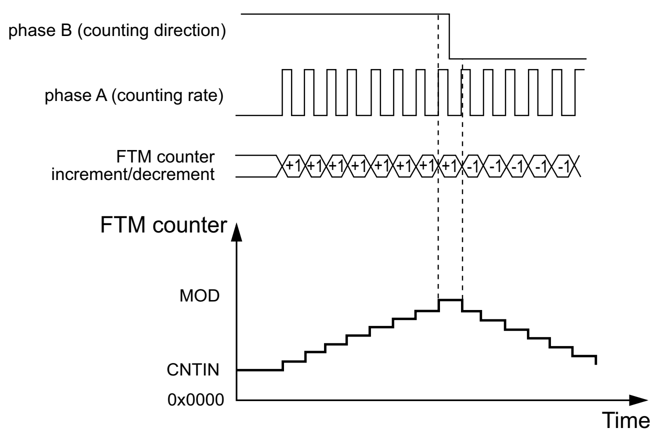


Fig 133. Quadrature Decoder - Count and Direction Encoding mode

If QUADMODE = 0, then the phase A and phase B Encoding mode is enabled; see the

following figure. In this mode, the relationship between phase A and B signals indicates

the counting direction, and phase A and B signals define the counting rate. The FTM counter is updated when there is an edge either at the phase A or phase B signals.

If PHAPOL = 0 and PHBPOL = 0, then the FTM counter increment happens when:

- there is a rising edge at phase A signal and phase B signal is at logic zero
- there is a rising edge at phase B signal and phase A signal is at logic one
- there is a falling edge at phase B signal and phase A signal is at logic zero
- there is a falling edge at phase A signal and phase B signal is at logic one

and the FTM counter decrement happens when:

- there is a falling edge at phase A signal and phase B signal is at logic zero
- there is a falling edge at phase B signal and phase A signal is at logic one
- there is a rising edge at phase B signal and phase A signal is at logic zero
- there is a rising edge at phase A signal and phase B signal is at logic one

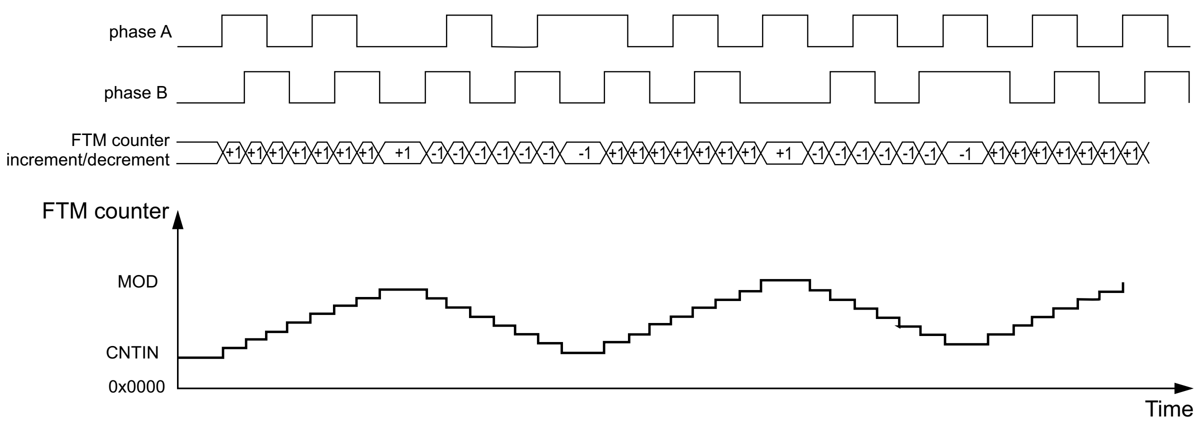


Fig 134. Quadrature Decoder Phase A and Phase B Encoding Mode

The following figure shows the FTM counter overflow in up counting. In this case, when the FTM counter changes from MOD to CNTIN, TOF and TOFDIR bits are set. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was up when the FTM counter overflow occurred.

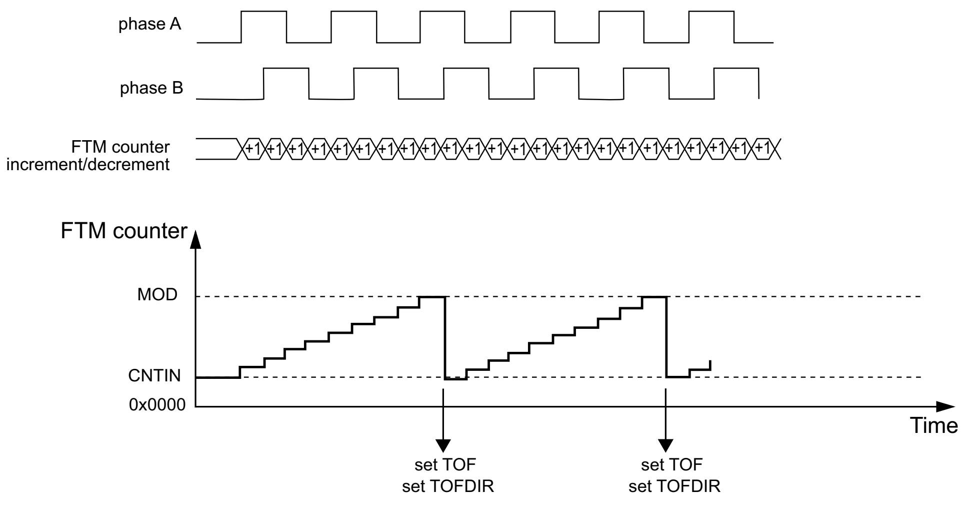


Fig 135. FTM Counter Overflow in Up Counting for Quadrature Decoder Mode

The following figure shows the FTM counter overflow in down counting. In this case, when the FTM counter changes from CNTIN to MOD, TOF bit is set and TOFDIR bit is cleared. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was down when the FTM counter overflow occurred.

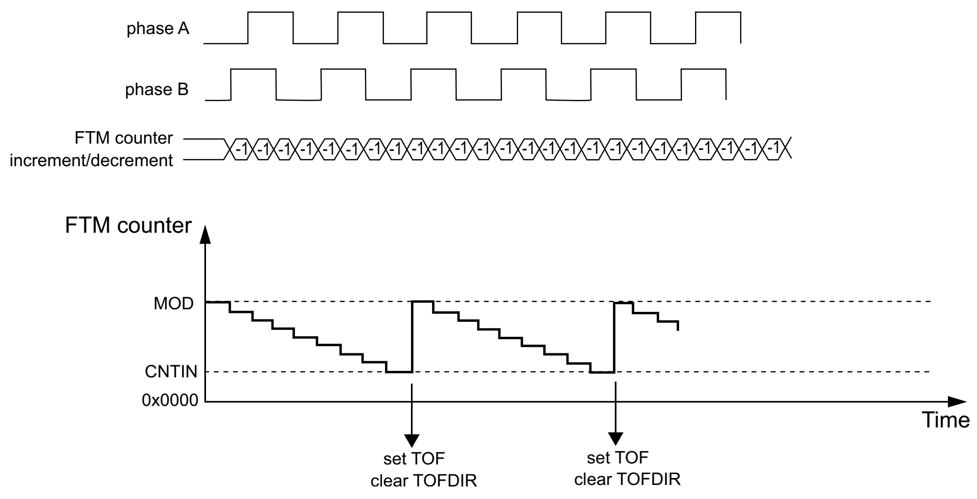


Fig 136. FTM Counter Overflow in Down Counting for Quadrature Decoder Mode

24.8.25.1 Quadrature Decoder boundary conditions

The following figures show the FTM counter responding to motor jittering typical in motor position control applications.

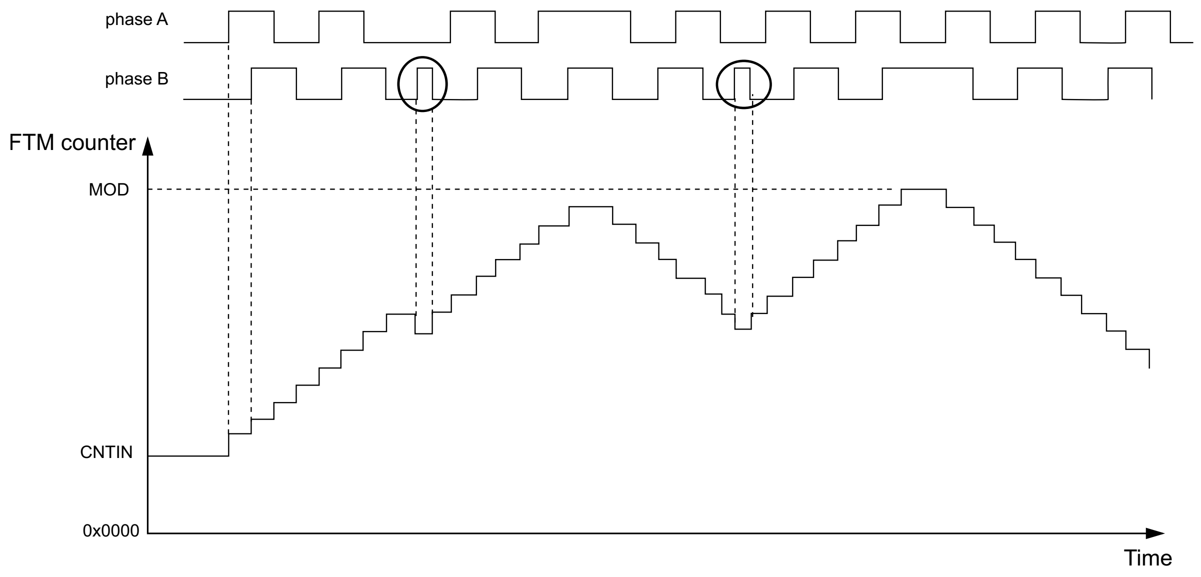


Fig 137. Motor position jittering in a mid count value

The following figure shows motor jittering produced by the phase B and A pulses respectively:

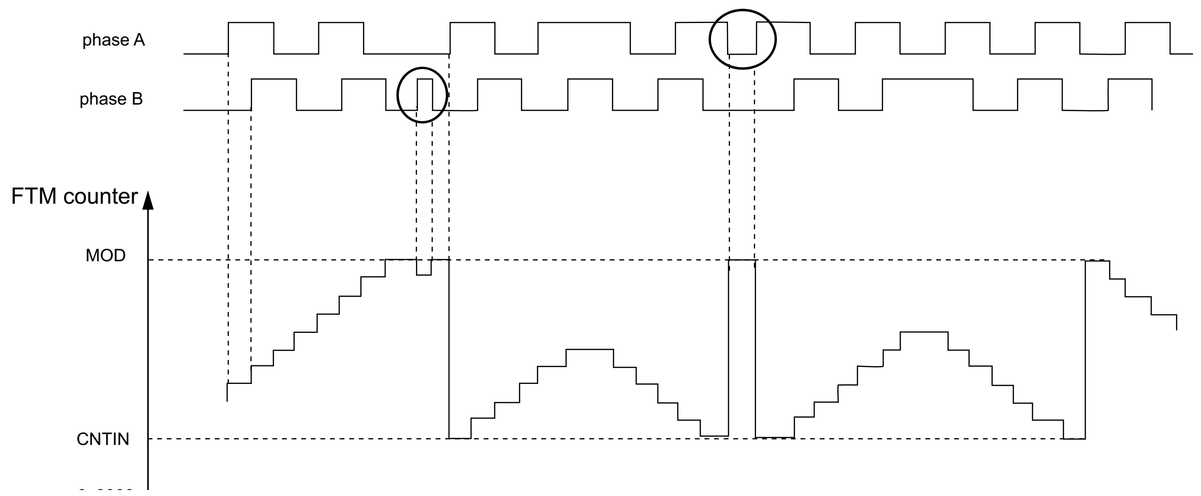


Fig 138. Motor position jittering near maximum and minimum count value

The first highlighted transition causes a jitter on the FTM counter value near the maximum count value (MOD). The second indicated transition occurs on phase A and causes the FTM counter transition between the maximum and minimum count values which are defined by MOD and CNTIN registers.

The appropriate settings of the phase A and phase B input filters are important to avoid glitches that may cause oscillation on the FTM counter value. The preceding figures show examples of oscillations that can be caused by poor input filter setup. Thus, it is important to guarantee a minimum pulse width to avoid these oscillations.

24.8.26 Debug mode

When the chip is in Debug mode, the BDMMODE[1:0] bits select the behavior of the FTM counter, the channel (n) CHF bit, the channels output, and the writes to the MOD, CNTIN, and C(n)V registers according to the following table.

Table 432. FTM behavior when the chip Is in Debug mode

BDMMODE	FTM Counter	channel (n) CHF bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
00	Stopped	can be set	Functional mode	Writes to these registers bypass the registers buffers
01	Stopped	is not set	The channels outputs are forced to their safe value according to POLn bit	Writes to these registers bypass the registers buffers
10	Stopped	is not set	The channels outputs are frozen when the chip enters in Debug mode	Writes to these registers bypass the registers buffers
11	Functional mode	can be set	Functional mode	Functional mode

Note that if BDMMODE[1:0] = 2’b00 then the channels outputs remain at the value when the chip enters in Debug mode, because the FTM counter is stopped. However, the following situations modify the channels outputs in this Debug mode.

- Write any value to CNT register; see ”Counter reset”. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for those channels set to Output Compare mode.
- FTM counter is reset by PWM Synchronization mode; see ”FTM counter synchronization”. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for channels in Output Compare mode.
- In the channels outputs initialization, the channel (n) output is forced to the CH(n)OI bit value when the value 1 is written to INIT bit. See ”Initialization”.

Note: The BDMMODE[1:0] = 2’b00 must not be used with the ”Fault Control”. Even if the fault control is enabled and a fault condition exists, the channels outputs are updated as above.

Note: If CLKS[1:0] = 2'b00 in Debug, a non-zero value is written to CLKS in Debug, and CnV = CNTIN when the Debug is disabled, then the CHF bit is set (since if the channel is a 0% EPWM signal) when the Debug is disabled.

24.8.27 Reload Points

This feature allows to update the registers CNTIN, HCR, MOD and C(n)V with the value of their write buffer at the selected reload point.

Note:

- This feature is independent of the PWM synchronization.
- At these reload points neither the channels outputs nor the FTM counter are changed. Software must select these reload points at the safe points in time.

24.8.27.1 Reload Opportunities

The reload opportunities are:

1. At the half cycle

This reload opportunity is enabled if (HCSEL = 1) and it happens at the half cycle (FTM counter = HCR register). The software should calculate the half cycle value according to the FTM counter configuration, then writes this value to the register HCR.

2. At the channel (n) match

This reload opportunity is enabled if (CH(n)SEL = 1) and it happens at the channel (n) match (FTM counter = C(n)V).

3. When the FTM counter is an up counterThis reload opportunity is when the FTM counter changes from (MOD) to (CNTIN -

   1) and it is always enabled.

   The following figure shows an example of the reload opportunities at the half cycle, at the channels match, and when the FTM counter is an up counter.

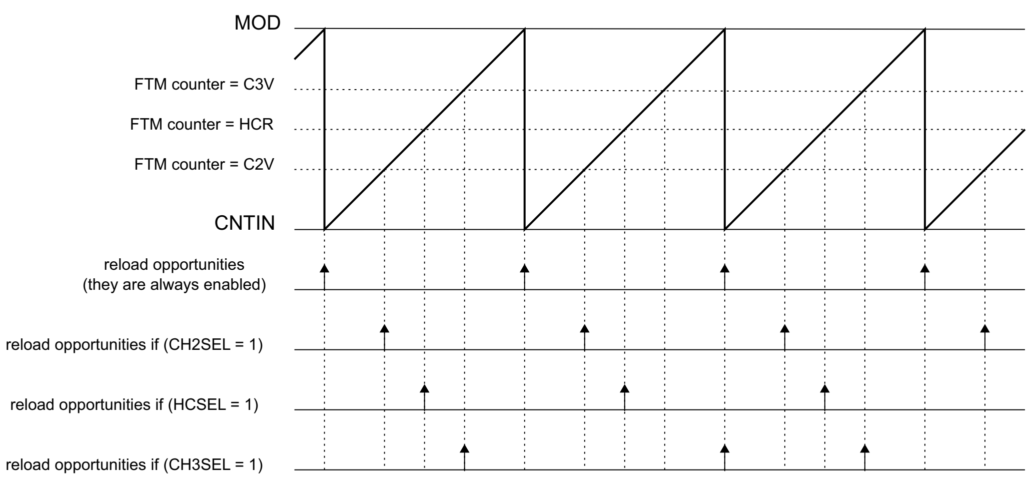

MOD
FTM counter = C3V
FTM counter = HCR
FTM counter = C2V
CNTIN
reload opportunities
(they are always enabled)
reload opportunities if (CH2SEL = 1)
reload opportunities if (HCSEL = 1)
reload opportunities if (CH3SEL = 1)

Fig 139. Reload opportunities when the FTM counter is an up counter

4. When the FTM counter is an up-down counter. In this case, the reload opportunities are enabled by the bits CNTMAX and CNTMIN

   according to Table 433 "Reload opportunities enabled by the bits CNTMAX and CNTMIN when the FTM counter is up-down counter".

Table 433. Reload opportunities enabled by the bits CNTMAX and CNTMIN when the FTM counter is up-down counter

CNTMAX	CNTMIN	Reload Opportunities
0	0	when the FTM counter changes from (MOD) to (MOD - 1)
0	1	when the FTM counter changes from (CNTMIN) to (CNTMIN + 1)
1	0	when the FTM counter changes from (MOD) to (MOD - 1)
1	1	when the FTM counter changes from (MOD) to (MOD - 1), and when the FTM counter changes from (CNTMIN) to (CNTMIN + 1)

The following figure shows an example of the reload opportunities at the half cycle,

at the channels match, and when the FTM counter is an up-down counter.

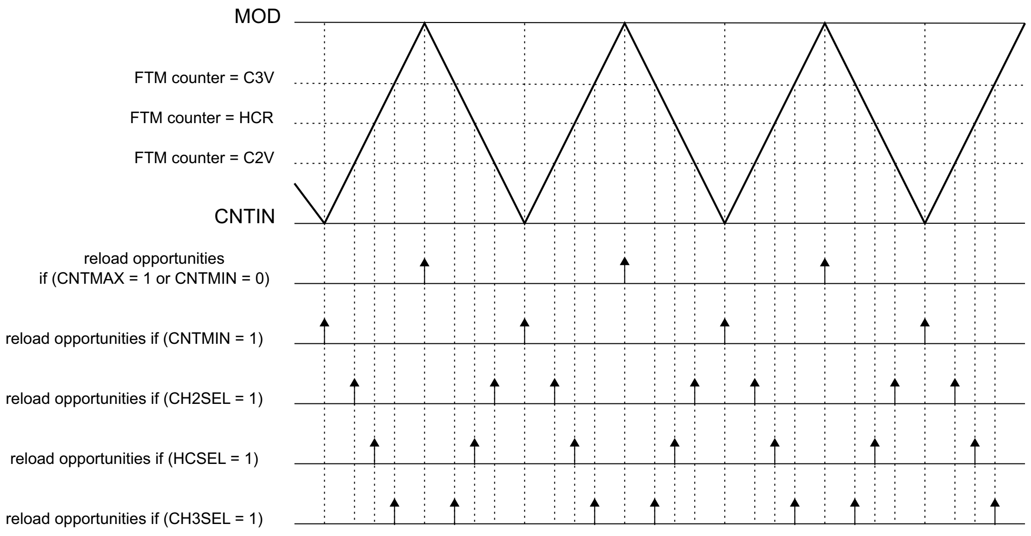

MOD
FTM counter = C3V
FTM counter = HCR
FTM counter = C2V
CNTIN
reload opportunities if (CNTMAX = 1 or CNTMIN = 0)
reload opportunities if (CNTMIN = 1)
reload opportunities if (CH2SEL = 1)
reload opportunities if (HCSEL = 1)
reload opportunities if (CH3SEL = 1)

Fig 140. Reload opportunities when the FTM counter is an up down counter

24.8.27.2 Frequency of Reload Opportunities

The LDFQ[4:0] bits define the number of enabled reload opportunities should happen until an enabled reload opportunity becomes a reload point. The following figure shows an example when the LDFQ[4:0] = 4.

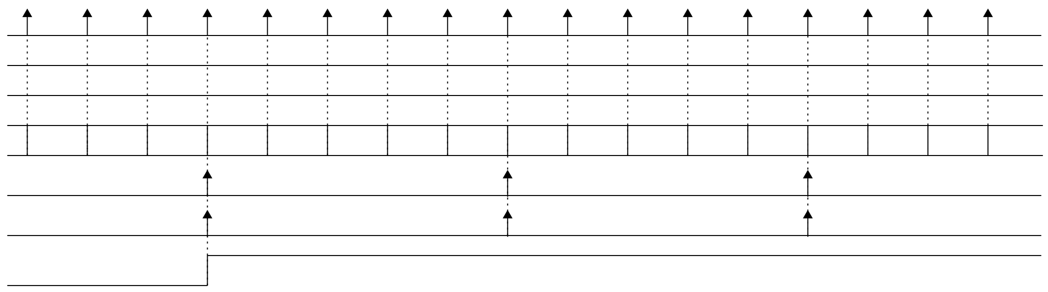


Fig 141. Frequency of Reload Opportunities with LDFQ[4:0] = 4

If LDFQ[4:0] = 0, then all reload opportunities are reload points.

The counter of the reload opportunities is reset when there is a write to the register CNT.

The RF bit is set at each reload point (see the figure above) independent of LDOK bit value. The reload point interrupt is generated when (RF = 1) and (RIE = 1).

24.8.27.3 Update of the Registers

After writing new value to the registers with write buffer, selecting which of them will be updated (according to Table 434 “Additional conditions to update the registers”), selecting the reload opportunities, selecting the frequency of the reload opportunities, thus the

LDOK bit should be set to enable the update of these registers at the next reload point.

Table 434. Additional conditions to update the registers

Register	Additional Condition
CNTIN	CNTINC = 1
HCR	-
MOD	-
C(n)V and C(n+1)V	SYNCENm = 1, where m is the pair of the channels (n) and (n+1)

24.8.28 Global Load

The global load mechanism allows several modules to have their double buffered registers synchronously reloaded after a synchronization event if a write to one operation is performed in the global load OK (GLDOK) bit in the PWMLOAD register. Global load may be enabled or disabled configuring the global load enable (GLEN) bit in the PWMLOAD register. Writing one in the GLDOK bit with GLEN enabled has the same effect of writing one in the LDOK bit. Refer to SoC specific information about global load connections.
Global load mechanism allows MOD, HCR, CNTIN, and C(n)V registers to be updated with the content of the register buffer at configurable reload point. The figure below shows an example of connection between FTM global load inputs and outputs considering that GLDOK bit is implemented outside from FTM module.

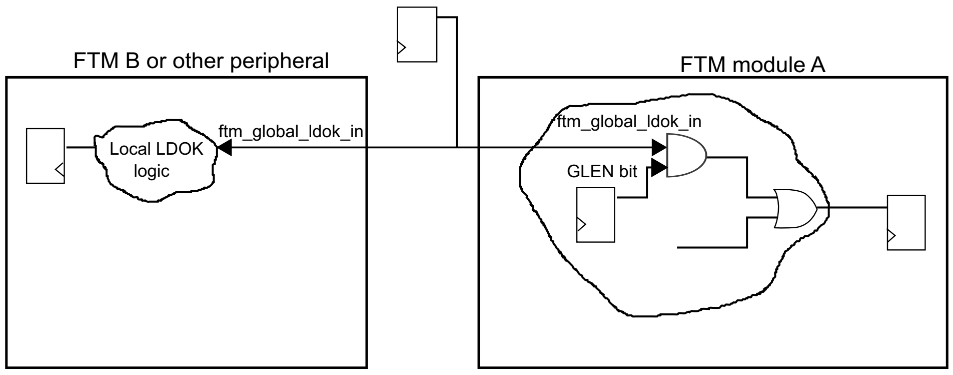


Fig 142. Global load logic

24.8.29 Global time base (GTB)

The global time base (GTB) is a FTM function that allows the synchronization of multiple FTM modules on a chip. The following figure shows an example of the GTB feature used to synchronize two FTM modules. In this case, the FTM A and B channels can behave as if just one FTM module was used, that is, a global time base.
The GTB functionality is implemented by the GTBEEN and GTBEOUT bits in the CONF register, the input signal *gtb_in*, and the output signal *gtb_out*. The GTBEEN bit enables gtb_in to control the FTM counter enable signal:

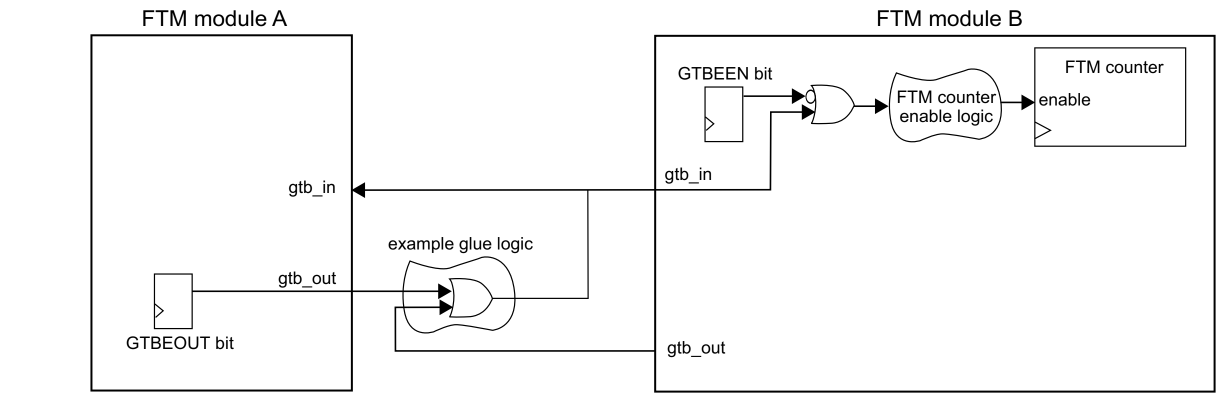


Fig 143. Global time base (GTB) block diagram

- If GTBEEN = 0, each one of FTM modules works independently according to their configured mode.
- If GTBEEN = 1, the FTM counter update is enabled only when gtb_in is 1.

