UM11029
LPC84x User manual

Rev. 1.6 — 8 December 2017

User manual

Revision history

Rev	Date	Description
1.6	20171208	LPC84x User manual
Modifications:		• Updated Table 371 "Timer/Counter pin description" :Changed function names.
		• Updated Section 8.3.2 "Set up the PLL". Added a line item: Power-down the system PLL in the PDRUNCFG register.
1.5	20171113	LPC84x User manual
Modifications:		• Updated Figure 66 "ROM pointer structure".
		• Updated reset value of the SEL bit of CLKOUTSEL register. See Table 158 "CLKOUT clock source select register (CLKOUTSEL, address 0x4004 80F0) bit description".
		• Added bit 8, ASYNCMODE to the CTRL register. See Table 454 "A/D Control Register (CTRL, addresses 0x4001 C000) bit description".
		• Updated Figure 3 "Boot ROM structure".
		• Updated Figure 61 "ADC clocking".
		• Updated Remark in Section 15.7.6.3 "Wake-up from power-down mode".
		• Updated Figure 12 "Example: Connect function U0_RXD and U0_TXD to pins 4 and 14".
1.4	20170914	LPC84x User manual
Modifications:		• Updated Figure 2 "LPC84x Memory mapping".
1.3	20170811	LPC84x User manual
Modifications:		• Updated Table 1 "Ordering information": Description of part number LPC844M201JHI48 and package name, HVQFN48.
1.2	20170801	LPC84x User manual
Modifications:		• Updated Table 2 "Ordering options". LPC845M301JHI33 does not have Capacitive Touch.
		• Updated Chapter 26 "LPC84x Capacitive Touch", Section 26.1 "How to read this chapter": The Capacitive Touch is available on LPC845M301JBD64, LPC845M301JBD48, and LPC845M301JHI48 devices only.
		• Updated the text in: Section 12.1 "How to read this chapter".
1.1	20170703	LPC84x User manual
Modifications:		• Added Capacitive Touch Interface Chapter.
		• Updated Section 5.8.6 "SH_CMD_WRITE_BLOCK (0xA6) command"If a block number crosses a sector boundary, the LPC84x automatically erases the sector prior to the write operation.
		• Updated Section 5.7.3 "I2C ISP mode transaction protocol": The LPC84x will respond to a host system on I2C addresses 0x18, 0x1C, 0x30, and 0x38. The host system's I2C master clock rate can be as high as 400 kHz. Was: 1 MHz.
1.0	20170616	Initial revision. LPC84x User manual.

Contact information

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

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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **2 of 530**

UM11029

Chapter 1: LPC84x Introductory information

Rev. 1.6 — 8 December 2017   **User manual**

1.1 Introduction

The LPC84x are an ARM Cortex-M0+ based, low-cost 32-bit MCU family operating at CPU frequencies of up to 30 MHz. The LPC84x support up to 64 KB of flash memory and 16 KB of SRAM.

The peripheral complement of the LPC84x includes a CRC engine, four I^2C-bus interfaces, up to five USARTs, up to two SPI interfaces, Capacitive Touch Interface (Cap Touch), one multi-rate timer, self-wake-up timer, SCTimer/PWM, one general purpose 32-bit counter/timer, a DMA, one 12-bit ADC, two 10-bit DACs, 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 33.2 "References".

1.2 Features

- System:
  - ARM Cortex-M0+ processor (revision r0p1), running at frequencies of up to 30 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.
  - Micro Trace Buffer (MTB)
- Memory:
  - Up to 64 KB on-chip flash programming memory with 64 Byte page write and erase.
  - Fast Initialization Memory (FAIM) allowing the user to configure chip behavior on power-up.
  - Code Read Protection (CRP).
  - Up to 16 KB SRAM consisting of two 8 KB contiguous SRAM banks. One 8 KB of SRAM can be used for MTB.
  - 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, SPI, and I^2C.
  - FAIM API.
  - FRO API.

- Flash In-Application Programming (IAP) and In-System Programming (ISP).
- On-chip ROM APIs for integer divide.

- Digital peripherals:

  - High-speed GPIO interface connected to the ARM Cortex-M0+ IO bus with up to 32 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.
  - 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 25 channels and 13 trigger inputs.
  - Capacitive Touch Interface.

- Timers:

  - One SCTimer/PWM with five input and seven output functions (including capture and match) for timing and PWM applications. Inputs and outputs can be routed to or from external pins and internally to or from selected peripherals. Internally, the SCTimer/PWM supports 8 match/captures, 8 events, and 8 states.
  - One 32-bit general purpose counter/timer, with four match outputs and three capture inputs. Supports PWM mode, external count, and DMA.
  - 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 1.2 Msamples/s. The ADC supports two independent conversion sequences.
  - Comparator with five input pins and external or internal reference voltage.
  - Two 10-bit DACs.