In the configuration described in the preceding figure, FTM modules A and B have their FTM counters enabled if at least one of the gtb_out signals from one of the FTM modules is 1. There are several possible configurations for the interconnection of the gtb_in and gtb_out signals, represented by the example glue logic shown in the figure. Note that these configurations are chip-dependent and implemented outside of the FTM modules. See the chip-specific FTM information for the chip's specific implementation.

Note:

- In order to use the GTB signals to synchronize the FTM counter of different FTM modules, the configuration of each FTM module should guarantee that its FTM counter starts counting as soon as the gtb_in signal is 1.
- The GTB feature does not provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during FTM operation. The GTB feature only allows the FTM counters to *start* their operation synchronously.

24.8.29.1 Enabling the global time base (GTB)

1. Stop the FTM counter: Write 00b to SC[CLKS].
2. Program the FTM to the intended configuration.The FTM counter mode needs to be consistent across all participating modules.
3. Write 1 to CONF[GTBEEN] and write 0 to CONF[GTBEOUT] at the same time.
4. Select the intended FTM counter clock source in SC[CLKS]. The clock source needs to be consistent across all participating modules.
5. Reset the FTM counter: Write any value to the CNT register.

To initiate the GTB feature in the configuration described in the preceding figure, write 1 to CONF[GTBEOUT] in the FTM module used as the time base.

24.8.30 Channel trigger output

The channel trigger output provides a trigger signal which has one FTM input clock period width in the channel (n) output.

If the TRIGMODE bit of the CnSC register is set (TRIGMODE = 1), a trigger pulse with one FTM input clock width is generated in the channel (n) output when a match occurs. It is only allowed to use trigger mode when channel (n) is in EPWM or CPWM modes. The figures below show some cases of channel (n) trigger generation in the channel (n) output.

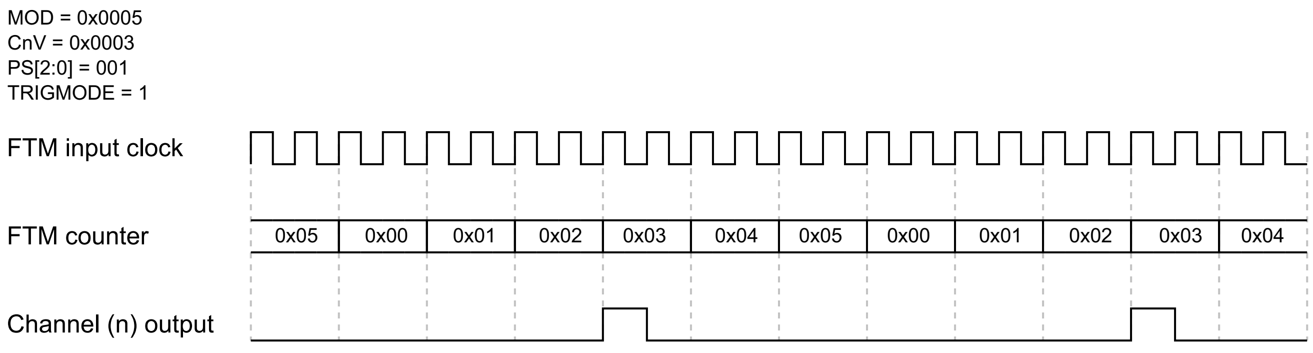


Fig 144. Example of channel (n) trigger at the channel (n) output in EPWM mode

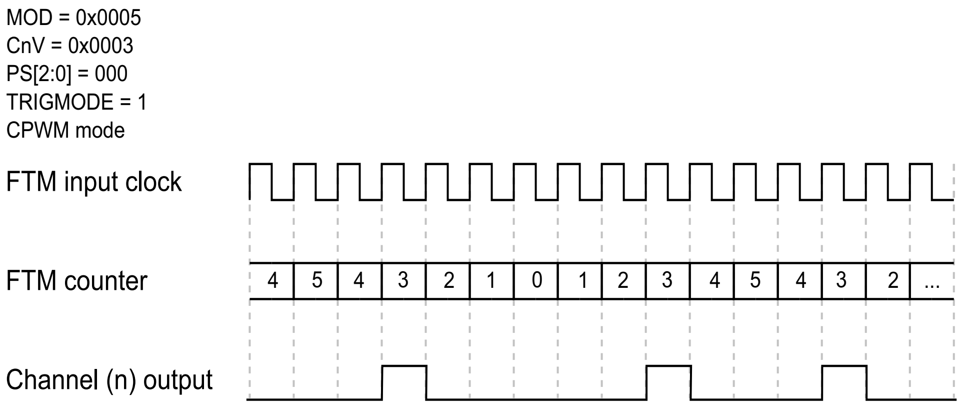


Fig 145. Example of channel (n) trigger at the channel (n) output in CPWM mode

24.8.31 External Control of Channels Output

The channel (n) PWMEN bit can be used in an FTM external logic to control the final value of the channel (n) output. This same logic can also control the channel (n) output when FSTATE = 1 and the channel (n) output is disabled by the Fault Control. The following figure shows an example of this external logic.

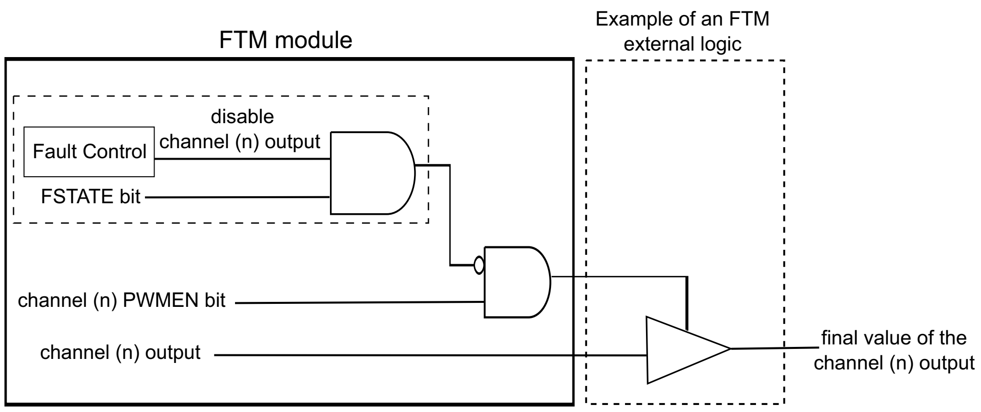

FTM module
Example of an FTM external logic
disable channel (n) output
Fault Control
FSTATE bit
channel (n) PWMEN bit
channel (n) output
final value of the channel (n) output

Fig 146. Example of the External Control of the Channel (n) Output

The term "channel (n) output" means the channel (n) output value after the Polarity Control. See ”Features Priority” and ”Polarity Control” for more details.

24.8.32 Dithering

FTM implements a fractional delay to achieve fine resolution on the generated PWM signals using dithering. The dithering can be used by applications where more resolution than one unit of the FTM counter is needed.

Two kinds of dithering are available: PWM period dithering and edge dithering.

24.8.32.1 PWM Period Dithering

The PWM period dithering is enabled when a non-zero value is written to FRACMOD.

The internal accumulator used in the PWM period dithering is reset when:

- the field MOD of the register MOD_MIRROR is updated with the value of its write buffer,
- the FRACMOD is updated with the value of its write buffer, or
- the FTM counter is stopped.

NOTE:
For the PWM period dithering, the register MOD_MIRROR should be used instead of the register MOD.

To avoid inconsistencies, the field FRACMOD is cleared when the field MOD of the register MOD is updated with the value of its write buffer.

The PWM period dithering is not available:

- when the FTM counter is a free running counter
- when the FTM is in quadrature decoder mode

24.8.32.1.1 Up Counting

When the FTM counter is an up counter and the PWM period dithering is enabled, at the end of each PWM period, the FRACMOD value is added to an internal 5-bit accumulator. When this accumulator overflows (that is, the result of the adding is greater or equal than 0x20), then one unit of FTM counter is added to the end of the current PWM period, and the accumulator remains with the rest of the subtraction: (the result of this adding - 0x20).

Due to one unit of FTM counter that can be added to the PWM period, the largest valid value for MOD is 0xFFFE for PWM period dithering with unsigned counting and 0x7FFE for PWM period dithering with signed counting.

The following figures show some examples of PWM period dithering when the FTM counter is an up counter.

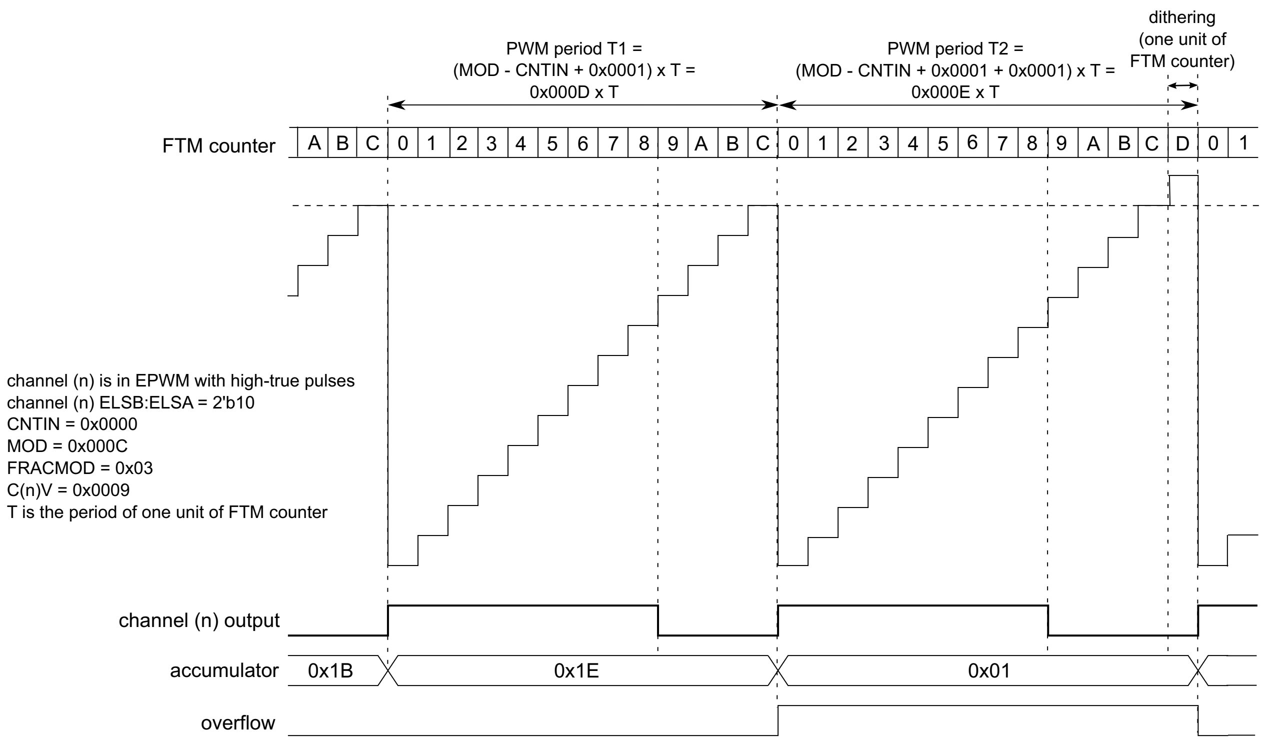

PWM period T1 =
(MOD - CNTIN + 0x0001) x T =
0x000D x T
PWM period T2 =
(MOD - CNTIN + 0x0001 + 0x0001) x T =
0x000E x T
dithering
(one unit of
FTM counter)
FTM counter
A B C 0 1 2 3 4 5 6 7 8 9 A B C 0 1 2 3 4 5 6 7 8 9 A B C D 0 1
channel (n) is in EPWM with high-true pulses
channel (n) ELSB:ELSA = 2'b10
CNTIN = 0x0000
MOD = 0x000C
FRACMOD = 0x03
C(n)V = 0x0009
T is the period of one unit of FTM counter
channel (n) output
accumulator
0x1B
0x1E
0x01
overflow

Fig 147. PWM Period Dithering with Up Counting

Assuming:

- the FTM counter is an up counter,
- T is one unit of FTM counter,
- the PWM period without period dithering is [(MOD - CNTIN + 1) x T],
- the number of PWM periods with period dithering is FRACMOD,
- the PWM period with period dithering is [(MOD - CNTIN + 1 + 1) x T],

thus, the average period (in decimal) is [(MOD - CNTIN + 1) + (FRACMOD/32)] x T, where the integer value is (MOD - CNTIN + 1) and the fractional value is (FRACMOD/32). See the example below.

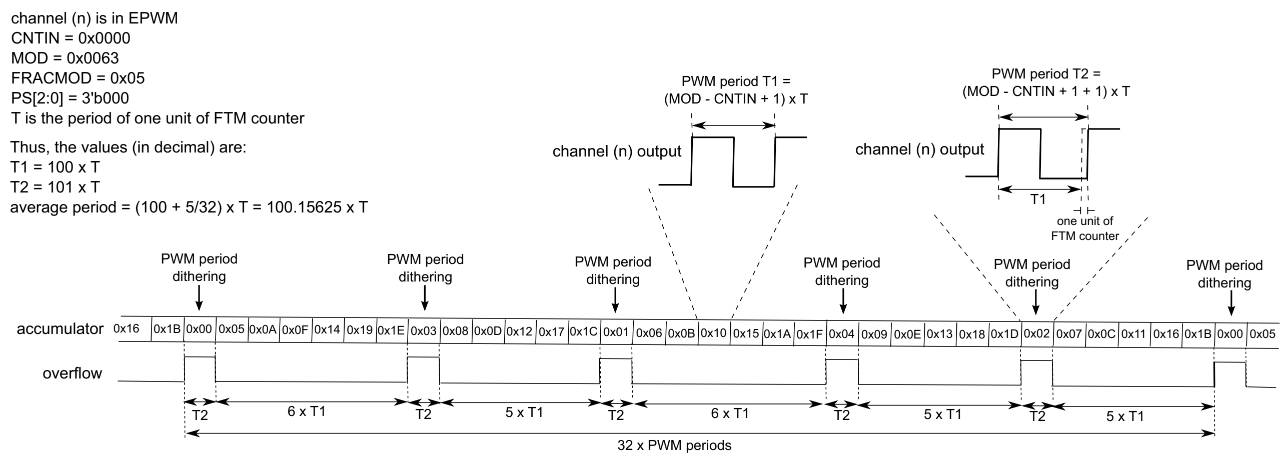

channel (n) is in EPWM
CNTIN = 0x0000
MOD = 0x0063
FRACMOD = 0x05
PS[2:0] = 3'b000
T is the period of one unit of FTM counter
Thus, the values (in decimal) are:
T1 = 100 x T
T2 = 101 x T
average period = (100 + 5/32) x T = 100.15625 x T
PWM period T1 = (MOD - CNTIN + 1) x T
channel (n) output
PWM period T2 = (MOD - CNTIN + 1 + 1) x T
channel (n) output
T1
one unit of FTM counter
PWM period dithering
accumulator
0x16 0x1B 0x00 0x05 0x0A 0x0F 0x14 0x19 0x1E 0x03 0x08 0x0D 0x12 0x17 0x1C 0x01 0x06 0x0B 0x10 0x15 0x1A 0x1F 0x04 0x09 0x0E 0x13 0x18 0x1D 0x02 0x07 0x0C 0x11 0x16 0x1B 0x00 0x05
overflow
T2
6 x T1
5 x T1
32 x PWM periods

Fig 148. Example of Average Period when the PWM Period Dithering is used with the Up Counting

Note:

For the generation of 100% PWM signal in the channel (n) (with channel (n) ELSB:ELSA = 2'b10) using EPWM mode and PWM Period Dithering, it is recommended to use (C(n) > MOD + 1).

For the generation of PWM signals in the channel (n) (with channel (n) ELSB:ELSA = 2'b10) using Combine mode and PWM Period Dithering, it is recommended to use:

- For 0% PWM signal: (C(n)V > MOD + 1) and (C(n+1)V >MOD + 1);
- For 100% PWM signal: (C(n)V = CNTIN) and (C(n+1)V >MOD + 1).

24.8.32.1.2 Up-Down Counting

When the FTM counter is an up-down counter and the PWM period dithering is enabled, at the end of each PWM period, the FRACMOD value is added to an internal 5-bit accumulator. When this accumulator overflows (that is, the result of the adding is greater or equal than 0x20), then one unit of FTM counter is added to the end of the current PWM period and other unit is added to the begin of the next PWM period (see the figure below). After the accumulator overflows, the accumulator remains with the rest of the subtraction: (the result of this adding - 0x20).

Due to one unit of FTM counter that can be added to the PWM period, the largest valid value for MOD is 0x7FFE for PWM period dithering in up-down counting (CPWM mode).

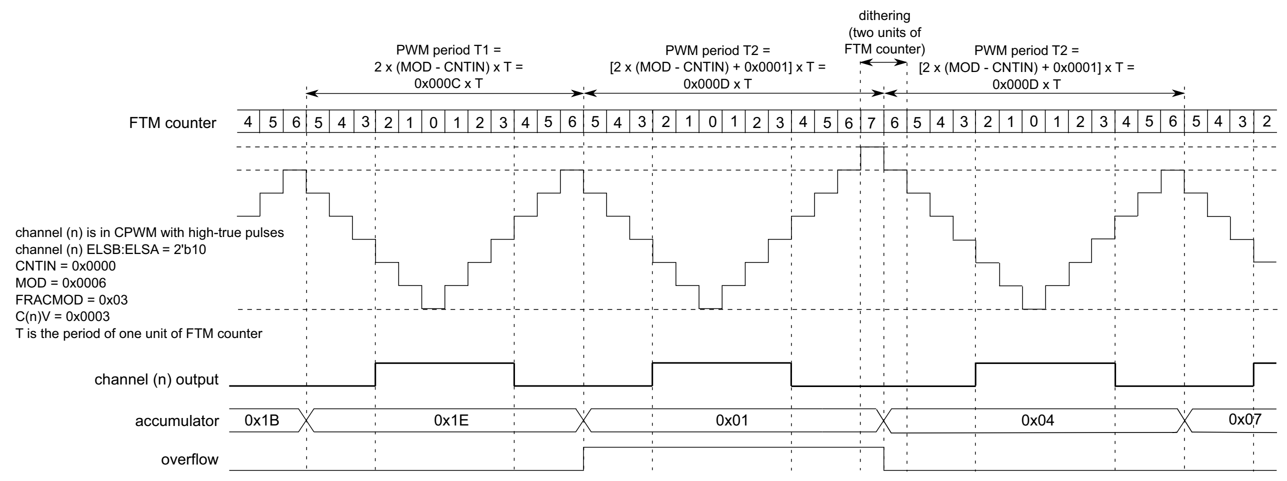

PWM period T1 = 2 x (MOD - CNTIN) x T = 0x000C x T
PWM period T2 = [2 x (MOD - CNTIN) + 0x0001] x T = 0x000D x T
dithering (two units of FTM counter)
PWM period T2 = [2 x (MOD - CNTIN) + 0x0001] x T = 0x000D x T
FTM counter
4 5 6 5 4 3 2 1 0 1 2 3 4 5 6 5 4 3 2 1 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 1 2 3 4 5 6 5 4 3 2
channel (n) is in CPWM with high-true pulses
channel (n) ELSB:ELSA = 2'b10
CNTIN = 0x0000
MOD = 0x0006
FRACMOD = 0x03
C(n)V = 0x0003
T is the period of one unit of FTM counter
channel (n) output
accumulator
0x1B
0x1E
0x01
0x04
0x07
overflow

Fig 149. PWM Period Dithering with Up-Down Counting

Note:

For the generation of 100% PWM signal in the channel (n) (with channel (n) ELSB:ELSA = 2'b10) using CPWM mode and PWM Period Dithering, it is recommended to use (C(n)V[15] = 0) and (C(n)V > MOD + 1) and (MOD ≠ 0x0000).

24.8.32.2 PWM Edge Dithering

The channel (n) internal accumulator used in the PWM edge dithering is reset when:

- the field VAL of the register C(n)V_MIRROR is updated with the value of its write buffer,
- the FRACVAL is updated with the value of its write buffer, or
- the FTM counter is stopped.

Note:

For the PWM edge dithering, the register C(n)V_MIRROR should be used instead of the register C(n)V.

To avoid inconsistencies, the field FRACVAL is cleared when the field VAL of the register C(n)V is updated with the value of its write buffer.

The PWM edge dithering is not available:

- to the channel in input modes, and
- to the channel in output compare mode.

24.8.32.2.1 EPWM Mode

The PWM edge dithering for channel (n) in EPWM mode is enabled when a non-zero value is written to the channel (n) FRACVAL.

If the channel (n) is in EPWM mode and the PWM edge dithering is enabled, at the end of each EPWM period, the channel (n) FRACVAL value is added to the channel (n) internal 5-bit accumulator. When this accumulator overflows (that is, the result of the adding is greater or equal than 0x20), the accumulator remains with the rest of the subtraction: (the result of this adding - 0x20).

In this configuration, the initial edge of EPWM duty cycle happens when (FTM counter = CNTIN), its position is not modified by the PWM edge dithering. If there was not the overflow of the channel (n) accumulator in the current EPWM period, then the final edge of EPWM duty cycle happens on the channel (n) match (FTM counter = C(n)V), that is, its position is not modified by the edge dithering. However, if there was the overflow of the channel (n) accumulator in the current EPWM period, then the final edge of EPWM duty cycle happens when (FTM counter = C(n)V + 0x0001).

The following figures show some examples of PWM edge dithering when the channel (n)is in EPWM mode.

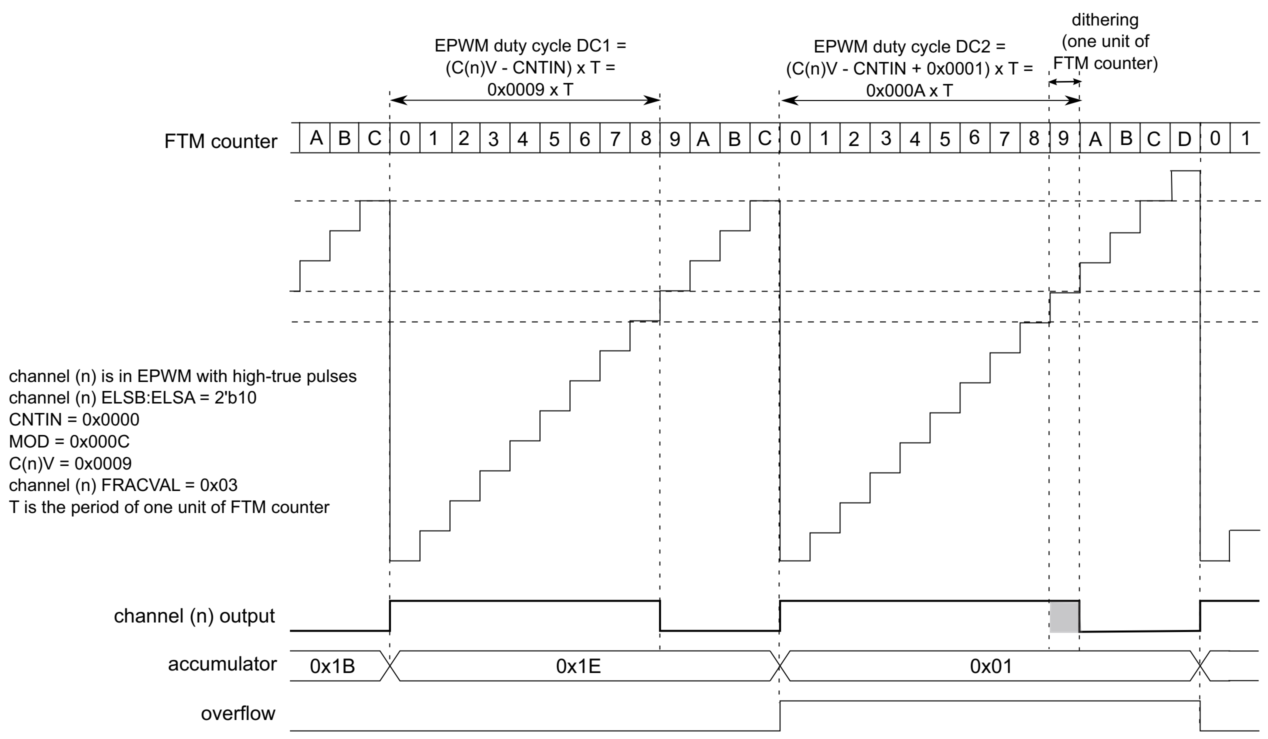

EPWM duty cycle DC1 = (C(n)V - CNTIN) x T = 0x0009 x T
EPWM duty cycle DC2 = (C(n)V - CNTIN + 0x0001) x T = 0x000A x T
dithering (one unit of FTM counter)
FTM counter
A B C 0 1 2 3 4 5 6 7 8 9 A B C 0 1 2 3 4 5 6 7 8 9 A B C D 0 1
channel (n) is in EPWM with high-true pulses
channel (n) ELSB:ELSA = 2'b10
CNTIN = 0x0000
MOD = 0x000C
C(n)V = 0x0009
channel (n) FRACVAL = 0x03
T is the period of one unit of FTM counter
channel (n) output
accumulator
0x1B
0x1E
0x01
overflow

Fig 150. Channel (n) is in EPWM Mode with PWM Edge Dithering

Assuming:

- the channel (n) is in EPWM mode,
- T is one unit of FTM counter,
- the EPWM duty cycle without edge dithering is [(C(n)V - CNTIN) x T],
- the number of PWM periods which duty cycle that has edge dithering is FRACVAL,
- the EWM duty cycle with edge dithering is [(C(n)V - CNTIN + 1) x T],

thus, the average duty cycle (in decimal) is [(C(n)V - CNTIN) + (FRACVAL/32)] x T,where the integer value is (C(n)V - CNTIN) and the fractional value is (FRACVAL/32). See the example below.

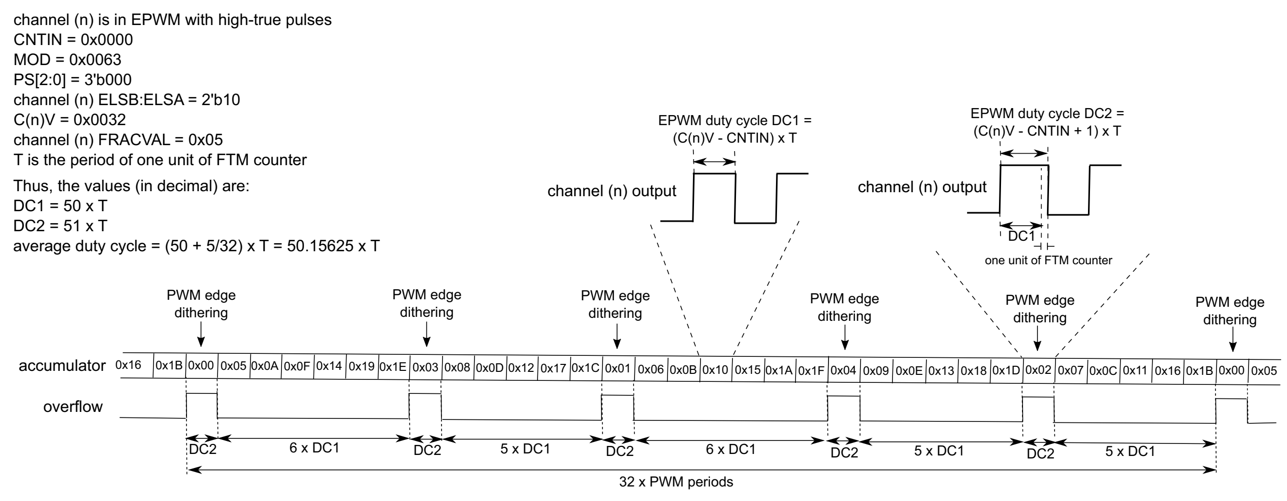

channel (n) is in EPWM with high-true pulses
CNTIN = 0x0000
MOD = 0x0063
PS[2:0] = 3'b000
channel (n) ELSB:ELSA = 2'b10
C(n)V = 0x0032
channel (n) FRACVAL = 0x05
T is the period of one unit of FTM counter
Thus, the values (in decimal) are:
DC1 = 50 x T
DC2 = 51 x T
average duty cycle = (50 + 5/32) x T = 50.15625 x T
EPWM duty cycle DC1 = (C(n)V - CNTIN) x T
channel (n) output
EPWM duty cycle DC2 = (C(n)V - CNTIN + 1) x T
channel (n) output
DC1
one unit of FTM counter
PWM edge dithering
accumulator
0x16 0x1B 0x00 0x05 0x0A 0x0F 0x14 0x19 0x1E 0x03 0x08 0x0D 0x12 0x17 0x1C 0x01 0x06 0x0B 0x10 0x15 0x1A 0x1F 0x04 0x09 0x0E 0x13 0x18 0x1D 0x02 0x07 0x0C 0x11 0x16 0x1B 0x00 0x05
overflow
DC2
6 x DC1
5 x DC1
32 x PWM periods

Fig 151. Example of Average Duty Cycle when the Channel (n) is in EPWM Mode with PWM Edge Dithering

24.8.32.2.2 CPWM Mode

The PWM edge dithering for channel (n) in CPWM mode is enabled when a non-zero value is written to the channel (n) FRACVAL.
If the channel (n) is in CPWM mode and the PWM edge dithering is enabled, at the end of each CPWM period, the channel (n) FRACVAL value is added to the channel (n) internal 5-bit accumulator. When this accumulator overflows (that is, the result of the adding is greater or equal than 0x20), the accumulator remains with the rest of the subtraction: (the result of this adding - 0x20).

In this configuration, if there was not the overflow of the channel (n) accumulator in the current CPWM period, then the duty cycle is not modified by the PWM edge dithering, that is, the initial edge of CPWM duty cycle happens on channel (n) match (FTM counter = C(n)V) when the FTM counter is decrementing, and the final edge of CPWM duty cycle on channel (n) match when the FTM counter is incrementing.

However, if there was the overflow of the channel (n) accumulator in the current CPWM period, then the initial edge of CPWM duty cycle happens when (FTM counter = C(n)V + 0x0001) and the FTM counter is decrementing, and the final edge of CPWM duty cycle when (FTM counter = C(n)V + 0x0001) and the FTM counter is incrementing.

The following figure shows an example of PWM edge dithering when the channel (n) is in CPWM mode.

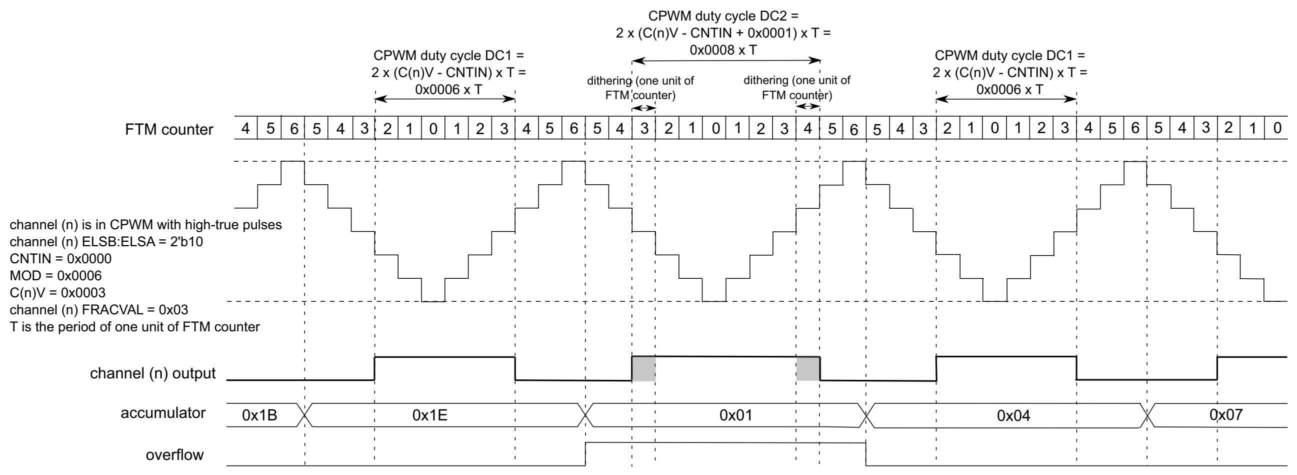

CPWM duty cycle DC2 =
2 x (C(n)V - CNTIN + 0x0001) x T =
0x0008 x T
CPWM duty cycle DC1 =
2 x (C(n)V - CNTIN) x T =
0x0006 x T
dithering (one unit of FTM counter)
dithering (one unit of FTM counter)
CPWM duty cycle DC1 =
2 x (C(n)V - CNTIN) x T =
0x0006 x T
FTM counter
4 5 6 5 4 3 2 1 0 1 2 3 4 5 6 5 4 3 2 1 0 1 2 3 4 5 6 5 4 3 2 1 0 1 2 3 4 5 6 5 4 3 2 1 0
channel (n) is in CPWM with high-true pulses
channel (n) ELSB:ELSA = 2'b10
CNTIN = 0x0000
MOD = 0x0006
C(n)V = 0x0003
channel (n) FRACVAL = 0x03
T is the period of one unit of FTM counter
channel (n) output
accumulator
0x1B
0x1E
0x01
0x04
0x07
overflow

Fig 152.

24.8.32.2.3 Combine Mode

In the Combine mode, the PWM edge dithering can be done:

- in the channel (n) match (FTM counter = C(n)V) edge or
- in the channel (n+1) match (FTM counter = C(n+1)V) edge.

The channel (n) match edge dithering is enabled when a non-zero value is written to the channel (n) FRACVAL.

For the channel (n) match edge dithering, the channel (n) has an internal 5-bit accumulator. At the end of each PWM period, the channel (n) FRACVAL value is added to the channel (n) accumulator. When this accumulator overflows (that is, the result of the adding is greater or equal than 0x20), the accumulator remains with the rest of the subtraction: (the result of this adding - 0x20).

If there was not the overflow of the channel (n) accumulator in the current PWM period, the channel (n) match edge is not modified, that is, it happens on channel (n) match. However, if there was the overflow of the channel (n) accumulator, the channel (n) match edge happens when (FTM counter = C(n)V + 0x0001).

The following figure shows an example of the channel (n) match edge dithering when the channels (n) and (n+1) are in Combine mode.

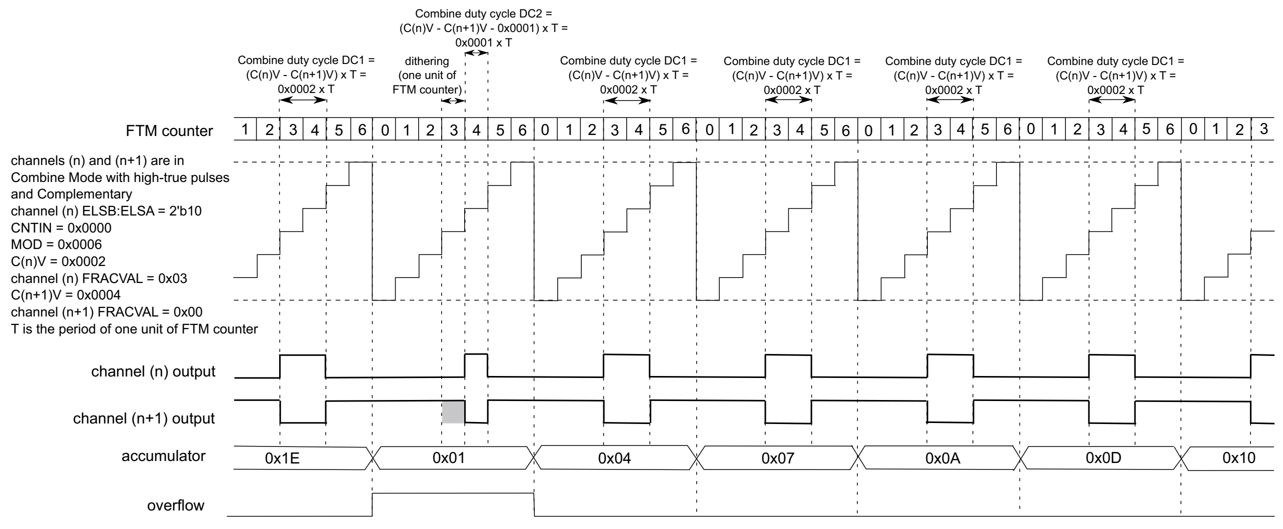

Combine duty cycle DC2 = (C(n)V - C(n+1)V - 0x0001) x T = 0x0001 x T
Combine duty cycle DC1 = (C(n)V - C(n+1)V) x T = 0x0002 x T
dithering (one unit of FTM counter)
FTM counter
channels (n) and (n+1) are in Combine Mode with high-true pulses and Complementary
channel (n) ELSB:ELSA = 2'b10
CNTIN = 0x0000
MOD = 0x0006
C(n)V = 0x0002
channel (n) FRACVAL = 0x03
C(n+1)V = 0x0004
channel (n+1) FRACVAL = 0x00
T is the period of one unit of FTM counter
channel (n) output
channel (n+1) output
accumulator
0x1E
0x01
0x04
0x07
0x0A
0x0D
0x10
overflow

Fig 153. Channel (n) Match Edge Dithering in Combine Mode

The channel (n+1) match edge dithering is enabled when a non-zero value is written to the channel (n+1) FRACVAL.

For the channel (n+1) match edge dithering, the channel (n+1) has an internal 5-bit accumulator. At the end of each PWM period, the channel (n+1) FRACVAL value is added to the channel (n+1) accumulator. When this accumulator overflows (that is, the result of the adding is greater or equal than 0x20), the accumulator remains with the rest of the subtraction: (the result of this adding - 0x20).

If there was not the overflow of the channel (n+1) accumulator in the current PWM period, the channel (n+1) match edge is not modified, that is, it happens on channel (n+1) match. However, if there was the overflow of the channel (n+1) accumulator, the channel (n+1) match edge happens when (FTM counter = C(n+1)V + 0x0001).

The following figure shows an example of the channel (n+1) match edge dithering when the channels (n) and (n+1) are in Combine mode.

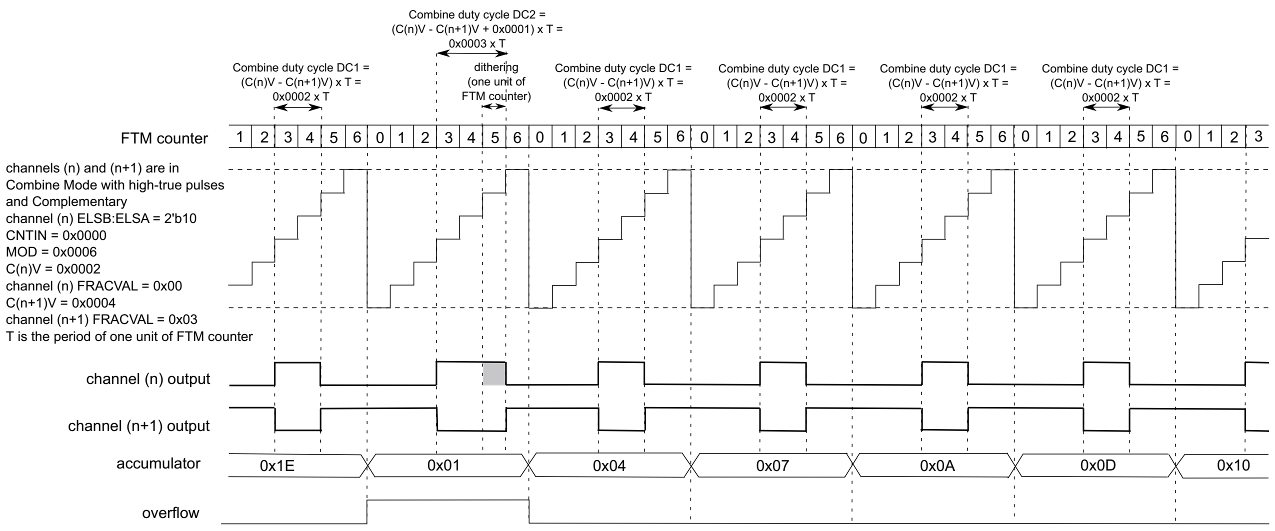

Combine duty cycle DC2 = (C(n)V - C(n+1)V + 0x0001) x T = 0x0003 x T
Combine duty cycle DC1 = (C(n)V - C(n+1)V) x T = 0x0002 x T
dithering (one unit of FTM counter)
FTM counter
channels (n) and (n+1) are in Combine Mode with high-true pulses and Complementary
channel (n) ELSB:ELSA = 2'b10
CNTIN = 0x0000
MOD = 0x0006
C(n)V = 0x0002
channel (n) FRACVAL = 0x00
C(n+1)V = 0x0004
channel (n+1) FRACVAL = 0x03
T is the period of one unit of FTM counter
channel (n) output
channel (n+1) output
accumulator
0x1E
0x01
0x04
0x07
0x0A
0x0D
0x10
overflow

Fig 154. Channel (n+1) Match Edge Dithering in Combine Mode

Note:
It is recommended to use only one PWM Edge Dithering (channel (n) PWM Edge Dithering or channel (n+1) PWM Edge Dithering) at a time.
For the generation of 0% PWM in the channel (n) (with channel (n) ELSB:ELSA = 2'b10) using Combine mode and PWM Edge Dithering, it is recommended to use:

- (C(n)V < CNTIN or C(n)V > MOD) and (channel (n) FRACVAL is zero) and
- (channel (n+1) FRACVAL is zero).

For the generation of 100% PWM in the channel (n) (with channel (n) ELSB:ELSA = 2'b10) using Combine mode and PWM Edge Dithering, it is recommended to use:

- (C(n)V = CNTIN) and (channel (n) FRACVAL is zero) and
- (C(n+1)V < CNTIN or C(n+1)V > MOD) and (channel (n+1) FRACVAL is zero).

24.9 Reset Overview

The FTM is reset whenever any chip reset occurs.
When the FTM exits from reset:

- the FTM counter and the prescaler counter are zero and are stopped (CLKS[1:0] = 2'b00);
- the timer overflow interrupt is zero ("Timer Overflow Interrupt");
- the channels interrupts are zero ("Channel (n) Interrupt");
- the fault interrupt is zero ("Fault Interrupt");
- the channels are in input capture mode ("Input Capture mode");

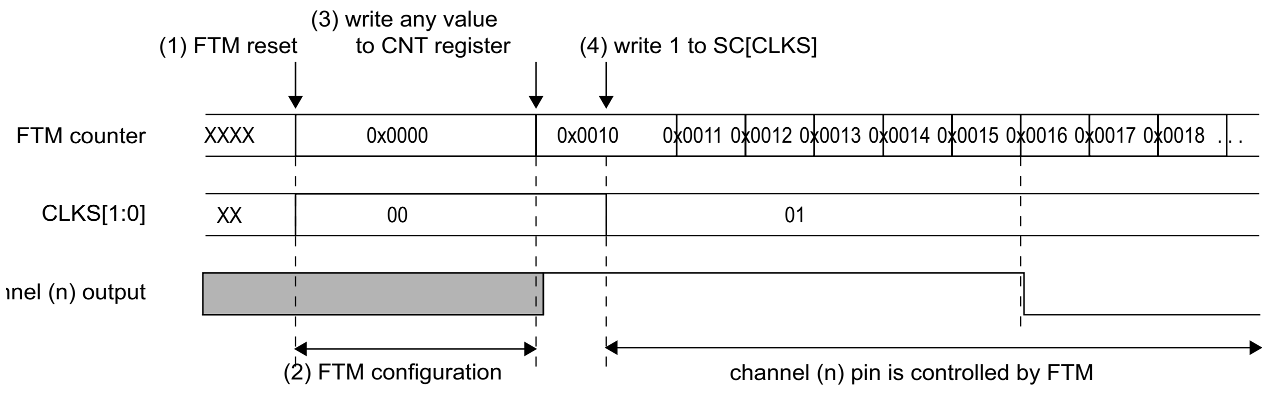
- the channels outputs are zero;
- the channels ELSB:ELSA = 0:0 ("Channel Modes") and PWMEN = 0 ("External Control of Channels Output").

The following figure shows the FTM behavior after the reset. At the reset (item 1), the FTM counter is disabled (CLKS[1:0] = 2'b00), its value is updated to zero and the pins are not controlled by FTM ("Channel Modes").

After the reset, the FTM should be configurated (item 2). It is necessary to define the FTM counter mode, the FTM counting limits (MOD and CNTIN registers value), the channels mode and CnV registers value according to the channels mode.

Thus, it is recommended to write any value to CNT register (item 3). This write updates the FTM counter with the CNTIN register value and the channels output with its initial value (except for channels in output compare mode) ("Counter reset").

The next step is to select the FTM counter clock by the CLKS[1:0] bits (item 4). It is important to highlight that the pins are only controlled by FTM when CLKS[1:0] bits are different from zero.


(1) FTM reset
(3) write any value to CNT register
(4) write 1 to SC[CLKS]
FTM counter
XXXX
0x0000
0x0010
0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017 0x0018 . . .
CLKS[1:0]
XX
00
01
nel (n) output
(2) FTM configuration
channel (n) pin is controlled by FTM

NOTES:
– CNTIN = 0x0010
– Channel (n) is in low-true combine mode with CNTIN < C(n)V < C(n+1)V < MOD
– C(n)V = 0x0015

Fig 155. FTM behavior after reset when the channel (n) is in Combine mode

The following figure shows an example when the channel (n) is in Output Compare mode and the channel (n) output is toggled when there is a match. In the Output Compare mode, the channel output is not updated to its initial value when there is a write to CNT register (item 3). In this case, use the software output control ("Software Output Control Mode") or the initialization ("Initialization") to update the channel output to the selected value (item 4).

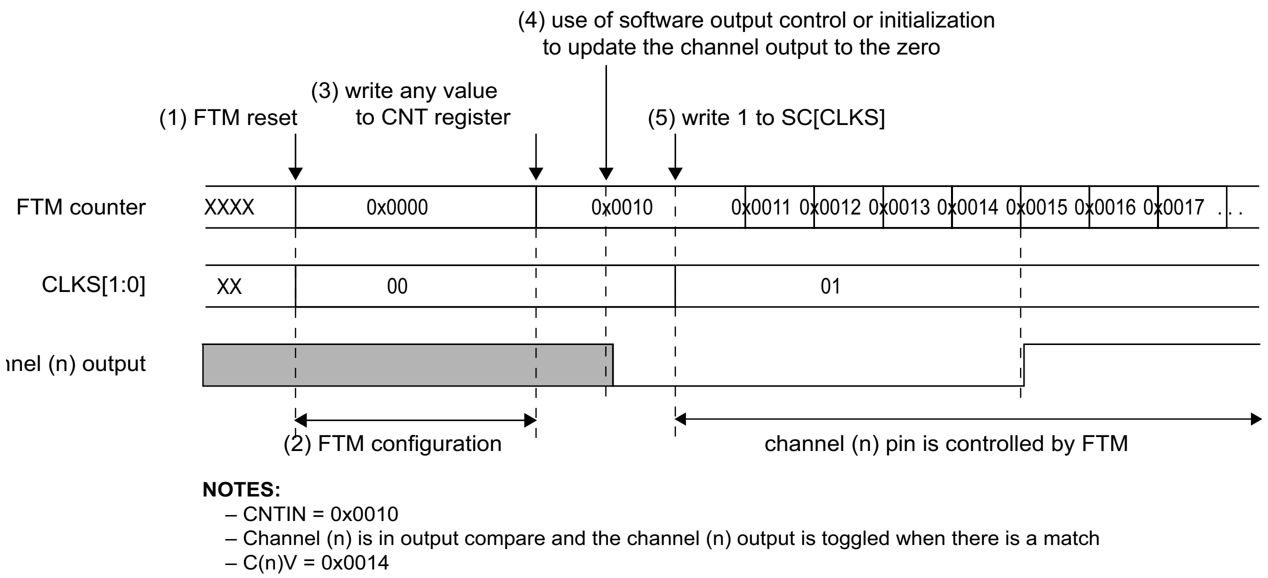

(4) use of software output control or initialization to update the channel output to the zero
(3) write any value to CNT register
(1) FTM reset
(5) write 1 to SC[CLKS]
FTM counter
XXXX
0x0000
0x0010
0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017 . . .
CLKS[1:0]
XX
00
01
ınel (n) output
(2) FTM configuration
channel (n) pin is controlled by FTM
NOTES:
– CNTIN = 0x0010
– Channel (n) is in output compare and the channel (n) output is toggled when there is a match
– C(n)V = 0x0014

Fig 156. FTM behavior after reset when the channel (n) is in Output Compare mode

24.10 FTM Interrupts

24.10.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

24.10.2 Reload Point Interrupt

The Reload Point interrupt is generated when (RIE = 1) and (RF = 1).

24.10.3 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHIE = 1) and (CHF = 1).

24.10.4 Fault Interrupt

The fault interrupt is generated when (FAULTIE = 1) and (FAULTF = 1).

24.11 Initialization Procedure

The following initialization procedure is recommended to configure the FlexTimer. This procedure can also be used to do a new configuration.

1. Define the POL bits.
2. Mask the channels outputs using SYNCHOM = 0. Two clocks after the write to OUTMASK, the channels outputs are in the safe value.
3. (Re)Configuration FTM counter and channels to generation of periodic signals:

- Disable the clock. Disable the Quadrature Decoder mode.
- Examples of (re)configuration:
  - Write to MOD
  - Write to CNTIN
  - Configure the channels that will be used
  - Write to CnV for the channels in output modes
  - (Re)Configure deadtime and fault control
  - Do not use the SWOC without SW synchronization (see item 6)
  - Do not use the Inverting without SW synchronization (see item 6)
  - Do not use the Initialization
  - Do not change the polarity control
  - Do not configure the HW synchronization
- Write any value to CNT. The FTM Counter is reset and the channels outputs are updated according to new configuration.
- Enable the clock. Write to CLKS[1:0] bits a value different from zero. Enable the Quadrature Decoder mode (if it is desired).
- Configure the SW synchronization for SWOC (if it is necessary), Inverting (if it is necessary) and Output Mask (always)

- Select synchronization for Output Mask
  - Write to SYNC (SWSYNC = 0, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)
- Write to SYNCONF
  - HW Synchronization can not be enabled (HWSOC = 0, HWINVC = 0, HWOM = 0, HWWRBUF = 0, HWRSTCNT = 0, HWTRIGMODE = 0)
  - SW Synchronization for SWOC (if it is necessary): SWSOC = [0/1] and SWOC = [0/1]
  - SW Synchronization for Inverting (if it is necessary): SWINVC = [0/1] and INVC = [0/1]
  - SW Synchronization for SWOM (always): SWOM = 1
  - No enable the SW Synchronization for write buffers (because the writes to registers with write buffer are done using CLKS[1:0] = 2'b00): SWWRBUF = 0 and CNTINC = 0
  - SW Synchronization for counter reset (always): SWRSTCNT = 1
  - Enhanced synchronization (always): SYNCMODE = 1
- If the SWOC is used (SWSOC = 1 and SWOC = 1), then write to SWOCTRL register.
- If the Inverting is used (SWINVC = 1 and INVC = 1), then write to INVCTRL register.
- Write to OUTMASK to enable the masked channels.
- Generate the Software Trigger
  - Write to SYNC (SWSYNC = 1, TRIG2 = 0, TRIG1 = 0, TRIG0 = 0, SYNCHOM = 1, REINIT = 0, CNTMAX = 0, CNTMIN = 0)
- Configure PWMEN bits ("External Control of Channels Output").

24.12 FTM usage guide

24.12.1 FTM Interrupts

The FlexTimer has multiple sources of interrupt. However, these sources are OR'd together to generate a single interrupt request to the interrupt controller. When an FTM interrupt occurs, read the FTM status registers (FMS, SC, and STATUS) to determine the exact interrupt source.

24.12.2 FTM Hall sensor support

For 3 phase motor control sensor-ed applications the use of Hall sensors, generally 3 sensors placed 120 degrees apart around the rotor, are deployed to detect position and speed. Each of the 3 sensors provides a pulse that applied to an input capture pin, can then be analyzed and both speed and position can be deduced. This device has two 2-channel FTMs. (FTM1 and FTM2) and thus provides 4 input capture pins. To simplify the calculations required by the CPU on each hall sensor's input, if all 3 inputs are "exclusively OR'd" into one timer channel and the free running counter is refreshed on every edge then this can simplify the speed calculation.
Via the SIM module and SIM_FTMOPT1 register the FTM2CH1SEL bit provides the choice of normal FTM2_CH1 input or the XOR of FTM2_CH0, FTM2_CH1 and FTM1_CH1 pins that will be applied to FTM2_CH1.

Note:
If the user utilizes FTM1_CH1 to be an input to FTM2_CH1, FTM1_CH0 can still be utilized for other functions.

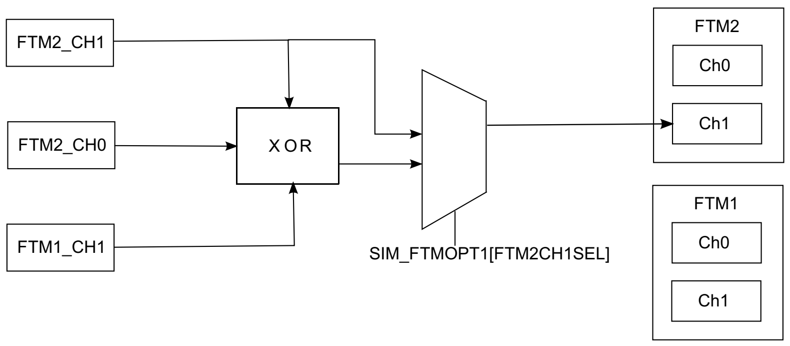


Fig 157. FTM Hall Sensor Configuration

24.12.3 FTM Modulation Implementation

FTM0 support a modulation function where the output channels when configured as PWM or Output Compare mode modulate another timer output when the channel signal is asserted. Any of the 8 channels of FTM0 can be configured to support this modulation function.

The SIM_FTMOPT1 register has control bits (FTMxCHySEL) that allow the user to select normal PWM/Output Compare mode on the corresponding FTM timer channel or modulate with FTM1_CH1. The diagram below shows the implementation for FTM0. See

SIM Block Guide for further information.
When FTM1_CH1 is used to modulate an FTM0 channel, then the user must configure FTM1_CH1 to provide a signal that has a higher frequency than the modulated FTM0 channel output. Also it limits the use of the FTM1_CH0 function, as the FTM1_CH1 will be programmed to provide a 50% duty PWM signal and limit the start and modulus values for the free running counter.

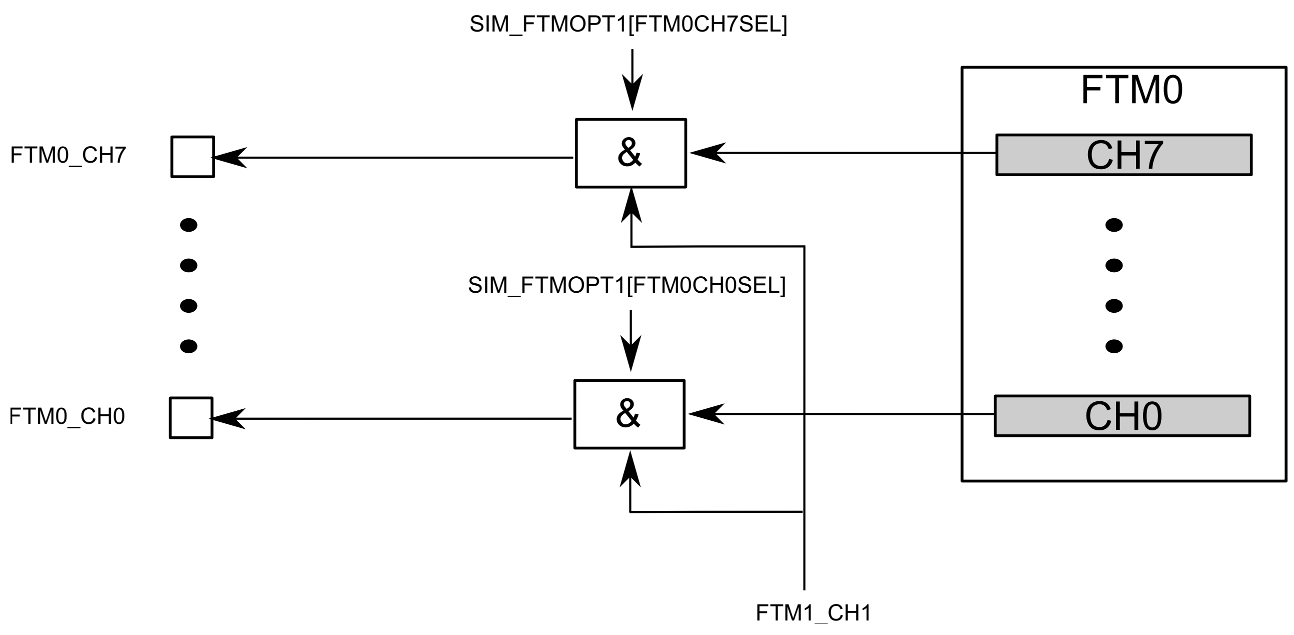


Fig 158. FTM Output Modulation

24.12.4 Global time base (GTB)

This chip provides the optional FTM global time base feature, see "Global time base (GTB)".
FTM supports global timer base through the GTB feature. Any of the FTM module could be used as the GTB_EN source. The global timer base only allows the FTM counters to start their operation synchronously, it does not automatically provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during misc FTM operation.

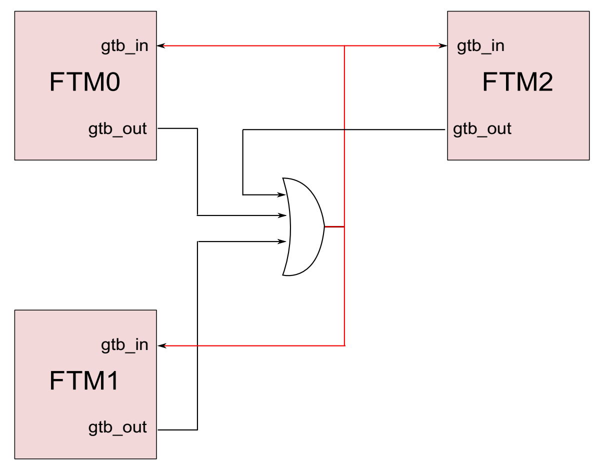


Fig 159. Global time base (GTB)

24.12.5 FTM BDM and debug halt mode

In the FTM chapter, references to the chip being in "BDM" are the same as the chip being in “debug halt mode", connected to DEBUG_HALTED signal.

UM11607

Chapter 25: Multi-Rate Timer (MRT)

Rev. 3 — April 2023 User manual

25.1 How to read this chapter

The MRT is available on all LPC86x parts.

25.2 Features

- 31-bit interrupt timer
- Four channels independently counting down from individually set values
- Repeat, bus-stall, and one-shot interrupt modes

25.3 Basic configuration

Configure the MRT using the following registers:

- In the SYSAHBCLKCTRL register, set bit 10 (Table 94) to enable the clock to the register interface.
- Clear the MRT reset using the PRESETCTRL register (Table 95).
- The global MRT interrupt is connected to interrupt #10 in the NVIC.

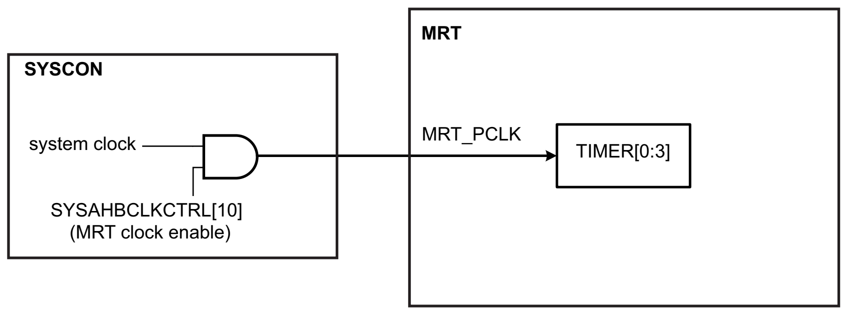


Fig 160. MRT clocking

25.4 Pin description

The MRT has no configurable pins.

25.5 General description

The Multi-Rate Timer (MRT) provides a repetitive interrupt timer with four channels. Each channel can be programmed with an independent time interval.

Each channel operates independently from the other channels in one of the following modes:

- Repeat interrupt mode. See Section 25.5.1.

- One-shot interrupt mode. See Section 25.5.2.
- Bus-stall mode.

The modes for each timer are set in the timer's control register. See Table 438.

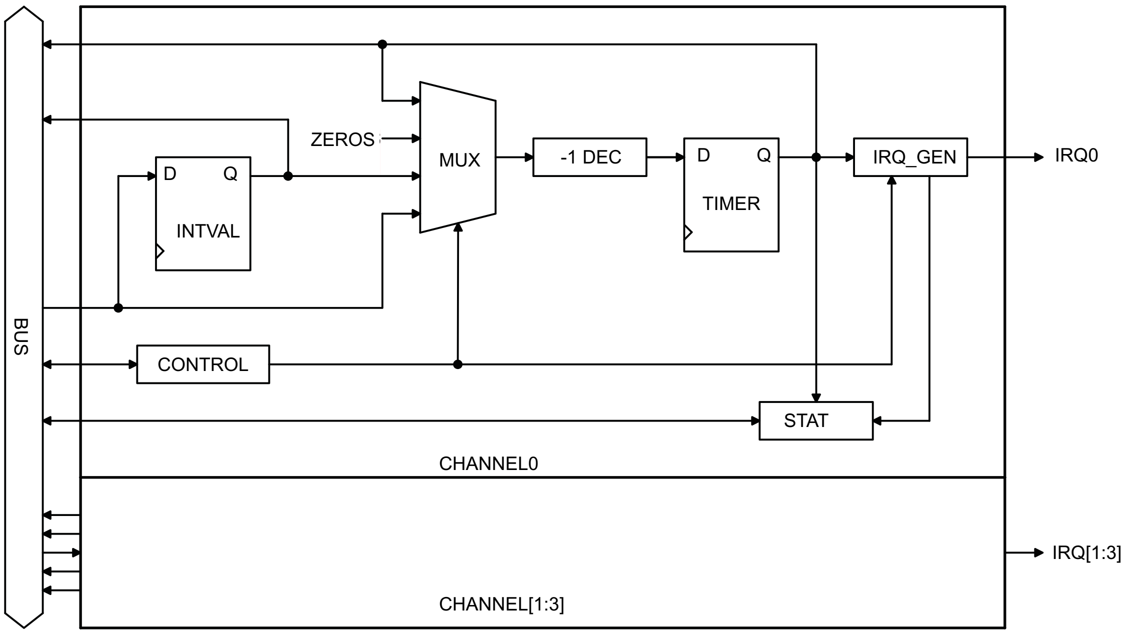


Fig 161. MRT block diagram

25.5.1 Repeat interrupt mode

The repeat interrupt mode generates repeated interrupts after a selected time interval. This mode can be used for software-based PWM or PPM applications.

When the timer n is in idle state, writing a non-zero value IVALUE to the INTVALn register immediately loads the time interval value IVALUE - 1, and the timer begins to count down from this value. When the timer reaches zero, an interrupt is generated, the value in the INTVALn register IVALUE - 1 is reloaded automatically, and the timer starts to count down again.

While the timer is running in repeat interrupt mode, you can perform the following actions:

- Change the interval value on the next timer cycle by writing a new value (>0) to the INTVALn register and setting the LOAD bit to 0. An interrupt is generated when the timer reaches zero. On the next cycle, the timer counts down from the new value.
- Change the interval value on-the-fly immediately by writing a new value (>0) to the INTVALn register and setting the LOAD bit to 1. The timer immediately starts to count down from the new timer interval value. An interrupt is generated when the timer reaches 0.
- Stop the timer at the end of time interval by writing a 0 to the INTVALn register and setting the LOAD bit to 0. An interrupt is generated when the timer reaches zero.
- Stop the timer immediately by writing a 0 to the INTVALn register and setting the LOAD bit to 1. No interrupt is generated when the INTVALn register is written.

25.5.2 One-shot interrupt mode

The one-shot interrupt generates one interrupt after a one-time count. With this mode, you can generate a single interrupt at any point. This mode can be used to introduce a specific delay in a software task.

When the timer is in the idle state, writing a non-zero value IVALUE to the INTVALn register immediately loads the time interval value IVALUE - 1, and the timer starts to count down. When the timer reaches 0, an interrupt is generated and the timer stops and enters the idle state.

While the timer is running in the one-shot interrupt mode, you can perform the following actions:

- Update the INTVALn register with a new time interval value (>0) and set the LOAD bit to 1. The timer immediately reloads the new time interval, and starts counting down from the new value. No interrupt is generated when the TIME_INTVALn register is updated.
- Write a 0 to the INTVALn register and set the LOAD bit to 1. The timer immediately stops counting and moves to the idle state. No interrupt is generated when the INTVALn register is updated.

25.5.3 One-shot bus stall mode

The one-shot bus stall mode stalls the bus interface for IVALUE +3 cycles of the system clock. For the Cortex-M0+, this mode effectively stops all CPU activity until the MRT has finished counting down to zero. At the end of the count-down, no interrupt is generated, instead the bus resumes its transactions. The bus stall mode allows to halt an application for a predefined amount of time and then resume, as opposed to creating a software loop or polling a timer. Since in bus-stall mode, there are no bus transactions while the MRT is counting down, the CPU consumes a minimum amount of power during that time. Typically, this mode can be used when an application must be idle for a short time (in the order of μs or 10 to 50 clock cycles) - for example when compensating for a settling time and thus no CPU activity is required.

For longer wait times, use the one-shot interrupt mode, which allows other enabled interrupts to be serviced.

Remark: Because the MRT resides on the APB, the total amount of wait cycles inserted in bus stall mode, 3 cycles have to be added to IVALUE to account for the AHB-to-APB bridge.

25.6 Register description

The reset values shown in Table 435 are POR reset values.

Table 435. Register overview: MRT (base address 0x4000 4000)

Name	Access	Address offset	Description	Reset value	Reference
INTVAL0	R/W	0x0	MRT0 Time interval value register. This value is loaded into the TIMER0 register.	0	Table 436
TIMER0	R	0x4	MRT0 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 437
CTRL0	R/W	0x8	MRT0 Control register. This register controls the MRT0 modes.	0	Table 438
STAT0	R/W	0xC	MRT0 Status register.	0	Table 439
INTVAL1	R/W	0x10	MRT1 Time interval value register. This value is loaded into the TIMER1 register.	0	Table 436
TIMER1	R/W	0x14	MRT1 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 437
CTRL1	R/W	0x18	MRT1 Control register. This register controls the MRT1 modes.	0	Table 438
STAT1	R/W	0x1C	MRT1 Status register.	0	Table 439
INTVAL2	R/W	0x20	MRT2 Time interval value register. This value is loaded into the TIMER2 register.	0	Table 436
TIMER2	R/W	0x24	MRT2 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 437
CTRL2	R/W	0x28	MRT2 Control register. This register controls the MRT2 modes.	0	Table 438
STAT2	R/W	0x2C	MRT2 Status register.	0	Table 439
INTVAL3	R/W	0x30	MRT3 Time interval value register. This value is loaded into the TIMER3 register.	0	Table 436
TIMER3	R/W	0x34	MRT3 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 437
CTRL3	R/W	0x38	MRT3 Control register. This register controls the MRT modes.	0	Table 438
STAT3	R/W	0x3C	MRT3 Status register.	0	Table 439
IDLE_CH	R	0xF4	Idle channel register. This register returns the number of the first idle channel.	0	Table 440
IRQ_FLAG	R/W	0xF8	Global interrupt flag register	0	Table 441

25.6.1 Time interval register

This register contains the MRT load value and controls how the timer is reloaded. The load value is IVALUE -1.

Table 436. Time interval register (INTVAL[0:3], address 0x4000 4000 (INTVAL0) to 0x4000 4030 (INTVAL3)) bit description

Bit	Symbol	Value	Description	Reset value
30:0	IVALUE		Time interval load value. This value is loaded into the TIMERn register and the MRTn starts counting down from IVALUE -1. If the timer is idle, writing a non-zero value to this bit field starts the timer immediately. If the timer is running, writing a zero to this bit field does the following: • If LOAD = 1, the timer stops immediately. • If LOAD = 0, the timer stops at the end of the time interval.	0
31	LOAD		Determines how the timer interval value IVALUE -1 is loaded into the TIMERn register. This bit is write-only. Reading this bit always returns 0.	0
		0	No force load. The load from the INTVALn register to the TIMERn register is processed at the end of the time interval if the repeat mode is selected.	
		1	Force load. The INTVALn interval value IVALUE -1 is immediately loaded into the TIMERn register while TIMERn is running.	

25.6.2 Timer register

The timer register holds the current timer value. This register is read-only.

Table 437. Timer register (TIMER[0:3], address 0x4000 4004 (TIMER0) to 0x4000 4034 (TIMER3)) bit description

Bit	Symbol	Description	Reset value
30:0	VALUE	Holds the current timer value of the down counter. The initial value of the TIMERn register is loaded as IVALUE - 1 from the INTVALn register either at the end of the time interval or immediately in the following cases: INTVALn register is updated in the idle state. INTVALn register is updated with LOAD = 1. When the timer is in idle state, reading this bit fields returns -1 (0x7FFF FFFF).	0x7FFF FFFF
31	-	Reserved.	0

25.6.3 Control register

The control register configures the mode for each MRT and enables the interrupt.

Table 438. Control register (CTRL[0:3], address 0x4000 4008 (CTRL0) to 0x4000 4038 (CTRL3)) bit description

Bit	Symbol	Value	Description	Reset value
0	INTEN		Enable the TIMERn interrupt.	0
		0	Disable.	
		1	Enable.	
2:1	MODE		Selects timer mode.	0
		0x0	Repeat interrupt mode.	
		0x1	One-shot interrupt mode.	
		0x2	One-shot bus stall mode.	
		0x3	Reserved.	
31:3	-		Reserved.	0

25.6.4 Status register

This register indicates the status of each MRT.

Table 439. Status register (STAT[0:3], address 0x4000 400C (STAT0) to 0x4000 403C (STAT3)) bit description

Bit	Symbol	Value	Description	Reset value
0	INTFLAG		Monitors the interrupt flag.	0
		0	No pending interrupt. Writing a zero is equivalent to no operation.	
		1	Pending interrupt. The interrupt is pending because TIMERn has reached the end of the time interval. If the INTEN bit in the CONTROLn is also set to 1, the interrupt for timer channel n and the global interrupt are raised. Writing a 1 to this bit clears the interrupt request.	
1	RUN		Indicates the state of TIMERn. This bit is read-only.	0
		0	Idle state. TIMERn is stopped.	
		1	Running. TIMERn is running.	
31:2	-		Reserved.	0

25.6.5 Idle channel register

The idle channel register returns the lowest idle channel number. The channel is considered idle when both flags is the STATUS register (RUN and INTFLAG) are zero.

In an application with multiple timers running independently, you can calculate the register offset of the next idle timer by reading the idle channel number in this register. The idle channel register allows you set up the next idle timer without checking the idle state of each timer.

Table 440. Idle channel register (IDLE_CH, address 0x4000 40F4) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved.	0
7:4	CHAN	Idle channel. Reading the CHAN bits, returns the lowest idle timer channel. If all timer channels are running, CHAN = 4.	0
31:8	-	Reserved.	0

25.6.6 Global interrupt flag register

The global interrupt register combines the interrupt flags from the individual timer channels in one register. Setting and clearing each flag behaves in the same way as setting and clearing the INTFLAG bit in each of the STATUSn registers.

Table 441. Global interrupt flag register (IRQ_FLAG, address 0x4000 40F8) bit description

Bit	Symbol	Value	Description	Reset value
0	GFLAG0		Monitors the interrupt flag of TIMER0.	0
		0	No pending interrupt. Writing a zero is equivalent to no operation.	
		1	Pending interrupt. The interrupt is pending because TIMER0 has reached the end of the time interval. If the INTEN bit in the CONTROL0 register is also set to 1, the interrupt for timer channel 0 and the global interrupt are raised. Writing a 1 to this bit clears the interrupt request.	
1	GFLAG1		Monitors the interrupt flag of TIMER1.	0
		0	No pending interrupt. Writing a zero is equivalent to no operation.	
		1	Pending interrupt. The interrupt is pending because TIMER1 has reached the end of the time interval. If the INTEN bit in the CONTROL1 register is also set to 1, the interrupt for timer channel 1 and the global interrupt are raised. Writing a 1 to this bit clears the interrupt request.	
2	GFLAG2		Monitors the interrupt flag of TIMER2.	0
		0	No pending interrupt. Writing a zero is equivalent to no operation.	
		1	Pending interrupt. The interrupt is pending because TIMER2 has reached the end of the time interval. If the INTEN bit in the CONTROL2 register is also set to 1, the interrupt for timer channel 2 and the global interrupt are raised. Writing a 1 to this bit clears the interrupt request.	
3	GFLAG3		Monitors the interrupt flag of TIMER3.	0
		0	No pending interrupt. Writing a zero is equivalent to no operation.	
		1	Pending interrupt. The interrupt is pending because TIMER3 has reached the end of the time interval. If the INTEN bit in the CONTROL3 register is also set to 1, the interrupt for timer channel 3 and the global interrupt are raised. Writing a 1 to this bit clears the interrupt request.	
31:4	-		Reserved.	0

26.1 How to read this chapter

The SysTick timer is available on all LPC86x parts.

26.2 Features

- Simple 24-bit timer.
- Uses dedicated exception vector.
- Clocked internally by the system clock or the system clock/2.

26.3 Basic configuration

The system tick timer is configured using the following registers:

1. The system tick timer is enabled through the SysTick control register (Table 443). The system tick timer clock is fixed to half of the system clock frequency.
2. Enable the clock source for the SysTick timer in the SYST_CSR register (Table 443).
3. The calibration value of the SysTick timer is contained in the SYSTCKCAL register in the system configuration block SYSCON (see Table 110).

26.4 Pin description

The SysTick has no configurable pins.

26.5 General description

The block diagram of the SysTick timer is shown in Figure 162.

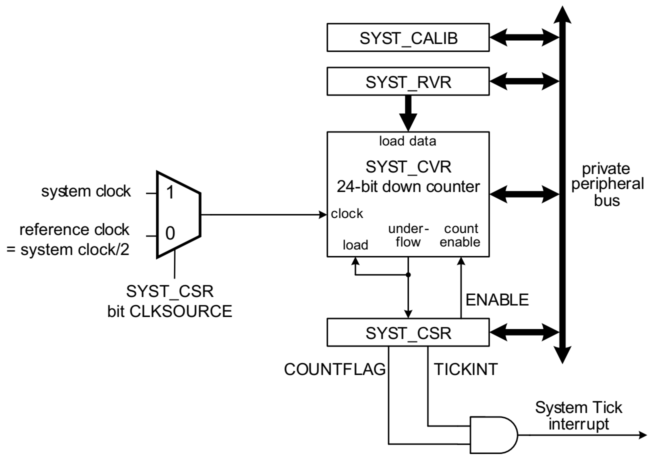


Fig 162. System tick timer block diagram

The SysTick timer is an integral part of the Cortex-M0+. The SysTick timer is intended to generate a fixed 10 millisecond interrupt for use by an operating system or other system management software.

Since the SysTick timer is a part of the Cortex-M0+, it facilitates porting of software by providing a standard timer that is available on Cortex-M0 based devices. The SysTick timer can be used for:

- An RTOS tick timer which fires at a programmable rate (for example 100 Hz) and invokes a SysTick routine.
- A high-speed alarm timer using the core clock.
- A simple counter. Software can use this to measure time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

Refer to Ref. 5 for details.

26.6 Register description

The SysTick timer registers are located on the ARM Cortex-M0+ private peripheral bus (see Figure 2), and are part of the ARM Cortex-M0+ core peripherals. For details, see Ref. 5.

Table 442. Register overview: SysTick timer (base address 0xE000 E000)

Name	Access	Address offset	Description	Reset value[1]
SYST_CSR	R/W	0x010	System Timer Control and status register	0x000 0000
SYST_RVR	R/W	0x014	System Timer Reload value register	0x00FF FFFF
SYST_CVR	R/W	0x018	System Timer Current value register	0x00FF FFFF
SYST_CALIB	R/W	0x01C	System Timer Calibration value register	0

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

26.6.1 System Timer Control and status register

The SYST_CSR register contains control information for the SysTick timer and provides a status flag. This register is part of the ARM Cortex-M0+ core system timer register block. For a bit description of this register, see Ref. 5.

This register determines the clock source for the system tick timer.

Table 443. SysTick Timer Control and status register (SYST_CSR, 0xE000 E010) bit description

Bit	Symbol	Description	Reset value
0	ENABLE	System Tick counter enable. When 1, the counter is enabled. When 0, the counter is disabled.	0
1	TICKINT	System Tick interrupt enable. When 1, the System Tick interrupt is enabled. When 0, the System Tick interrupt is disabled. When enabled, the interrupt is generated when the System Tick counter counts down to 0.	0
2	CLKSOURCE	System Tick clock source selection. When 1, the system clock (CPU) clock is selected. When 0, the system clock/2 is selected as the reference clock.	0
15:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
16	COUNTFLAG	Returns 1 if the SysTick timer counted to 0 since the last read of this register.	0
31:17	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

26.6.2 System Timer Reload value register

The SYST_RVR register is set to the value that will be loaded into the SysTick timer whenever it counts down to zero. This register is loaded by software as part of timer initialization. The SYST_CALIB register may be read and used as the value for SYST_RVR register if the CPU is running at the frequency intended for use with the SYST_CALIB value.

Table 444. System Timer Reload value register (SYST_RVR, 0xE000 E014) bit description

Bit	Symbol	Description	Reset value
23:0	RELOAD	This is the value that is loaded into the System Tick counter when it counts down to 0.	0x00FF FFFF
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

26.6.3 System Timer Current value register

The SYST_CVR register returns the current count from the System Tick counter when it is read by software.

Table 445. System Timer Current value register (SYST_CVR, 0xE000 E018) bit description

Bit	Symbol	Description	Reset value
23:0	CURRENT	Reading this register returns the current value of the System Tick counter. Writing any value clears the System Tick counter and the COUNTFLAG bit in STCTRL.	0x00F F FFFF
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

26.6.4 System Timer Calibration value register

The value of the SYST_CALIB register is driven by the value of the SYSTCKCAL register in the system configuration block SYSCON (see Table 110).

Table 446. System Timer Calibration value register (SYST_CALIB, 0xE000 E01C) bit description

Bit	Symbol	Value	Description	Reset value
23:0	TENMS		See Ref. 5.	0
29:24	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
30	SKEW		See Ref. 5.	0
31	NOREF		See Ref. 5.	0

26.7 Functional description

The SysTick timer is a 24-bit timer that counts down to zero and generates an interrupt. The intent is to provide a fixed 10 millisecond time interval between interrupts. The SysTick timer is clocked from the CPU clock (the system clock, see Figure 7) or from the reference clock, which is fixed to half the frequency of the CPU clock. In order to generate recurring interrupts at a specific interval, the SYST_RVR register must be initialized with the correct value for the desired interval. A default value is provided in the SYST_CALIB register and may be changed by software.

26.7.1 Example timer calculation

To use the system tick timer, do the following:

1. Program the SYST_RVR register with the reload value RELOAD to obtain the desired time interval.
2. Clear the SYST_CVR register by writing to it. This ensures that the timer will count from the SYST_RVR value rather than an arbitrary value when the timer is enabled.
3. Program the SYST_SCR register with the value 0x7 which enables the SysTick timer and the SysTick timer interrupt.

The following example illustrates selecting the SysTick timer reload value to obtain a 10 ms time interval with the system clock set to 20 MHz.

Example (system clock = 20 MHz)

The system tick clock = system clock = 20 MHz. Bit CLKSOURCE in the SYST_CSR register set to 1 (system clock).

RELOAD = (system tick clock frequency $\times$ 10 ms) –1 = (20 MHz $\times$ 10 ms) –1 = 200000 –1 = 199999 = 0x00030D3F.

UM11607

Chapter 27: 12-bit Analog-to-Digital Converter (ADC)

Rev. 2 — April 2023 **User manual**

27.1 How to read this chapter

The ADC is available on all parts. The number of available ADC channels depends on the package type.

Table 447. Pinout summary

Package	ADC channels available
HVQFN48	ADC_0 to ADC_11
LQFP64	ADC_0 to ADC_11
HVQFN32	ADC_0 to ADC_11

27.2 Features

- 12-bit successive approximation analog to digital converter.
- Input multiplexing among 12 pins.
- Two configurable conversion sequences with independent triggers.
- Optional automatic high/low threshold comparison and “zero crossing” detection.
- Power-down mode and low-power operating mode.
- Measurement range VREFN to VREFP (typically 3 V; not to exceed VDDA voltage level).
- 12-bit conversion rate of up to 1.9 Msamples/s.
- Burst conversion mode for single or multiple inputs.
- DMA support.
- Hardware calibration mode.

27.3 Basic configuration

Configure the ADC as follows:

- Use the PDRUNCFG register to power the ADC. See Table 118. Once the ADC is powered by the PDRUNCFG register bit, the low-power mode bit in the ADC CTRL register can be used to turn off the ADC when it is not sampling and turn on the ADC automatically when any of the ADC conversion triggers are raised. See Table 451 and Section 27.7.5.
- Use the SYSAHBCLKCTRL register (Table 94) to enable the clock to the ADC register interface and the ADC clock.
- The ADC block creates four interrupts with individual entries in the NVIC. See Table 46.
- The ADC analog inputs are enabled in the switch matrix block.See Table 139.
- The power to the ADC block is controlled by the PDRUNCFG register in the SYSCON block. See Table 118.

- Calibration is required after every power-up or wake-up from Deep power-down mode. See Section 27.3.4 “Hardware self-calibration”.
- For a sampling rate higher than 1 Msamples/s, VDDA must be higher than 2.7 V. See Table 451.
- Configure the ADC for the appropriate analog supply voltage using the TRM register (Table 464). The default setting assumes $V_{DDA} \geq 2.7$ V.

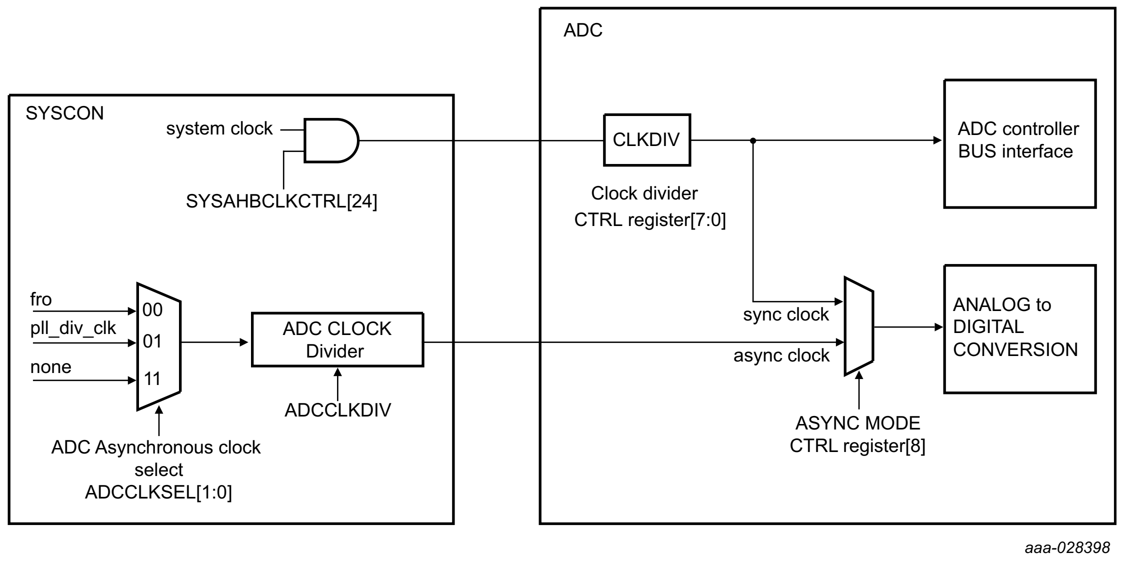

SYSCON
system clock
SYSAHBCLKCTRL[24]
fro
pll_div_clk
none
00
01
11
ADC CLOCK Divider
ADCCLKDIV
ADC Asynchronous clock select ADCCLKSEL[1:0]
ADC
CLKDIV
Clock divider CTRL register[7:0]
ADC controller BUS interface
sync clock
async clock
ANALOG to DIGITAL CONVERSION
ASYNC MODE CTRL register[8]
aaa-028398

Fig 163. ADC clocking

27.3.1 Perform a single ADC conversion using a software trigger

Remark: When A/D conversions are triggered by software only and hardware triggers are not used in the conversion sequence, set the trigger source in the SEQA_CTRL and SEQB_CTRL registers to 0x0 (default).

Once the sequence is enabled, the ADC converts a sample whenever the START bit is written to. The TRIGPOL bit can be set in the same write that sets the SEQ_ENA and the START bits. Be careful not to modify the TRIGGER, TRIGPOL, and SEQ_ENA bits on subsequent writes to the START bit. See also Section 27.7.2.1 “Avoiding spurious hardware triggers”.

The ADC converts an analog input signal VIN on the ADC_[11:0]. The VREFP and VREFN pins provide a positive and negative reference voltage input. The result of the conversion is (4095 x VIN- VREFN)/(VREFP - VREFN). The result of an input voltage below VREFN is 0, and the result of an input voltage above VREFP is 4095 (0xFFF).

To perform a single ADC conversion for ADC channel 1 using the analog signal on pin ADC_1, follow these steps:

1. Enable the analog function ADC_1.
2. Configure the system clock to be 25 MHz and select a CLKDIV value of 0 for a sampling rate of 1 Msamples/s using the ADC CTRL register.

3. Select ADC channel 1 to perform the conversion by setting the CHANNELS bits to 0x2 in the SEQA_CTL register.
4. Set the TRIGPOL bit to 1 and the SEQA_ENA bit to 1 in the SEQA_CTRL register.
5. Set the START bit to 1 in the SEQA_CTRL register.
6. Read the RESULT bits in the DAT1 register for the conversion result.

27.3.2 Perform a sequence of conversions triggered by an external pin

The ADC can perform conversions on a sequence of selected channels. Each individual conversion of the sequence (single-step) or the entire sequence can be triggered by hardware. Hardware triggers are either a signal from an external pin or an internal signal. See Section 27.3.3.

To perform a single-step conversion on the first four channels of ADC triggered by a rising edge on PINT0 or PINT1 pin, follow these steps:

1. Enable the analog functions ADC_0 to ADC_3 through the switch matrix. See Table 449.
2. Configure the system clock to be 25 MHz and select a CLKDIV value of 0 for a sampling rate of 1 Msamples/s using the ADC CTRL register.
3. Select ADC channels 0 to 3 to perform the conversion by setting the CHANNELS bits to 0xF in the SEQA_CTL register.
4. Assign the input port PIO0_15 to be pin interrupt 0 by writing 0xF to PINTSEL[0] in SYSCON.
5. Configure the pin interrupt block for level-sensitive and active-high on pin interrupt 0, and enable it.
   – LPC_PIN_INT->ISEL |= 0x1; // level-sensitive
   – LPC_PIN_INT->IENF |= 0x1; // active high
   – LPC_PIN_INT->SIENR = 0x1; // enabled
6. Select PININT0_IRQ by writing 0x1 to the TRIGGER bits in the SEQA_CTRL register.
7. To generate one interrupt at the end of the entire sequence, set the MODE bit to 1 in the SEQA_CTRL register.
8. Select single-step mode by setting the SINGLESTEP bit in the SEQA_CTRL register to 1.
9. Enable the Sequence A by setting the SEQA_ENA bit.

   A conversion on ADC channel 0 will be triggered whenever the pin PIO0_15 goes from LOW to HIGH. The conversion on the next channel (channel 1) is triggered on the next rising edge of PIO0_15. The ADC_SEQA_IRQ interrupt is generated when the sequence has finished after four rising edges on PIO0_15.
10. Read the RESULT bits in the DAT0 to DAT3 registers for the conversion result.

27.3.3 ADC hardware trigger inputs

An analog-to-digital conversion can be initiated by a hardware trigger. You can select the trigger independently for each of the two conversion sequences in the ADC SEQA_CTRL or SEQB_CTRL registers by programming the hardware trigger input # into the TRIGGER bits.

See Table 64 “ADC trigger inputs” ADC hardware trigger input source selection.

Related registers:

- Table 452 “A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description”
- Table 453 “A/D Conversion Sequence B Control Register (SEQB_CTRL, address 0x4001 0x00C)bit description”

27.3.4 Hardware self-calibration

The A/D converter includes a built-in, hardware self-calibration mode. In order to achieve the specified ADC accuracy, the A/D converter must be recalibrated, at a minimum, following every chip reset before initiating normal ADC operation.

The calibration voltage level is VREFP - VREFN.

To calibrate the ADC follow these steps:

1. Save the current contents of the ADC CTRL register if different from default.
2. In a single write to the ADC CTRL register, do the following to start the calibration:
   – Set the calibration mode bit CALMODE.
   – Write a divider value to the CLKDIV bit field that divides the system clock to yield an ADC clock of about 500 kHz.
   – Clear the LPWR bit.
3. Poll the CALMODE bit until it is cleared.

Before launching a new A/D conversion, restore the contents of the CTRL register or use the default values.

A calibration cycle requires approximately 290 μs to complete. While calibration is in progress, normal ADC conversions cannot be launched, and the ADC Control Register must not be written to. The calibration procedure does not use the CPU or memory, so other processes can be executed during calibration.

27.4 Pin description

The ADC cell can measure the voltage on any of the input signals on the analog input channel. Digital signals are disconnected from the ADC input pins when the ADC function is selected on that pin in the IOCON register.

Remark: If the ADC is used, signal levels on analog input pins must not be above the level of V_{DD} at any time. Otherwise, ADC readings will be invalid. If the ADC is not used in an application, then the pins associated with ADC inputs can be configured as digital I/O pins and are 5 V tolerant.

The VREFP and VREFN pins provide a positive and negative reference voltage input. The result of the conversion is (4095 x input voltage VIN)/(VREFP - VREFN). The result of an input voltage below VREFN is 0, and the result of an input voltage above VREFP is 4095 (0xFFF).

When the ADC is not used, tie VREFP to VDD and VREFP to V_{SS}.

Remark: For best performance, select VREFP and VREFN at the same voltage levels as V_{DD} and V_{SS}. When selecting VREFP and VREFN different from VDD and VSS, ensure that the voltage midpoints are the same:

(VREFP-VREFN)/2 + VREFN = $V_{DD}/2$

Table 448. ADC supply and reference voltage pins

Function	Description
V_{REFP}	Positive voltage reference. The VREFP voltage level must be between 2.4 V and V_{DDA}. For best performance, select VREFP = V_{DDA} and VREFN = V_{SSA}.
V_{REFN}	Negative voltage reference.
V_{DDA} = V_{DD}	The analog supply voltage is internally connected to V_{DD}.
V_{SSA} = V_{SS}	ADC ground is internally connected to VSS.

Table 449. ADC pin description

Function	Direction	Type	Connect to	Use register	Reference	Description
ADC_0	AI	external to pin	PIO0_7	PINENABLE0	Table 139	Analog input channel 0.
ADC_1	AI	external to pin	PIO0_6	PINENABLE0	Table 139	Analog input channel 1.
ADC_2	AI	external to pin	PIO0_14	PINENABLE0	Table 139	Analog input channel 2.
ADC_3	AI	external to pin	PIO0_23	PINENABLE0	Table 139	Analog input channel 3.
ADC_4	AI	external to pin	PIO0_22	PINENABLE0	Table 139	Analog input channel 4.
ADC_5	AI	external to pin	PIO0_21	PINENABLE0	Table 139	Analog input channel 5.
ADC_6	AI	external to pin	PIO0_20	PINENABLE0	Table 139	Analog input channel 6.
ADC_7	AI	external to pin	PIO0_19	PINENABLE0	Table 139	Analog input channel 7.
ADC_8	AI	external to pin	PIO0_18	PINENABLE0	Table 139	Analog input channel 8.
ADC_9	AI	external to pin	PIO0_17	PINENABLE0	Table 139	Analog input channel 9.
ADC_10	AI	external to pin	PIO0_13	PINENABLE0	Table 139	Analog input channel 10.
ADC_11	AI	external to pin	PIO0_4	PINENABLE0	Table 139	Analog input channel 11.

27.4.1 ADC vs. digital receiver

The ADC function must be selected via the switch matrix registers in order to get accurate voltage readings on the monitored pin. The MODE bits in the IOCON register should also disable both pull-up and pull-down resistors. For a pin hosting an ADC input, it is not possible to have a have a digital function selected and yet get valid ADC readings. An inside circuit disconnects ADC hardware from the associated pin whenever a digital function is selected on that pin.

27.5 General description

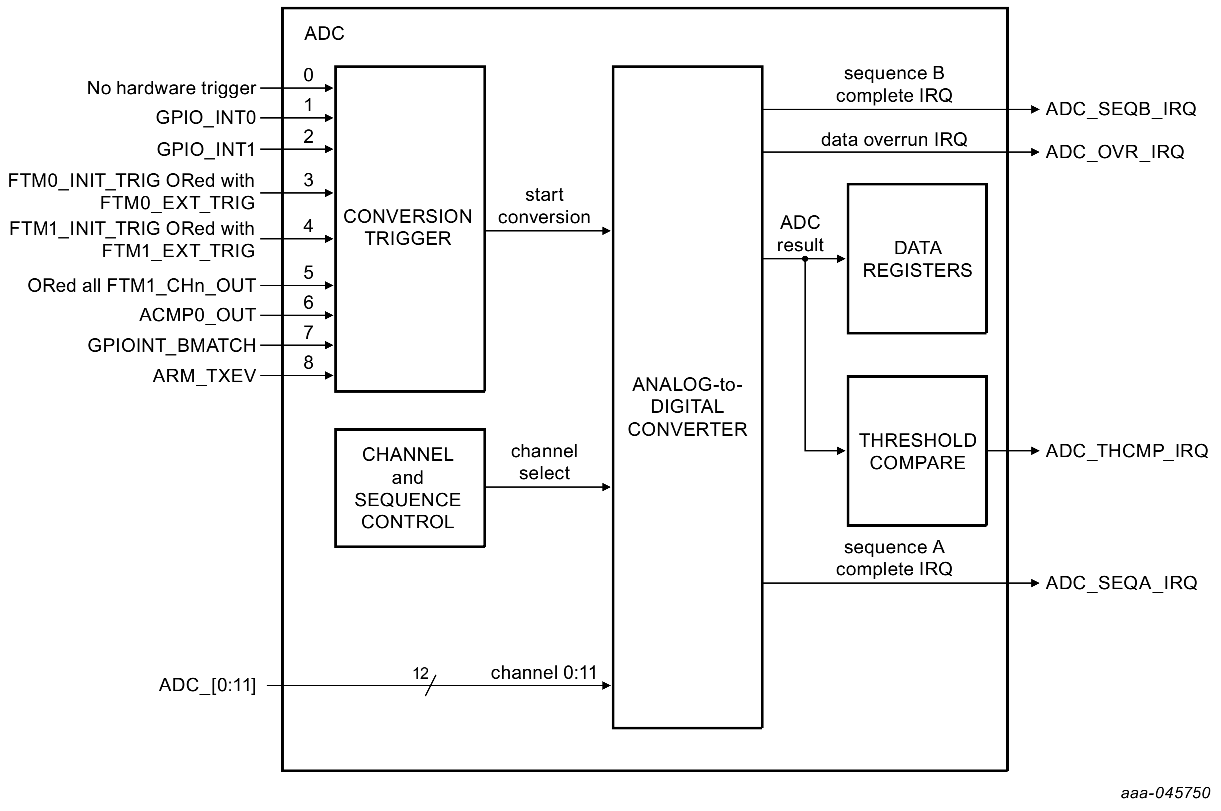


Fig 164. ADC block diagram

The ADC controller provides great flexibility in launching and controlling sequences of A/D conversions using the associated 12-bit, successive approximation A/D converter. A/D conversion sequences can be initiated under software control or in response to a selected hardware trigger. The ADC supports eight hardware triggers.

Once the triggers are set up (software and hardware triggers can be mixed), the ADC runs through the pre-defined conversion sequence, converting a sample whenever a trigger signal arrives, until the sequence is disabled.

The ADC controller uses the system clock as a bus clock. The ADC clock is derived from the system clock. A programmable divider is included to scale the system clock to the maximum ADC clock rate of 48 MHz. The ADC clock drives the successive approximation process.

A fully accurate conversion requires 25 of these ADC clocks.

27.5.1 Sample Time

The analog input from the selected channel is sampled at the start of each new A/D conversion. The default (and shortest) duration of the sample period is 6.5 ADC clocks. With this default setting, the total duration of each A/D conversion is 25 ADC clocks.

Under certain conditions longer sample times may be required. A variety of factors influence the required sampling time. These factors include: the operating conditions, the ADC clock frequency, and, most importantly, the output impedance of the analog source driver. Use the TSAMP value in the SEQn_CTRL register to increase the sampling time.

27.6 Register description

The reset value reflects the data stored in used bits only. It does not include reserved bits content.

Table 450. Register overview : ADC (base address 0x4001 C000)

Name	Access	Address offset	Description	Reset value	Reference
CTRL	R/W	0x000	A/D Control Register. Contains the clock divide value, enable bits for each sequence and the A/D power-down bit.	0x0	Table 451
-	-	0x004	Reserved.	-	-
SEQA_CTRL	R/W	0x008	A/D Conversion Sequence-A control Register: Controls triggering and channel selection for conversion sequence-A. Also specifies interrupt mode for sequence-A.	0x0	Table 452
SEQB_CTRL	R/W	0x00C	A/D Conversion Sequence-B Control Register: Controls triggering and channel selection for conversion sequence-B. Also specifies interrupt mode for sequence-B.	0x0	Table 453
SEQA_GDAT	RO	0x010	A/D Sequence-A Global Data Register. This register contains the result of the most recent A/D conversion performed under sequence-A	NA	Table 454
SEQB_GDAT	RO	0x014	A/D Sequence-B Global Data Register. This register contains the result of the most recent A/D conversion performed under sequence-B	NA	Table 455
DAT0	RO	0x020	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0.	NA	Table 456
DAT1	RO	0x024	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.	NA	Table 456
DAT2	RO	0x028	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.	NA	Table 456
DAT3	RO	0x02C	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.	NA	Table 456
DAT4	RO	0x030	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.	NA	Table 456
DAT5	RO	0x034	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.	NA	Table 456
DAT6	RO	0x038	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.	NA	Table 456
DAT7	RO	0x03C	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	Table 456

Table 450. Register overview : ADC (base address 0x4001 C000)

Name	Access	Address offset	Description	Reset value	Reference
DAT8	RO	0x040	A/D Channel 8 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	Table 456
DAT9	RO	0x044	A/D Channel 9 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	Table 456
DAT10	RO	0x048	A/D Channel 10 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	Table 456
DAT11	RO	0x04C	A/D Channel 11 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	Table 456
THR0_LOW	R/W	0x050	A/D Low Compare Threshold Register 0 : Contains the lower threshold level for automatic threshold comparison for any channels linked to threshold pair 0.	0x0	Table 457
THR1_LOW	R/W	0x054	A/D Low Compare Threshold Register 1: Contains the lower threshold level for automatic threshold comparison for any channels linked to threshold pair 1.	0x0	Table 458
THR0_HIGH	R/W	0x058	A/D High Compare Threshold Register 0: Contains the upper threshold level for automatic threshold comparison for any channels linked to threshold pair 0.	0x0	Table 459
THR1_HIGH	R/W	0x05C	A/D High Compare Threshold Register 1: Contains the upper threshold level for automatic threshold comparison for any channels linked to threshold pair 1.	0x0	Table 460
CHAN_THRSEL	R/W	0x060	A/D Channel-Threshold Select Register. Specifies which set of threshold compare registers are to be used for each channel	0x0	Table 461
INTEN	R/W	0x064	A/D Interrupt Enable Register. This register contains enable bits that enable the sequence-A, sequence-B, threshold compare and data overrun interrupts to be generated.	0x0	Table 462
FLAGS	R/W	0x068	A/D Flags Register. Contains the four interrupt request flags and the individual component overrun and threshold-compare flags. (The overrun bits replicate information stored in the result registers).	0x0	Table 463
TRM	R/W	0x06C	ADC trim register.	0x0	Table 464

27.6.1 ADC Control Register

This register specifies the clock divider value to be used to generate the ADC clock and general operating mode bits including a low power mode that allows the A/D to be turned off to save power when not in use.

Table 451. A/D Control Register (CTRL, addresses 0x4001 C000) bit description

Bit	Symbol	Value	Description	Reset value
7:0	CLKDIV		The system clock is divided by this value plus one to produce the sampling clock. The sampling clock should be less than or equal to 48 MHz for 1.9 Msamples/s. Typically, software should program the smallest value in this field that yields this maximum clock rate or slightly less, but in certain cases (such as a high-impedance analog source) a slower clock may be desirable.	0
8	ASYNCMODE		Asynchronous operation mode	0
		0	Synchronous mode. The ADC clock is derived from the main system clock based on the divide value selected in the CLKDIV field. The ADC clock starts in response to a trigger to eliminate any uncertainty in the launching of an ADC conversion in response to any synchronous (on-chip) trigger. In synchronous mode with the SYNC-BYPASS bit set, sampling of the A/D input and start of a conversion initiates two system clocks after the leading edge of a (synchronous) trigger pulse.	
		1	Asynchronous mode. The ADC clock is based on an alternative independent clock source. The nature of this clock source and the mechanism for programming it is chip-specific. The frequency of this clock is limited to 15 MHz max. In addition, the ADC clock must never be faster than 10 times the APB bus clock rate.	
9	-		Reserved. Do not write a one to these bits.	0
10	LPWRMODE		Select low-power ADC mode. The analog circuitry is automatically powered-down when no conversions are taking place. When any (hardware or software) triggering event is detected, the analog circuitry is enabled. After the required start-up time, the requested conversion will be launched. Once the conversion completes, the analog-circuitry will again be powered-down provided no further conversions are pending. Using this mode can save an appreciable amount of current when conversions are required relatively infrequently. The penalty for using this mode is an approximately 15 ADC clock delay, based on the frequency specified in the CLKDIV field, from the time the trigger event occurs until sampling of the A/D input commences. **Remark:** This mode will NOT power-up the ADC when the ADC analog block is powered down in the system control block.	0
		0	Disabled. The low-power ADC mode is disabled. The analog circuitry remains activated even when no conversions are requested.	
		1	Enabled. The low-power ADC mode is enabled.	
29:11			Reserved, do not write ones to reserved bits.	0
30	CALMODE		Writing a 1 to this bit initiates a self-calibration cycle. This bit will be automatically cleared by hardware after the calibration cycle is complete. To calibrate the ADC, set the ADC clock to 500 kHz. **Remark:** Other bits of this register may be written to concurrently with setting this bit, however once this bit has been set no further writes to this register are permitted until the full calibration cycle has ended.	0
31	-		Reserved.	0

27.6.2 A/D Conversion Sequence A Control Register

There are two, independent conversion sequences that can be configured, each consisting of a set of conversions on one or more channels. This control register specifies the channel selection and trigger conditions for the A sequence and contains bits to allow software to initiate that conversion sequence.

To avoid conversions on spurious triggers, only change the trigger configuration when the conversion sequence is disabled. A conversion can be triggered by software or hardware in the conversion sequence, but if conversions are triggered by software only, spurious hardware triggers must be prevented. See Section 27.3.1 “Perform a single ADC conversion using a software trigger”.

Remark: Set the BURST and SEQU_ENA bits at the same time.

Table 452. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
11:0	CHANNELS		Selects which one or more of the twelve channels will be sampled and converted when this sequence is launched. A 1 in any bit of this field will cause the corresponding channel to be included in the conversion sequence, where bit 0 corresponds to channel 0, bit 1 to channel 1 and so forth. When this conversion sequence is triggered, either by a hardware trigger or via software command, A/D conversions will be performed on each enabled channel, in sequence, beginning with the lowest-ordered channel. **Remark:** This field can ONLY be changed while the SEQA_ENA bit (bit 31) is LOW. It is allowed to change this field and set bit 31 in the same write.	0x00
15:12	TRIGGER		Selects which of the available hardware trigger sources will cause this conversion sequence to be initiated. Program the trigger input number in this field.See Table 64 “ADC trigger inputs” for details. **Remark:** In order to avoid generating a spurious trigger, it is recommended writing to this field only when the SEQA_ENA bit (bit 31) is low. It is safe to change this field and set bit 31 in the same write.	0x0
17:16	-		Reserved.	-
18	TRIGPOL		Select the polarity of the selected input trigger for this conversion sequence. **Remark:** In order to avoid generating a spurious trigger, it is recommended writing to this field only when the SEQA_ENA bit (bit 31) is low. It is safe to change this field and set bit 31 in the same write.	0
		0	Negative edge. A negative edge launches the conversion sequence on the selected trigger input.	
		1	Positive edge. A positive edge launches the conversion sequence on the selected trigger input.	

Table 452. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
19	SYNCBYPASS		Setting this bit allows the hardware trigger input to bypass synchronization flip-flops stages and therefore shorten the time between the trigger input signal and the start of a conversion. There are slightly different criteria for whether or not this bit can be set depending on the clock operating mode: Synchronous mode: Synchronization may be bypassed (this bit may be set) if the selected trigger source is already synchronous with the main system clock (eg. coming from an on-chip, system-clock-based timer). Whether this bit is set or not, a trigger pulse must be maintained for at least one system clock period. Asynchronous mode: Synchronization may be bypassed (this bit may be set) if it is certain that the duration of a trigger input pulse will be at least one cycle of the ADC clock (regardless of whether the trigger comes from and on-chip or off-chip source). If this bit is NOT set, the trigger pulse must at least be maintained for one system clock period.	0
		0	Enable synchronization. The hardware trigger bypass is not enabled.	
		1	Bypass synchronization. The hardware trigger bypass is enabled.	
25	-		Reserved	-
24-20	TSAMP		The default sample period (TSAMP = “00000”) at the beginning of each new conversion is 6.5 ADC clock periods. Depending on a variety of factors including ADC clock rate, output impedance of the analog source driver, ADC resolution, and the selection of channels, the sample time may need to be increased. The value programmed into the TSAMP fields dictates the number of additional ADC clock cycles (beyond 6.5) that the sample period will be extended by. Note: Any additional clocks of sample time inserted will add directly to the overall number of clocks required for a conversion, effectively reducing the overall conversion throughput rate.	0
26	START		Writing a 1 to this field will launch one pass through this conversion sequence. The behavior will be identical to a sequence triggered by a hardware trigger. Do not write 1 to this bit if the BURST bit is set. **Remark:** This bit is only set to a 1 momentarily when written to launch a conversion sequence. It will consequently always read-back as a zero.	0
27	BURST		Writing a 1 to this bit will cause this conversion sequence to be continuously cycled through. Other sequence A triggers will be ignored while this bit is set. Repeated conversions can be halted by clearing this bit. The sequence currently in progress will be completed before conversions are terminated.	0
28	SINGLESTEP		When this bit is set, a hardware trigger or a write to the START bit will launch a single conversion on the next channel in the sequence instead of the default response of launching an entire sequence of conversions. Once all of the channels comprising a sequence have been converted, a subsequent trigger will repeat the sequence beginning with the first enabled channel. Interrupt generation will still occur either after each individual conversion or at the end of the entire sequence, depending on the state of the MODE bit.	0

Table 452. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
29	LOWPRIO		Set priority for sequence B.	0
		0	Low priority. Any B trigger which occurs while an A conversion sequence is active will be ignored and lost.	
		1	High priority. Setting this bit to a 1 will permit any enabled B sequence trigger (including a B sequence software start) to immediately interrupt this sequence and launch a B sequence in it's place. The conversion currently in progress will be terminated. The A sequence that was interrupted will automatically resume after the B sequence completes. The channel whose conversion was terminated will be re-sampled and the conversion sequence will resume from that point.	
30	MODE		Indicates whether the primary method for retrieving conversion results for this sequence will be accomplished via reading the global data register (SEQA_GDAT) at the end of each conversion, or the individual channel result registers at the end of the entire sequence. Impacts when conversion-complete interrupt/DMA triggers for sequence-A will be generated and which overrun conditions contribute to an overrun interrupt as described below:	0
		0	End of conversion. The sequence A interrupt/DMA flag will be set at the end of each individual A/D conversion performed under sequence A. This flag will mirror the DATAVALID bit in the SEQA_GDAT register. The OVERRUN bit in the SEQA_GDAT register will contribute to generation of an overrun interrupt if enabled.	
		1	End of sequence. The sequence A interrupt/DMA flag will be set when the entire set of sequence-A conversions completes. This flag will need to be explicitly cleared by software or by the DMA-clear signal in this mode. The OVERRUN bit in the SEQA_GDAT register will NOT contribute to generation of an overrun interrupt/DMA trigger since it is assumed this register may not be utilized in this mode.	
31	SEQA_ENA		Sequence Enable. In order to avoid spuriously triggering the sequence, care should be taken to only set the SEQA_ENA bit when the selected trigger input is in its INACTIVE state (as defined by the TRIGPOL bit). If this condition is not met, the sequence will be triggered immediately upon being enabled.	0
		0	Disabled. Sequence A is disabled. Sequence A triggers are ignored. If this bit is cleared while sequence A is in progress, the sequence will be halted at the end of the current conversion. After the sequence is re-enabled, a new trigger will be required to restart the sequence beginning with the next enabled channel.	
		1	Enabled. Sequence A is enabled.	

27.6.3 A/D Conversion Sequence B Control Register

There are two, independent conversion sequences that can be configured, each consisting of a set of conversions on one or more channels. This control register specifies the channel selection and trigger conditions for the B sequence, as well bits to allow software to initiate that conversion sequence.

To avoid conversions on spurious triggers, only change the trigger configuration when the conversion sequence is disabled. A conversion can be triggered by software or hardware in the conversion sequence, but if conversions are triggered by software only, spurious hardware triggers must be prevented. See Section 27.3.1 “Perform a single ADC conversion using a software trigger”.

Remark: Set the BURST and SEQU_ENA bits at the same time.

Table 453. A/D Conversion Sequence B Control Register (SEQB_CTRL, address 0x4001 0x00C)bit description

Bit	Symbol	Value	Description	Reset value
11:0	CHANNELS		Selects which one or more of the twelve channels will be sampled and converted when this sequence is launched. A 1 in any bit of this field will cause the corresponding channel to be included in the conversion sequence, where bit 0 corresponds to channel 0, bit 1 to channel 1 and so forth. When this conversion sequence is triggered, either by a hardware trigger or via software command, A/D conversions will be performed on each enabled channel, in sequence, beginning with the lowest-ordered channel. **Remark:** This field can ONLY be changed while the SEQB_ENA bit (bit 31) is LOW. It is permissible to change this field and set bit 31 in the same write.	0x00
15:12	TRIGGER		Selects which of the available hardware trigger sources will cause this conversion sequence to be initiated. Program the trigger input number in this field. See Figure 164 “ADC block diagram” for details. **Remark:** In order to avoid generating a spurious trigger, it is recommended writing to this field only when the SEQA_ENA bit (bit 31) is low. It is safe to change this field and set bit 31 in the same write.	0x0
17:16	-		Reserved.	-
18	TRIGPOL		Select the polarity of the selected input trigger for this conversion sequence. **Remark:** In order to avoid generating a spurious trigger, it is recommended writing to this field only when the SEQA_ENA bit (bit 31) is low. It is safe to change this field and set bit 31 in the same write.	0
		0	Negative edge. A negative edge launches the conversion sequence on the selected trigger input.	
		1	Positive edge. A positive edge launches the conversion sequence on the selected trigger input.	

Table 453. A/D Conversion Sequence B Control Register (SEQB_CTRL, address 0x4001 0x00C)bit description

Bit	Symbol	Value	Description	Reset value
19	SYNCBYPASS		Setting this bit allows the hardware trigger input to bypass synchronization flip-flops stages and therefore shorten the time between the trigger input signal and the start of a conversion. There are slightly different criteria for whether or not this bit can be set depending on the clock operating mode: Synchronous mode: Synchronization may be bypassed (this bit may be set) if the selected trigger source is already synchronous with the main system clock (eg. coming from an on-chip, system-clock-based timer). Whether this bit is set or not, a trigger pulse must be maintained for at least one system clock period. Asynchronous mode: Synchronization may be bypassed (this bit may be set) if it is certain that the duration of a trigger input pulse will be at least one cycle of the ADC clock (regardless of whether the trigger comes from and on-chip or off-chip source). If this bit is NOT set, the trigger pulse must at least be maintained for one system clock period.	0
		0	Enable synchronization. The hardware trigger bypass is not enabled.	
		1	Bypass synchronization. The hardware trigger bypass is enabled.	
25	-		Reserved	-
24-20	TSAMP		The default sample period (TSAMP = “00000”) at the beginning of each new conversion is 6.5 ADC clock periods. Depending on a variety of factors including ADC clock rate, output impedance of the analog source driver, ADC resolution, and the selection of channels, the sample time may need to be increased. The value programmed into the TSAMP fields dictates the number of additional ADC clock cycles (beyond 6.5) that the sample period will be extended by. Note: Any additional clocks of sample time inserted will add directly to the overall number of clocks required for a conversion, effectively reducing the overall conversion throughput rate.	0
26	START		Writing a 1 to this field will launch one pass through this conversion sequence. The behavior will be identical to a sequence triggered by a hardware trigger. Do not write a 1 to this bit if the BURST bit is set. **Remark:** This bit is only set to a 1 momentarily when written to launch a conversion sequence. It will consequently always read-back as a zero.	0
27	BURST		Writing a 1 to this bit will cause this conversion sequence to be continuously cycled through. Other B triggers will be ignored while this bit is set. Repeated conversions can be halted by clearing this bit. The sequence currently in progress will be completed before conversions are terminated.	0

Table 453. A/D Conversion Sequence B Control Register (SEQB_CTRL, address 0x4001 0x00C)bit description

Bit	Symbol	Value	Description	Reset value
28	SINGLESTEP		When this bit is set, a hardware trigger or a write to the START bit will launch a single conversion on the next channel in the sequence instead of the default response of launching an entire sequence of conversions. Once all of the channels comprising a sequence have been converted, a subsequent trigger will repeat the sequence beginning with the first enabled channel. Interrupt generation will still occur either after each individual conversion or at the end of the entire sequence, depending on the state of the MODE bit.	0
29	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	N/A
30	MODE		Indicates whether the primary method for retrieving conversion results for this sequence will be accomplished via reading the global data register (SEQB_GDAT) at the end of each conversion, or the individual channel result registers at the end of the entire sequence. Impacts when conversion-complete interrupt/DMA trigger for sequence-B will be generated and which overrun conditions contribute to an overrun interrupt as described below:	0
		0	End of conversion. The sequence B interrupt/DMA flag will be set at the end of each individual A/D conversion performed under sequence B. This flag will mirror the DATAVALID bit in the SEQB_GDAT register. The OVERRUN bit in the SEQB_GDAT register will contribute to generation of an overrun interrupt if enabled.	
		1	End of sequence. The sequence B interrupt/DMA flag will be set when the entire set of sequence B conversions completes. This flag will need to be explicitly cleared by software or by the DMA-clear signal in this mode. The OVERRUN bit in the SEQB_GDAT register will NOT contribute to generation of an overrun interrupt since it is assumed this register will not be utilized in this mode.	
31	SEQB_ENA		Sequence Enable. In order to avoid spuriously triggering the sequence, care should be taken to only set the SEQA_ENA bit when the selected trigger input is in its INACTIVE state (as defined by the TRIGPOL bit). If this condition is not met, the sequence will be triggered immediately upon being enabled.	0
		0	Disabled. Sequence B is disabled. Sequence B triggers are ignored. If this bit is cleared while sequence B is in progress, the sequence will be halted at the end of the current conversion. After the sequence is re-enabled, a new trigger will be required to restart the sequence beginning with the next enabled channel.	
		1	Enabled. Sequence B is enabled.	

27.6.4 A/D Global Data Register A and B

The A/D Global Data Registers contain the result of the most recent A/D conversion completed under each conversion sequence.

Results of A/D conversions can be read in one of two ways. One is to use these A/D Global Data Registers to read data from the ADC at the end of each A/D conversion. Another is to read the individual A/D Channel Data Registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

The global registers are useful in conjunction with DMA operation - particularly when the channels selected for conversion are not sequential (hence the addresses of the individual result registers will not be sequential, making it difficult for the DMA engine to address them). For interrupt-driven code it will more likely be advantageous to wait for an entire sequence to complete and then retrieve the results from the individual channel registers.

Remark: The method to be employed for each sequence should be reflected in the MODE bit in the corresponding ADSEQn_CTRL register since this will impact interrupt and overrun flag generation.

Table 454. A/D Sequence A Global Data Register (SEQA_GDAT, address 0x4001 C010) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	RESULT	This field contains the 12-bit A/D conversion result from the most recent conversion performed under conversion sequence associated with this register. The result is the a binary fraction representing the voltage on the currently-selected input channel as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}. DATAVALID = 1 indicates that this result has not yet been read.	NA
17:16	THCMPRANGE	Indicates whether the result of the last conversion performed was above, below or within the range established by the designated threshold comparison registers (THRn_LOW and THRn_HIGH).	
19:18	THCMPCROSS	Indicates whether the result of the last conversion performed represented a crossing of the threshold level established by the designated LOW threshold comparison register (THRn_LOW) and, if so, in what direction the crossing occurred.	
25:20	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 454. A/D Sequence A Global Data Register (SEQA_GDAT, address 0x4001 C010) bit description

Bit	Symbol	Description	Reset value
29:26	CHN	These bits contain the channel from which the RESULT bits were converted (e.g. 0000 identifies channel 0, 0001 channel 1...).	NA
30	OVERRUN	This bit is set if a new conversion result is loaded into the RESULT field before a previous result has been read - i.e. while the DATAVALID bit is set. This bit is cleared, along with the DATAVALID bit, whenever this register is read. This bit will contribute to an overrun interrupt request if the MODE bit (in SEQA_CTRL) for the corresponding sequence is set to ‘0’ (and if the overrun interrupt is enabled).	0
31	DATAVALID	This bit is set to ‘1’ at the end of each conversion when a new result is loaded into the RESULT field. It is cleared whenever this register is read. This bit will cause a conversion-complete interrupt for the corresponding sequence if the MODE bit (in SEQA_CTRL) for that sequence is set to 0 (and if the interrupt is enabled).	0

Table 455. A/D Sequence B Global Data Register (SEQB_GDAT, address 0x4001 C014) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	RESULT	This field contains the 12-bit A/D conversion result from the most recent conversion performed under conversion sequence associated with this register. This will be a binary fraction representing the voltage on the currently-selected input channel as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}. DATAVALID = 1 indicates that this result has not yet been read.	NA
17:16	THCMPRANGE	Indicates whether the result of the last conversion performed was above, below or within the range established by the designated threshold comparison registers (THRn_LOW and THRn_HIGH). Threshold Range Comparison result. 0x0 = In Range: The last completed conversion was greater than or equal to the value programmed into the designated LOW threshold register (THRn_LOW) but less than or equal to the value programmed into the designated HIGH threshold register (THRn_HIGH). 0x1 = Below Range: The last completed conversion on was less than the value programmed into the designated LOW threshold register (THRn_LOW). 0x2 = Above Range: The last completed conversion was greater than the value programmed into the designated HIGH threshold register (THRn_HIGH). 0x3 = Reserved.	

Table 455. A/D Sequence B Global Data Register (SEQB_GDAT, address 0x4001 C014) bit description

Bit	Symbol	Description	Reset value
19:18	THCMPCROSS	Indicates whether the result of the last conversion performed represented a crossing of the threshold level established by the designated LOW threshold comparison register (THRn_LOW) and, if so, in what direction the crossing occurred. 0x0 = No threshold Crossing detected: The most recent completed conversion on this channel had the same relationship (above or below) to the threshold value established by the designated LOW threshold register (THRn_LOW) as did the previous conversion on this channel. 0x1 = Reserved. 0x2 = Downward Threshold Crossing Detected. Indicates that a threshold crossing in the downward direction has occurred - i.e. the previous sample on this channel was above the threshold value established by the designated LOW threshold register (THRn_LOW) and the current sample is below that threshold. 0x3 = Upward Threshold Crossing Detected. Indicates that a threshold crossing in the upward direction has occurred - i.e. the previous sample on this channel was below the threshold value established by the designated LOW threshold register (THRn_LOW) and the current sample is above that threshold.	
25:20	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
29:26	CHN	These bits contain the channel from which the RESULT bits were converted (e.g. 0b0000 identifies channel 0, 0b0001 channel 1...).	NA
30	OVERRUN	This bit is set if a new conversion result is loaded into the RESULT field before a previous result has been read - i.e. while the DATAVALID bit is set. This bit is cleared, along with the DATAVALID bit, whenever this register is read. This bit will contribute to an overrun interrupt request if the MODE bit (in SEQB_CTRL) for the corresponding sequence is set to 0 (and if the overrun interrupt is enabled).	0
31	DATAVALID	This bit is set to 1 at the end of each conversion when a new result is loaded into the RESULT field. It is cleared whenever this register is read. This bit will cause a conversion-complete interrupt for the corresponding sequence if the MODE bit (in SEQB_CTRL) for that sequence is set to 0 (and if the interrupt is enabled).	0

27.6.5 A/D Channel Data Registers 0 to 11

The A/D Channel Data Registers hold the result of the last conversion completed for each A/D channel. They also include status bits to indicate when a conversion has been completed, when a data overrun has occurred, and where the most recent conversion fits relative to the range dictated by the high and low threshold registers.

Results of A/D conversion can be read in one of two ways. One is to use the A/D Global Data Registers for each of the sequences to read data from the ADC at the end of each A/D conversion. Another is to use these individual A/D Channel Data Registers, typically after the entire sequence has completed. It is recommended to use one method consistently for a given conversion sequence.

Remark: The method to be employed for each sequence should be reflected in the MODE bit in the corresponding SEQ_CTRL register since this will impact interrupt and overrun flag generation.

The information presented in the DAT registers always pertains to the most recent conversion completed on that channel regardless of what sequence requested the conversion or which trigger caused it.

The OVERRUN fields for each channel are also replicated in the FLAGS register.

Table 456. A/D Data Registers (DAT[0:11], address 0x4001 C020 (DAT0) to 0x4001 C04C (DAT11)) bit description

Bit	Symbol	Description	Reset value
3:0	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	RESULT	This field contains the 12-bit A/D conversion result from the last conversion performed on this channel. This will be a binary fraction representing the voltage on the AD0[n] pin, as it falls within the range of V_{REFP} to V_{REFN}. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on V_{REFN}, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on V_{REFP}.	NA
17:16	THCMPRANGE	Threshold Range Comparison result. 0x0 = In Range: The last completed conversion was greater than or equal to the value programmed into the designated LOW threshold register (THRn_LOW) but less than or equal to the value programmed into the designated HIGH threshold register (THRn_HIGH). 0x1 = Below Range: The last completed conversion on was less than the value programmed into the designated LOW threshold register (THRn_LOW). 0x2 = Above Range: The last completed conversion was greater than the value programmed into the designated HIGH threshold register (THRn_HIGH). 0x3 = Reserved.	NA
19:18	THCMPCROSS	Threshold Crossing Comparison result. 0x0 = No threshold Crossing detected: The most recent completed conversion on this channel had the same relationship (above or below) to the threshold value established by the designated LOW threshold register (THRn_LOW) as did the previous conversion on this channel. 0x1 = Reserved. 0x2 = Downward Threshold Crossing Detected. Indicates that a threshold crossing in the downward direction has occurred - i.e. the previous sample on this channel was above the threshold value established by the designated LOW threshold register (THRn_LOW) and the current sample is below that threshold. 0x3 = Upward Threshold Crossing Detected. Indicates that a threshold crossing in the upward direction has occurred - i.e. the previous sample on this channel was below the threshold value established by the designated LOW threshold register (THRn_LOW) and the current sample is above that threshold.	NA
25:20	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 456. A/D Data Registers (DAT[0:11], address 0x4001 C020 (DAT0) to 0x4001 C04C (DAT11)) bit description

Bit	Symbol	Description	Reset value
29:26	CHANNEL	This field is hard-coded to contain the channel number that this particular register relates to (i.e. this field will contain 0b0000 for the DAT0 register, 0b0001 for the DAT1 register, etc)	NA
30	OVERRUN	This bit will be set to a 1 if a new conversion on this channel completes and overwrites the previous contents of the RESULT field before it has been read - i.e. while the DONE bit is set. This bit is cleared, along with the DONE bit, whenever this register is read or when the data related to this channel is read from either of the global SEQn_GDAT registers. This bit (in any of the 12 registers) will cause an overrun interrupt request to be asserted if the overrun interrupt is enabled. **Remark:** While it is allowed to include the same channels in both conversion sequences, doing so may cause erratic behavior of the DONE and OVERRUN bits in the data registers associated with any of the channels that are shared between the two sequences. Any erratic OVERRUN behavior will also affect overrun interrupt generation, if enabled.	NA
31	DATAVALID	This bit is set to 1 when an A/D conversion on this channel completes. This bit is cleared whenever this register is read or when the data related to this channel is read from either of the global SEQn_GDAT registers. **Remark:** While it is allowed to include the same channels in both conversion sequences, doing so may cause erratic behavior of the DONE and OVERRUN bits in the data registers associated with any of the channels that are shared between the two sequences. Any erratic OVERRUN behavior will also affect overrun interrupt generation, if enabled.	NA

27.6.6 A/D Compare Low Threshold Registers 0 and 1

These registers set the LOW threshold levels against which A/D conversions on all channels will be compared.

Each channel will either be compared to the THR0_LOW/HIGH registers or to the THR1_LOW/HIGH registers depending on what is specified for that channel in the CHAN_THRSEL register.

A conversion result LESS THAN this value on any channel will cause the THCMP_RANGE status bits for that channel to be set to 0b01. This result will also generate an interrupt request if enabled to do so via the ADCMPINTEN bits associated with each channel in the INTEN register.

If, for two successive conversions on a given channel, one result is below this threshold and the other is equal-to or above this threshold, than a threshold crossing has occurred. In this case the MSB of the THCMP_CROSS status bits will indicate that a threshold crossing has occurred and the LSB will indicate the direction of the crossing. A threshold crossing event will also generate an interrupt request if enabled to do so via the ADCMPINTEN bits associated with each channel in the INTEN register.

Table 457. A/D Compare Low Threshold register 0 (THR0_LOW, address 0x4001 C050) bit description

Bit	Symbol	Description	Reset value
3:0		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	THRLOW	Low threshold value against which A/D results will be compared	0x000
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 458. A/D Compare Low Threshold register 1 (THR1_LOW, address 0x4001 C054) bit description

Bit	Symbol	Description	Reset value
3:0		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	THRLOW	Low threshold value against which A/D results will be compared	0x000
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

27.6.7 A/D Compare High Threshold Registers 0 and 1

These registers set the HIGH threshold level against which A/D conversions on all channels will be compared.

Each channel will either be compared to the THR0_LOW/HIGH registers or to the THR1_LOW/HIGH registers depending on what is specified for that channel in the CHAN_THRSEL register.

A conversion result greater than this value on any channel will cause the THCMP status bits for that channel to be set to 0b10. This result will also generate an interrupt request if enabled to do so via the ADCMPINTEN bits associated with each channel in the INTEN register.

Table 459. Compare High Threshold register0 (THR0_HIGH, address 0x4001 C058) bit description

Bit	Symbol	Description	Reset value
3:0		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	THRHIGH	High threshold value against which A/D results will be compared	0x000
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 460. Compare High Threshold register 1 (THR1_HIGH, address 0x4001 C05C) bit description

Bit	Symbol	Description	Reset value
3:0		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	THRHIGH	High threshold value against which A/D results will be compared	0x000
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

27.6.8 A/D Channel Threshold Select register

For each channel, this register indicates which pair of threshold registers conversion results should be compared to.

Table 461. A/D Channel Threshold Select register (CHAN_THRSEL, addresses 0x4001 C060) bit description

Bit	Symbol	Value	Description	Reset value
0	CH0_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 0 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 0 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
1	CH1_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 1 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 1 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
2	CH2_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 2 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 2 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
3	CH3_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 3 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 3 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
4	CH4_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 4 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 4 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
5	CH5_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 5 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 5 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	

Table 461. A/D Channel Threshold Select register (CHAN_THRSEL, addresses 0x4001 C060) bit description

Bit	Symbol	Value	Description	Reset value
6	CH6_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 6 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 6 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
7	CH7_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 7 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 7 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
8	CH8_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 8 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 8 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
9	CH9_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 9 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 9 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
10	CH10_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 10 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 10 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
11	CH11_THRSEL		Threshold select by channel.	0
		0	Threshold 0. Channel 11 results will be compared against the threshold levels indicated in the THR0_LOW and THR0_HIGH registers	
		1	Threshold 1. Channel 11 results will be compared against the threshold levels indicated in the THR1_LOW and THR1_HIGH registers	
31:12			Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

27.6.9 A/D Interrupt Enable Register

There are four separate interrupt requests generated by the ADC: conversion-complete or sequence-complete interrupts for each of the two sequences, a threshold-comparison out-of-range interrupt, and a data overrun interrupt. The two conversion/sequence-complete interrupts can also serve as DMA triggers.

These interrupts may be combined into one request on some chips if there is a limited number of interrupt slots. This register contains the interrupt-enable bits for each interrupt.

In this register, threshold events selected in the ADCMPINTENn bits are described as follows:

- Disabled: Threshold comparisons on channel n will not generate an A/D threshold-compare interrupt request.
- Outside threshold: A conversion result on channel n which is outside the range specified by the designated HIGH and LOW threshold registers will set the channel n THCMP flag in the FLAGS register and generate an A/D threshold-compare interrupt request.
- Crossing threshold: Detection of a threshold crossing on channel n will set the channel n THCMP flag in the ADFLAGS register and generate an A/D threshold-compare interrupt request.

Remark: Overrun and threshold-compare interrupts related to a particular channel will occur regardless of which sequence was in progress at the time the conversion was performed or what trigger caused the conversion.

Table 462. A/D Interrupt Enable register (INTEN, address 0x4001 C064) bit description

Bit	Symbol	Value	Description	Reset value
0	SEQA_INTEN		Sequence A interrupt enable.	0
		0	Disabled. The sequence A interrupt/DMA trigger is disabled.	
		1	Enabled. The sequence A interrupt/DMA trigger is enabled and will be asserted either upon completion of each individual conversion performed as part of sequence A, or upon completion of the entire A sequence of conversions, depending on the MODE bit in the SEQA_CTRL register.	
1	SEQB_INTEN		Sequence B interrupt enable.	0
		0	Disabled. The sequence B interrupt/DMA trigger is disabled.	
		1	Enabled. The sequence B interrupt/DMA trigger is enabled and will be asserted either upon completion of each individual conversion performed as part of sequence B, or upon completion of the entire B sequence of conversions, depending on the MODE bit in the SEQB_CTRL register.	
2	OVR_INTEN		Overrun interrupt enable.	0
		0	Disabled. The overrun interrupt is disabled.	
		1	Enabled. The overrun interrupt is enabled. Detection of an overrun condition on any of the 12 channel data registers will cause an overrun interrupt request. In addition, if the MODE bit for a particular sequence is 0, then an overrun in the global data register for that sequence will also cause this interrupt request to be asserted.	
4:3	ADCMPINTEN0		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
6:5	ADCMPINTEN1		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved.	

Table 462. A/D Interrupt Enable register (INTEN, address 0x4001 C064) bit description

Bit	Symbol	Value	Description	Reset value
8:7	ADCMPINTEN2		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
10:9	ADCMPINTEN3		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
12:11	ADCMPINTEN4		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
14:13	ADCMPINTEN5		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
16:15	ADCMPINTEN6		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved.	
18:17	ADCMPINTEN7		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
20:19	ADCMPINTEN8		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
22:21	ADCMPINTEN9		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	

Table 462. A/D Interrupt Enable register (INTEN, address 0x4001 C064) bit description

Bit	Symbol	Value	Description	Reset value
24:23	ADCMPINTEN10		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
26:25	ADCMPINTEN11		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	
		0x2	Crossing threshold.	
		0x3	Reserved	
31:27	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

27.6.10 A/D Flag register

The A/D Flags registers contains the four interrupt request flags along with the individual overrun flags that contribute to an overrun interrupt and the component threshold-comparison flags that contribute to that interrupt.

The channel OVERRUN flags, mirror those in the appearing in the individual ADDAT registers for each channel, indicate a data overrun in each of those registers.

Likewise, the SEQA_OVR and SEQB_OVR bits mirror the OVERRUN bits in the two global data registers (SEQA_GDAT and SEQB_GDAT).

Remark: The SEQn_INT conversion/sequence-complete flags also serve as DMA triggers.

Table 463. A/D Flags register (FLAGS, address 0x4001 C068) bit description

Bit	Symbol	Description	Reset value
0	THCMP0	Threshold comparison event on Channel 0. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
1	THCMP1	Threshold comparison event on Channel 1. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
2	THCMP2	Threshold comparison event on Channel 2. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
3	THCMP3	Threshold comparison event on Channel 3. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
4	THCMP4	Threshold comparison event on Channel 4. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0

Table 463. A/D Flags register (FLAGS, address 0x4001 C068) bit description

Bit	Symbol	Description	Reset value
5	THCMP5	Threshold comparison event on Channel 5. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
6	THCMP6	Threshold comparison event on Channel 6. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
7	THCMP7	Threshold comparison event on Channel 7. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
8	THCMP8	Threshold comparison event on Channel 8. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
9	THCMP9	Threshold comparison event on Channel 9. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
10	THCMP10	Threshold comparison event on Channel 10. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
11	THCMP11	Threshold comparison event on Channel 11. Set to 1 upon either an out-of-range result or a threshold-crossing result if enabled to do so in the INTEN register. This bit is cleared by writing a 1.	0
12	OVERRUN0	Mirrors the OVERRRUN status flag from the result register for A/D channel 0	0
13	OVERRUN1	Mirrors the OVERRRUN status flag from the result register for A/D channel 1	0
14	OVERRUN2	Mirrors the OVERRRUN status flag from the result register for A/D channel 2	0
15	OVERRUN3	Mirrors the OVERRRUN status flag from the result register for A/D channel 3	0
16	OVERRUN4	Mirrors the OVERRRUN status flag from the result register for A/D channel 4	0
17	OVERRUN5	Mirrors the OVERRRUN status flag from the result register for A/D channel 5	0
18	OVERRUN6	Mirrors the OVERRRUN status flag from the result register for A/D channel 6	0
19	OVERRUN7	Mirrors the OVERRRUN status flag from the result register for A/D channel 7	0
20	OVERRUN8	Mirrors the OVERRRUN status flag from the result register for A/D channel 8	0
21	OVERRUN9	Mirrors the OVERRRUN status flag from the result register for A/D channel 9	0
22	OVERRUN10	Mirrors the OVERRRUN status flag from the result register for A/D channel 10	0
23	OVERRUN11	Mirrors the OVERRRUN status flag from the result register for A/D channel 11	0
24	SEQA_OVR	Mirrors the global OVERRUN status flag in the SEQA_GDAT register	0
25	SEQB_OVR	Mirrors the global OVERRUN status flag in the SEQB_GDAT register	0
27:26	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
28	SEQA_INT	Sequence A interrupt/DMA flag. If the MODE bit in the SEQA_CTRL register is 0, this flag will mirror the DATAVALID bit in the sequence A global data register (SEQA_GDAT), which is set at the end of every A/D conversion performed as part of sequence A. It will be cleared automatically when the SEQA_GDAT register is read. If the MODE bit in the SEQA_CTRL register is 1, this flag will be set upon completion of an entire A sequence. In this case it must be cleared by writing a 1 to this SEQA_INT bit. This interrupt must be enabled in the INTEN register.	0

Table 463. A/D Flags register (FLAGS, address 0x4001 C068) bit description

Bit	Symbol	Description	Reset value
29	SEQB_INT	Sequence A interrupt/DMA flag. If the MODE bit in the SEQB_CTRL register is 0, this flag will mirror the DATAVALID bit in the sequence A global data register (SEQB_GDAT), which is set at the end of every A/D conversion performed as part of sequence B. It will be cleared automatically when the SEQB_GDAT register is read. If the MODE bit in the SEQB_CTRL register is 1, this flag will be set upon completion of an entire B sequence. In this case it must be cleared by writing a 1 to this SEQB_INT bit. This interrupt must be enabled in the INTEN register.	0
30	THCMP_INT	Threshold Comparison Interrupt/DMA flag. This bit will be set if any of the 12 THCMP flags in the lower bits of this register are set to 1 (due to an enabled out-of-range or threshold-crossing event on any channel). Each type of threshold comparison interrupt on each channel must be individually enabled in the INTEN register to cause this interrupt. This bit will be cleared when all of the component flags in bits 11:0 are cleared via writing 1s to those bits.	0
31	OVR_INT	Overrun Interrupt flag. Any overrun bit in any of the individual channel data registers will cause this interrupt. In addition, if the MODE bit in either of the SEQn_CTRL registers is 0 then the OVERRUN bit in the corresponding SEQn_GDAT register will also cause this interrupt. This interrupt must be enabled in the INTEN register. This bit will be cleared when all of the individual overrun bits have been cleared via reading the corresponding data registers.	0

27.6.11 A/D trim register

The A/D trim register configures the ADC for the appropriate operating range of the analog supply voltage VDDA.

Remark: Failure to set the VRANGE bit correctly causes the ADC to return incorrect conversion results.

Table 464. A/D trim register (TRM, addresses 0x4001 C06C) bit description

Bit	Symbol	Value	Description	Reset value
4:0	-		Reserved.	-
5	VRANGE		Reserved.	0
		0	High voltage. VDD = 2.7 V to 3.6 V.	
		1	Low voltage. VDD = 2.4 V to 2.7 V.	
31:6	-		Reserved.	-

27.7 Functional description

27.7.1 Conversion Sequences

A conversion sequence is a single pass through a series of A/D conversions performed on a selected set of A/D channels. Software can set-up two independent conversion sequences, either of which can be triggered by software or by a transition on one of the hardware triggers. Each sequence can be triggered by a different hardware trigger. One of these conversion sequences is referred to as the A sequence and the other as the B sequence. It is not necessary to employ both sequences.

An optional single-step mode allows advancing through the channels of a sequence one at a time on each successive occurrence of a trigger.

The user can select whether a trigger on the B sequence can interrupt an already-in-progress A sequence. The B sequence, however, can never be interrupted by an A trigger.

27.7.2 Hardware-triggered conversion

Software can select which of these hardware triggers will launch each conversion sequence and it can specify the active edge for the selected trigger independently for each conversion sequence.

For each conversion sequence, if a designated trigger event occurs, one single cycle through that conversion sequence will be launched unless:

- The BURST bit in the ADSEQn_CTRL register for this sequence is set to 1.
- The requested conversion sequence is already in progress.
- A set of conversions for the alternate conversion sequence is already in progress except in the case of a B trigger interrupting an A sequence if the A sequence is set to LOWPRIO.

If any of these conditions is true, the new trigger event will be ignored and will have no effect.

In addition, if the single-step bit for a sequence is set, each new trigger will cause a single conversion to be performed on the next channel in the sequence rather than launching a pass through the entire sequence.

If the A sequence is enabled to be interrupted (i.e. the LOWPRIO bit in the SEQA_CTRL register is set) and a B trigger occurs while an A sequence is in progress, then the following will occur:

- The A/D conversion which is currently in-progress will be aborted.
- The A sequence will be paused, and the B sequence will immediately commence.
- The interrupted A sequence will resume after the B sequence completes, beginning with the conversion that was aborted when the interruption occurred. The channel for that conversion will be re-sampled.

27.7.2.1 Avoiding spurious hardware triggers

Care should be taken to avoid generating a spurious trigger when writing to the SEQn_CTRL register to change the trigger selected for the sequence, switch the polarity of the selected trigger, or to enable the sequence for operation.

In general, the TRIGGER and TRIGPOL bits in the SEQn_CTRL register should only be written to when the sequence is disabled (while the SEQn_ENA bit is LOW). The SEQn_ENA bit itself should only be set when the selected trigger input is in its INACTIVE state (as designated by the TRIGPOL bit). If this condition is not met, a trigger will be generated immediately upon enabling the sequence - even though no actual transition has occurred on the trigger input.

27.7.3 Software-triggered conversion

There are two ways that software can trigger a conversion sequence:

1. Start Bit: The first way to software-trigger an sequence is by setting the START bit in the corresponding SEQn_CTRL register. The response to this is identical to occurrence of a hardware trigger on that sequence. Specifically, one cycle of conversions through that conversion sequence will be immediately triggered except as indicated above.
2. Burst Mode: The other way to initiate conversions is to set the BURST bit in the SEQn_CTRL register. As long as this bit is 1 the designated conversion sequence will be continuously and repetitively cycled through. Any new software or hardware trigger on this sequence will be ignored.

If a bursting A sequence is allowed to be interrupted (i.e. the LOWPRIO bit in its SEQA_CTRL register is set to 1 and a software or hardware trigger for the B sequence occurs, then the burst will be immediately interrupted and a B sequence will be initiated. The interrupted A sequence will resume continuous cycling, starting with the aborted conversion, after the alternate sequence has completed.

27.7.4 Interrupts

There are four interrupts that can be generated by the ADC:

- Conversion-Complete or Sequence-Complete interrupts for sequences A and B
- Threshold-Compare Out-of-Range Interrupt
- Data Overrun Interrupt

Any of these interrupt requests may be individually enabled or disabled in the INTEN register.

27.7.4.1 Conversion-Complete or Sequence-Complete interrupts

For each of the two sequences, an interrupt request can either be asserted at the end of each A/D conversion performed as part of that sequence or when the entire sequence of conversions is completed. The MODE bits in the SEQn_CTRL registers select between these alternative behaviors.

If the MODE bit for a sequence is 0 (conversion-complete mode) then the interrupt flag for that sequence will reflect the state of the DATAVALID bit in the global data register (SEQn_GDAT) for that sequence. In this case, reading the SEQn_GDAT register will automatically clear the interrupt request.

If the MODE bit for the sequence is 1 (sequence-complete mode) then the interrupt flag must be written-to by software to clear it (except when used as a DMA trigger, in which case it will be cleared in hardware by the DMA engine).

27.7.4.2 Threshold-Compare Out-of-Range Interrupt

Every conversion performed on any channel is automatically compared against a designated set of low and high threshold levels specified in the THRn_HIGH and THRn_LOW registers. The results of this comparison on any individual channel(s) can be enabled to cause a threshold-compare interrupt if that result was above or below the range specified by the two thresholds or, alternatively, if the result represented a crossing of the low threshold in either direction.

This flag must be cleared by a software write to clear the individual THCMP flags in the FLAGS register.

27.7.4.3 Data Overrun Interrupt

This interrupt request will be asserted if any of the OVERRUN bits in the individual channel data registers are set. In addition, the OVERRUN bits in the two sequence global data (SEQn_GDAT) registers will cause this interrupt request IF the MODE bit for that sequence is set to 0 (conversion-complete mode).

This flag will be cleared when the OVERRUN bit that caused it is cleared via reading the register containing it.

Note that the OVERRUN bits in the individual data registers are cleared when data related to that channel is read from either of the global data registers as well as when the individual data registers themselves are read.

27.7.5 Optional operating modes

The following optional mode of A/D operation may be selected in the CTRL register:

Low-power mode. When this mode is selected, the analog portions of the ADC are automatically shut down when no conversions are in progress. The ADC is automatically restarted whenever any hardware or software trigger event occurs. This mode can save an appreciable amount of power when the ADC is not in continuous use, but at the expense of a delay between the trigger event and the onset of sampling and conversion.

27.7.6 DMA control

The sequence-A or sequence-B conversion/sequence-complete interrupts may also be used to generate a DMA trigger. To trigger a DMA transfer, the same conditions must be met as the conditions for generating an interrupt (see Section 27.7.4 and Section 27.6.9).

Remark: If the DMA is used, the ADC interrupt must be disabled in the NVIC.

For DMA transfers, only burst requests are supported. The burst size can be set to one in the DMA channel control register (see Table 262). If the number of ADC channels is not equal to one of the other DMA-supported burst sizes (applicable DMA burst sizes are 1, 4, 8), set the burst size to one.

The DMA transfer size determines when a DMA interrupt is generated. The transfer size can be set to the number of ADC channels being converted. Non-contiguous channels can be transferred by the DMA using the scatter/gather linked lists.

27.7.7 Hardware Trigger Source Selection

Each ADC has a selection of several on-chip and off-chip hardware trigger sources. The trigger to be used for each conversion sequence is specified in the TRIGGER fields in the two SEQn_CTRL registers. See Section 7.4 “ADC trigger inputs” for real hardware trigger source selection.

UM11607

Chapter 28: Analog comparator

Rev. 3 — April 2023 User manual

28.1 How to read this chapter

The analog comparator is available on all LPC86x parts.

28.2 Features

- Selectable external inputs can be used as either the positive or negative input of the comparator.
- The Internal voltage reference (0.9 V bandgap reference) can be used as either the positive or negative input of the comparator.
- 32-stage voltage ladder can be used as either the positive or negative input of the comparator.
- Voltage ladder source selectable between the supply pin V_{DD} or VDDCMP pin.
- Voltage ladder can be separately powered down when not required.
- Interrupt capability

28.3 Basic configuration

Configure the analog comparator using the following registers:

- In the SYSAHBCLKCTRL register, set bit 19 (Table 94) to enable the clock to the register interface.
- You can enable or disable the power to the analog comparator through the PDRUNCFG register (Table 118).
- Clear the analog comparator peripheral reset using the PRESETCTRL register (Table 95).
- The analog comparator interrupt is connected to interrupt #11 in the NVIC.
- Configure the analog comparator pin functions through the switch matrix. See Section 28.4.

28.3.1 Connect the comparator output to the ADC

The comparator output function (ACMP_O) can be used to start the ADC conversion, more generally, to create an ADC conversion event without assigning a pin through the switch matrix. To create an ADC event internally connected to the comparator output, select the comparator output as one of the ADC trigger inputs through the ADC trigger select register (see Table 451 “A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description” and Table 452 “A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description”.

28.4 Pin description

The analog comparator reference voltage, the inputs, and the output are assigned to external pins through the switch matrix. You can assign the analog comparator output to any pin on the package that is not a supply or ground pin. The comparator inputs and the reference voltage are fixed-pin functions that must be enabled through the switch matrix and can only be assigned to special pins on the package.

See Section 11.3.1 “Connect an internal signal to a package pin” to assign the analog comparator output to any pin on the LPC86x package.

Table 465. Analog comparator pin description

Function	Type	Pin	Description	SWM register	Reference
ACMP_I1	I	PIO0_0	Comparator input 1	PINENABLE0	Table 139
ACMP_I2	I	PIO0_1	Comparator input 2	PINENABLE0	Table 139
ACMP_I3	I	PIO0_14	Comparator input 3	PINENABLE0	Table 139
ACMP_I4	I	PIO0_23	Comparator input 4	PINENABLE0	Table 139
ACMP_I5	I	PIO0_30	Comparator input 5	PINENABLE0	Table 139
ACMP_O	O	any	Comparator output	PINASSIGN11	Table 136
VDDCMP	I	PIO0_6	External reference voltage source for 32-stage Voltage Ladder.	PINENABLE0	Table 139

28.5 General description

The analog comparator can compare voltage levels on external pins and internal voltages.

The comparator has seven inputs multiplexed separately to its positive and negative inputs. The multiplexers are controlled by the comparator register CTL (see Figure 165 and Table 467).

Input 0 of the multiplexer is the programmable voltage ladder output.

Inputs 1 to 5 connect the external inputs ACMP_**I**[5:1].

Input 6 of the multiplexer connects the internal reference voltage input.

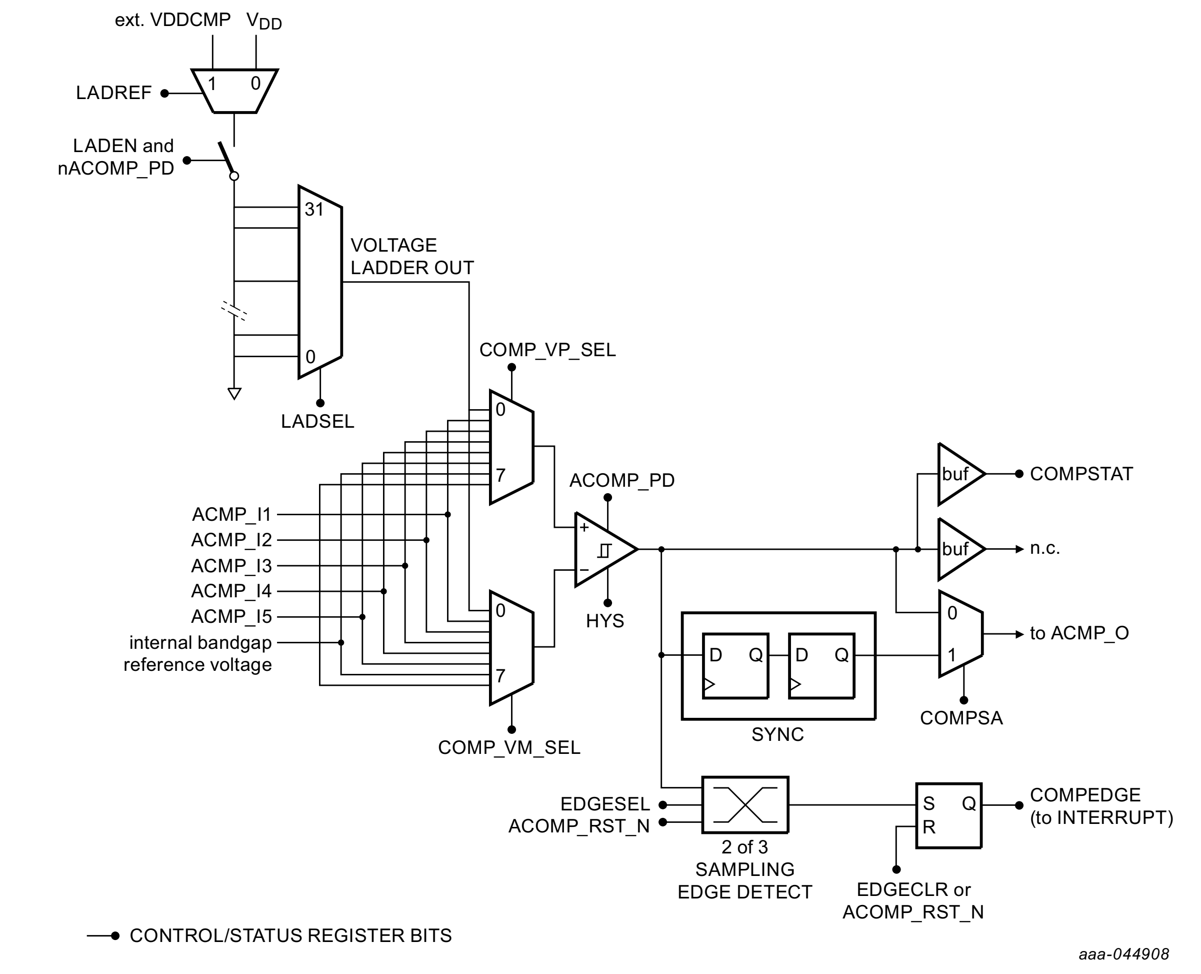


Fig 165. Comparator block diagram

28.5.1 Reference voltages

The voltage ladder can use two reference voltages, from the VDDCMP or the VDD pin. The voltage ladder selects one of 32 steps between the pin voltage and VSS inclusive. The voltage on VDDCMP should not exceed that on VDD.

28.5.2 Settling times

After the voltage ladder is powered on, it requires stabilization time until comparisons using it are accurate. Much shorter settling times apply after the LADSEL value is changed and when either or both voltage sources are changed. Software can deal with these factors by repeatedly reading the comparator output until a number of readings yield the same result.

28.5.3 Interrupts

The interrupt output comes from edge detection circuitry in this module. Rising edges, falling edges, or both edges can set the COMPEDGE bit and thus request an interrupt. COMPEDGE and the interrupt request are cleared when software writes a 1 to EDGECLR.

28.5.4 Comparator outputs

The comparator output (conditioned by COMPSA bit) can be routed to an external pin. When COMPSA is 0 and the comparator interrupt is disabled, the comparator can be used with the bus clock disabled (Table 94 “System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description”) to save power if the control registers don’t need to be written.

The status of the comparator output can be observed through the comparator status register bit.

28.6 Register description

Table 466. Register overview: Analog comparator (base address 0x4002 4000)

Name	Access	Address offset	Description	Reset value	Reference
CTRL	R/W	0x000	Comparator control register	0	Table 467
LAD	R/W	0x004	Voltage ladder register	0	Table 468

28.6.1 Comparator control register

This register enables the comparator, configures the interrupts, and controls the input multiplexers on both sides of the comparator. All bits not shown in Table 467 are reserved and should be written as 0.

Table 467. Comparator control register (CTRL, address 0x4002 4000) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved. Write as 0.	0
4:3	EDGESEL		This field controls which edges on the comparator output set the COMPEDGE bit (bit 23 below):	0
		0x0	Falling edges	
		0x1	Rising edges	
		0x2	Both edges	
		0x3	Both edges	
5	-		Reserved. Write as 0.	0
6	COMPSA		Comparator output control	0
		0	Comparator output is used directly.	
		1	Comparator output is synchronized to the bus clock for output to other modules.	
7	-		Reserved. Write as 0.	0

Table 467. Comparator control register (CTRL, address 0x4002 4000) bit description

Bit	Symbol	Value	Description	Reset value
10:8	COMP_VP_SEL		Selects positive voltage input	0
		0x0	Voltage ladder output	
		0x1	ACMP_I1	
		0x2	ACMP_I2	
		0x3	ACMP_I3	
		0x4	ACMP_I4	
		0x5	ACMP_I5	
		0x6	Band gap. Internal reference voltage.	
		0x7	Reserved.	
13:11	COMP_VM_SEL		Selects negative voltage input	0
		0x0	Voltage ladder output	
		0x1	ACMP_I1	
		0x2	ACMP_I2	
		0x3	ACMP_I3	
		0x4	ACMP_I4	
		0x5	ACMP_I5	
		0x6	Band gap. Internal reference voltage.	
		0x7	Reserved.	
19:14	-		Reserved. Write as 0.	0
20	EDGECLR		Interrupt clear bit. To clear the COMPEDGE bit and thus negate the interrupt request, toggle the EDGECLR bit by first writing a 1 and then a 0.	0
21	COMPSTAT		Comparator status. This bit reflects the state of the comparator output.	0
22	-		Reserved. Write as 0.	0
23	COMPEDGE		Comparator edge-detect status.	0
24	INTENA		Must be set to generate interrupts.	1
26:25	HYS		Controls the hysteresis of the comparator. When the comparator is outputting a certain state, this is the difference between the selected signals, in the opposite direction from the state being output, that will switch the output.	0
		0x0	None (the output will switch as the voltages cross)	
		0x1	5 mV	
		0x2	10 mV	
		0x3	20 mV	
31:27	-		Reserved	-

28.6.2 Voltage ladder register

This register enables and controls the voltage ladder. The fraction of the reference voltage produced by the ladder is programmable in steps of 1/31.

Table 468. Voltage ladder register (LAD, address 0x4002 4004) bit description

Bit	Symbol	Value	Description	Reset value
0	LADEN		Voltage ladder enable	0
5:1	LADSEL		Voltage ladder value. The reference voltage Vref depends on the LADREF bit below. 00000 = V_{SS} 00001 = 1 x Vref/31 00010 = 2 x Vref/31 ... 11111 = Vref	0
6	LADREF		Selects the reference voltage Vref for the voltage ladder:	0
		0	Supply pin VDD	
		1	VDDCMP pin	
31:7	-		Reserved.	0

UM11607

Chapter 29: CRC engine

Rev. 3 — April 2023 **User manual**

29.1 How to read this chapter

The CRC engine is available on all LPC86x parts.

29.2 Features

- Supports three common polynomials CRC-CCITT, CRC-16, and CRC-32.
  - CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$
  - CRC-16: $x^{16} + x^{15} + x^2 + 1$
  - CRC-32: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
- Bit order reverse and 1’s complement programmable setting for input data and CRC sum.
- Programmable seed number setting.
- Accept any size of data width per write: 8, 16 or 32-bit.
  - 8-bit write: 1-cycle operation
  - 16-bit write: 2-cycle operation (8-bit x 2-cycle)
  - 32-bit write: 4-cycle operation (8-bit x 4-cycle)

29.3 Basic configuration

Enable the clock to the CRC engine in the SYSAHBCLKCTRL register (Table 94, bit 13).

29.4 Pin description

The CRC engine has no configurable pins.

29.5 General description

The Cyclic Redundancy Check (CRC) generator with programmable polynomial settings supports several CRC standards commonly used.

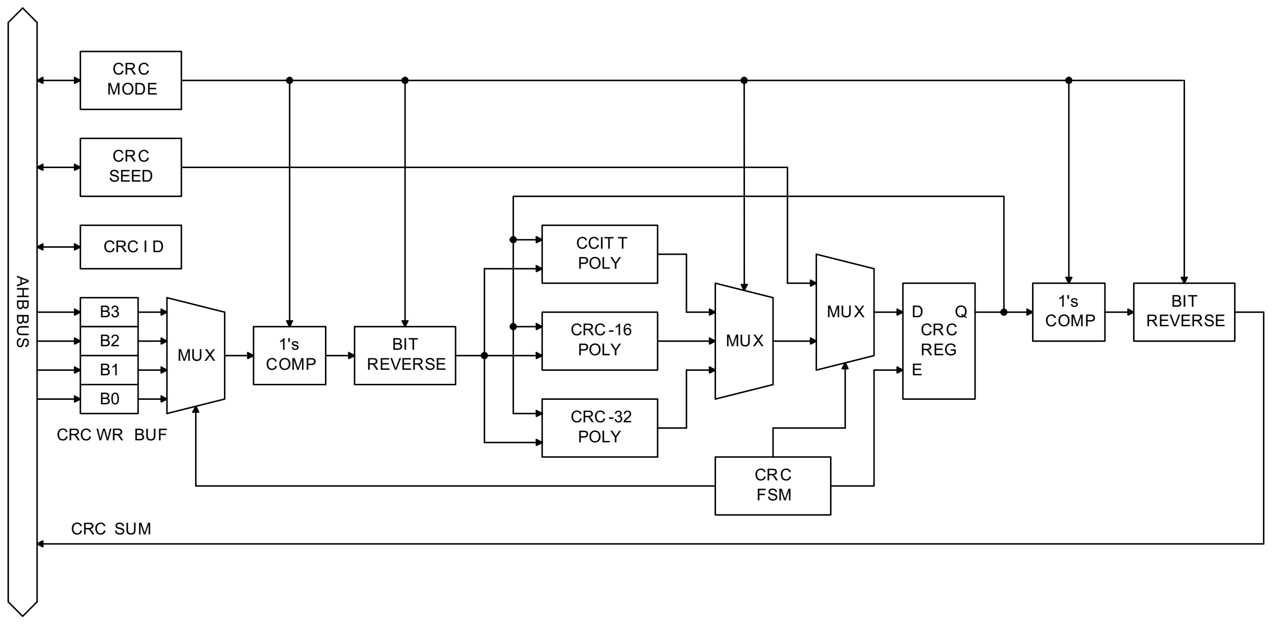


Fig 166. CRC block diagram

29.6 Register description

Table 469. Register overview: CRC engine (base address 0x5000 4000)

Name	Access	Address offset	Description	Reset value	Reference
MODE	R/W	0x000	CRC mode register	0x0000 0000	Table 470
SEED	R/W	0x004	CRC seed register	0x0000 FFFF	Table 471
SUM	RO	0x008	CRC checksum register	0x0000 FFFF	Table 472
WR_DATA	WO	0x00C	CRC data register	-	Table 473

29.6.1 CRC mode register

Table 470. CRC mode register (MODE, address 0x5000 4000) bit description

Bit	Symbol	Description	Reset value
1:0	CRC_POLY	CRC polynom: 1X= CRC-32 polynomial 01= CRC-16 polynomial 00= CRC-CCITT polynomial	00
2	BIT_RVS_WR	Data bit order: 1= Bit order reverse for CRC_WR_DATA (per byte) 0= No bit order reverse for CRC_WR_DATA (per byte)	0
3	CMPL_WR	Data complement: 1= 1’s complement for CRC_WR_DATA 0= No 1’s complement for CRC_WR_DATA	0
4	BIT_RVS_SUM	CRC sum bit order: 1= Bit order reverse for CRC_SUM 0= No bit order reverse for CRC_SUM	0
5	CMPL_SUM	CRC sum complement: 1= 1’s complement for CRC_SUM 0=No 1’s complement for CRC_SUM	0
31:6	Reserved	Always 0 when read	0x0000000

29.6.2 CRC seed register

Table 471. CRC seed register (SEED, address 0x5000 0004) bit description

Bit	Symbol	Description	Reset value
31:0	CRC_SEED	A write access to this register will load CRC seed value to CRC_SUM register with selected bit order and 1's complement pre-processes. **Remark:** A write access to this register will overrule the CRC calculation in progresses.	0x0000 FFFF

29.6.3 CRC checksum register

This register is a Read-only register containing the most recent checksum. The read request to this register is automatically delayed by a finite number of wait states until the results are valid and the checksum computation is complete.

Table 472. CRC checksum register (SUM, address 0x5000 0008) bit description

Bit	Symbol	Description	Reset value
31:0	CRC_SUM	The most recent CRC sum can be read through this register with selected bit order and 1's complement post-processes.	0x0000 FFFF

29.6.4 CRC data register

This register is a Write-only register containing the data block for which the CRC sum will be calculated.

Table 473. CRC data register (WR_DATA, address 0x5000 000C) bit description

Bit	Symbol	Description	Reset value
31:0	CRC_WR_DATA	Data written to this register will be taken to perform CRC calculation with selected bit order and 1's complement pre-process. Any write size 8, 16 or 32-bit are allowed and accept back-to-back transactions.	-

29.7 Functional description

The following sections describe the register settings for each supported CRC standard:

29.7.1 CRC-CCITT set-up

Polynomial = $x^{16} + x^{12} + x^5 + 1$

Seed Value = 0xFFFF

Bit order reverse for data input: NO

1's complement for data input: NO

Bit order reverse for CRC sum: NO

1's complement for CRC sum: NO

CRC_MODE = 0x0000 0000

CRC_SEED = 0x0000 FFFF

29.7.2 CRC-16 set-up

Polynomial = $x^{16} + x^{15} + x^2 + 1$

Seed Value = 0x0000

Bit order reverse for data input: YES

1's complement for data input: NO

Bit order reverse for CRC sum: YES

1's complement for CRC sum: NO

CRC_MODE = 0x0000 0015

CRC_SEED = 0x0000 0000

29.7.3 CRC-32 set-up

Polynomial = $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Seed Value = 0xFFFF FFFF

Bit order reverse for data input: YES

1's complement for data input: NO

Bit order reverse for CRC sum: YES

1's complement for CRC sum: YES

CRC_MODE = 0x0000 0036

CRC_SEED = 0xFFFF FFFF

UM11607

Chapter 30: Serial Wire Debug (SWD)

Rev. 3 — April 2023 **User manual**

30.1 How to read this chapter

The debug functionality is identical for all LPC86x parts.

30.2 Features

- Supports ARM Serial Wire Debug mode.
- Direct debug access to all memories, registers, and peripherals.
- No target resources are required for the debugging session.
- Four breakpoints.
- Two data watchpoints that can also be used as triggers.
- Supports JTAG boundary scan.

30.3 General description

Debug functions are integrated into the ARM Cortex-M0+. Serial wire debug functions are supported. The ARM Cortex-M0+ is configured to support up to four breakpoints and two watchpoints.

JTAG only support for boundary scan.

30.4 Pin description

The SWD functions are assigned to pins through the switch matrix. The SWD functions are fixed-pin functions that are enabled through the switch matrix and can only be assigned to special pins on the package. The SWD functions are enabled by default.

See Section 11.3.2 to enable the analog comparator inputs and the reference voltage input.

Table 474. SWD pin description

Function	Type	Pin	Description	SWM register	Reference
SWCLK	I/O	SWCLK/PIO0_3/ TCK	Serial Wire **Clock.** This pin is the clock for SWD debug logic when in the Serial Wire Debug mode (SWD). This pin is pulled up internally.	PINENABLE0	Table 139
SWDIO	I/O	SWDIO/PIO0_2/ TMS	**Serial wire debug data input/output.** The SWDIO pin is used by an external debug tool to communicate with and control the LPC86x. This pin is pulled up internally.	PINENABLE0	Table 139

The boundary scan mode and the pins needed are selected by hardware (see Section 30.5.3). There is no access to the boundary scan pins through the switch matrix.

Table 475. JTAG boundary scan pin description

Function	Pin name	Type	Description
TCK	SWCLK/PIO0_3/ TCK	I	**JTAG Test Clock.** This pin is the clock for JTAG boundary scan when the $\overline{\text{RESET}}$ pin is LOW.
TMS	SWDIO/PIO0_2/ TMS	I	JTAG **Test Mode Select.** The TMS pin selects the next state in the TAP state machine. This pin includes an internal pull-up and is used for JTAG boundary scan when the $\overline{\text{RESET}}$ pin is LOW.
TDI	PIO0_1/ACMP_I2/ CLKIN/TDI	I	JTAG **Test Data In.** This is the serial data input for the shift register. This pin includes an internal pull-up and is used for JTAG boundary scan when the $\overline{\text{RESET}}$ pin is LOW.
TDO	PIO0_0/ACMP_I1/ TDO	O	JTAG **Test Data Output.** This is the serial data output from the shift register. Data is shifted out of the device on the negative edge of the TCK signal. This pin is used for JTAG boundary scan when the $\overline{\text{RESET}}$ pin is LOW.
TRST	PIO0_4/ WAKEUP/$\overline{\text{TRST}}$/ ADC_11	I	JTAG **Test Reset.** The TRST pin can be used to reset the test logic within the debug logic. This pin includes an internal pull-up and is used for JTAG boundary scan when the $\overline{\text{RESET}}$ pin is LOW.

30.5 Functional description

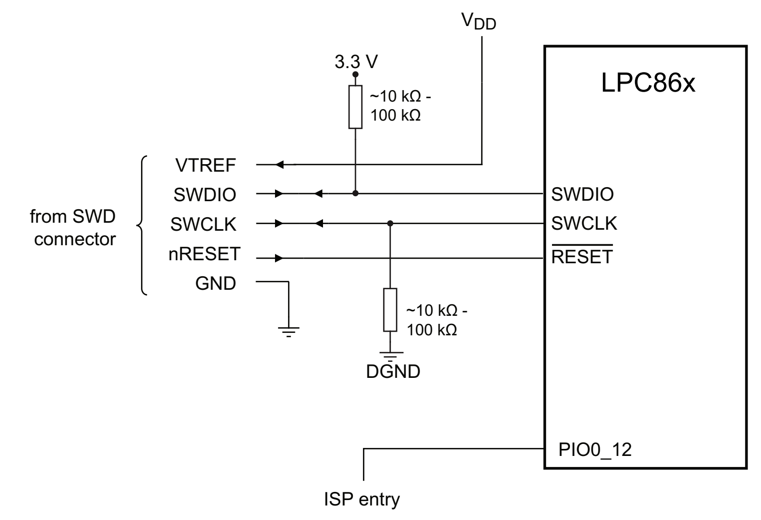
30.5.1 Debug limitations

It is recommended not to use the debug mode during Deep-sleep or Power-down mode.

During a debugging session, the System Tick Timer is automatically stopped whenever the CPU is stopped. Other peripherals are not affected.

30.5.2 Debug connections for SWD

For debugging purposes, it is useful to provide access to the ISP entry pin PIO0_12. This pin can be used to recover the part from configurations which would disable the SWD port such as improper PLL configuration, re-configuration of SWD pins, entry into Deep power-down mode out of reset, etc. This pin can be used for other functions such as GPIO, but it should not be held LOW on power-up or reset.



The VTREF pin on the SWD connector enables the debug connector to match the target voltage.

Fig 167. Connecting the SWD pins to a standard SWD connector

30.5.3 Boundary scan

The $\overline{\text{RESET}}$ pin selects between the JTAG boundary scan ($\overline{\text{RESET}}$ = LOW) and the ARM SWD debug ($\overline{\text{RESET}}$ = HIGH). The ARM SWD debug port is disabled while the part is in reset.

To perform boundary scan testing, follow these steps:

1. Erase any user code residing in flash.
2. Power up the part with the $\overline{\text{RESET}}$ pin pulled HIGH externally.
3. Wait for at least 250 μs.
4. Pull the $\overline{\text{RESET}}$ pin LOW externally.
5. Perform boundary scan operations.
6. Once the boundary scan operations are completed, assert the TRST pin to enable the SWD debug mode and release the $\overline{\text{RESET}}$ pin (pull HIGH).

Remark: The JTAG interface cannot be used for debug purposes.

Remark: POR, BOD reset, or a LOW on the TRST pin puts the test TAP controller in the Test-Logic Reset state. The first TCK clock while $\overline{\text{RESET}}$ = HIGH places the test TAP in Run-Test Idle mode.

UM11607
Chapter 31: Supplementary information
Rev. 3 — April 2023 User manual

31.1 Abbreviations

Table 476. Abbreviations

Acronym	Description
A/D	Analog-to-Digital
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
APB	Advanced Peripheral Bus
BOD	BrownOut Detection
FTM	FlexTimer
GPIO	General Purpose Input/Output
I3C	Improved Inter Integrated Circuit
JTAG	Joint Test Action Group
PLL	Phase-Locked Loop
RC	Resistor-Capacitor
SPI	Serial Peripheral Interface
SSI	Serial Synchronous Interface
SSP	Synchronous Serial Port
SWD	Serial Wire Debug
TAP	Test Access Port
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous Asynchronous Receiver/Transmitter

31.2 References

[1] **LPC86x —** LPC86x Data sheet

[2] **ES_LPC86x —** LPC86x Errata sheet

[3] **DDI0484B_cortex_m0p_r0p0_trm —** ARM Cortex-M0+ Technical Reference Manual

[4] **DDI0486A —** ARM technical reference manual

[5] ARMv6-M Architecture Reference Manual

31.3 Tables

31.4 Figures

31.5 Contents

Chapter 18: USART0/1/2

Chapter 19: SPI0/1

Chapter 20: I2C0

Chapter 21: I3C bus interface

Date of release: April 2023

Document identifier: UM11607

Legal information

Trademarks

Date of release: April 2023

Document identifier: UM11607