- Serial peripherals:

  - Five USART interfaces with pin functions assigned through the switch matrix and two fractional baud rate generators.
  - Two SPI controllers with pin functions assigned through the switch matrix.
  - Four I^{2}C-bus interfaces. One I^{2}C supports Fast-mode Plus with 1 Mbit/s data rates on two true open-drain pins and listen mode. Three I^{2}Cs support data rates up to 400 kbit/s on standard digital pins.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **4 of 530**

- Clock generation:
  - Free Running Oscillator (FRO). This oscillator provides a selectable 18 MHz, 24 MHz, and 30 MHz outputs that can be used as a system clock. Also, these outputs can be divided down to 1.125 MHz, 1.5 MHz, 1.875 MHz, 9 MHz, 12 MHz, and15 MHz for system clock. The FRO is trimmed to ±1 % accuracy over the entire voltage and temperature range 0 C to 70 C.
  - Low power boot at 1.5 MHz using FAIM memory.
  - External clock input for clock frequencies of up to 25 MHz.
  - Crystal oscillator with an operating range of 1 MHz to 25 MHz.
  - Low power oscillator can be used as a clock source to the watchdog timer.
  - Programmable watchdog oscillator with a frequency range of 9.4 kHz to 2.3 MHz.
  - 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:
  - Integrated PMU (Power Management Unit) to minimize power consumption.
  - 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, and I2C 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, LQFP48, HVQFN48, and HVQFN33 packages

1.3 Ordering options

Table 1.   Ordering information

Type number	Package			Version
	Name	**Description**		
LPC845M301JBD64	LQFP64	Plastic low profile quad flat package; 64 leads; body 10× 10 × 1.4 mm		SOT314-2
LPC845M301JBD48	LQFP48	Plastic low profile quad flat package; 48 leads; body 7× 7 × 1.4 mm		SOT313-2
LPC845M301JHI48	HVQFN48	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 48 terminals; body 7× 7 × 0.85 mm		SOT619-1
LPC845M301JHI33	HVQFN33	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 33 terminals; body 5 × 5 × 0.85 mm		SOT617-11
LPC844M201JBD64	LQFP64	Plastic low profile quad flat package; 64 leads; body 10× 10 × 1.4 mm		SOT314-2
LPC844M201JBD48	LQFP48	Plastic low profile quad flat package; 48 leads; body 7× 7 × 1.4 mm		SOT313-2
LPC844M201JHI48	HVQFN48	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 48 terminals; body 7 × 7 × 0.85 mm		SOT619-1
LPC844M201JHI33	HVQFN33	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 33 terminals; body 5 × 5 × 0.85 mm		SOT617-11

Table 2.   Ordering options

| Type number	Flash/KB	SRAM/KB	USART	I^{2}C	SPI	DAC	Capacitive Touch	GPIO	Package
LPC845M301JBD64	64	16	5	4	2	2	yes	54	LQFP64
LPC845M301JBD48	64	16	5	4	2	2	yes	42	LQFP48
LPC845M301JHI48	64	16	5	4	2	2	yes	42	HVQFN48
LPC845M301JHI33	64	16	5	4	2	1	-	29	HVQFN33
LPC844M201JBD64	64	8	2	2	2	-	-	54	LQFP64
LPC844M201JBD48	64	8	2	2	2	-	-	42	LQFP48
LPC844M201JHI48	64	8	2	2	2	-	-	42	HVQFN48
LPC844M201JHI33	64	8	2	2	2	-	-	29	HVQFN33

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 30 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

JTAG Test and Boundary Scan interface

SWD Port

CLKIN XTALOUT
XTALIN RESET

DEBUG INTERFACE
ARM Cortex M0+

General Purpose DMA controller

Clock Generation, Power Control, and other System Functions → CLKOUT

Voltage Regulator ← Vdd

GPIOs ← GPIOs AND GPOINT

IOP bus

M0 M1

P0

Flash interface

Flash 64 kB

Boot ROM 16 kB

Multilayer AHB Matrix

P1 SRAM/MTB 8 kB

P2 SRAM 8 kB

P4

SCT Timer/ PWM

DMA registers

CRC

P3

MTB slave interface

FAIM 256-bit

AHB to APB bridge

APB slave group

IOCON Registers

UARTs 0-4 → UART0,1,2, 3,4

Flash Registers (NVMC)

CapTouch → CAPT

T0 Match/ Capture → CTIMER32

SPIs 0 and 1 → SPI0,1

I2C2,3 → I2Cs 2 and 3

I2Cs 0 and 1 → I2C0,1

Periph Input Mux Selects

System control

COMP Inputs → Comparator

PMU Registers

ADC Inputs and Triggers → 12-bit ADC

DAC1 outputs ← 10-bit DAC1

DAC0 outputs ← 10-bit DAC0

FAIM Registers

PIOs ← Switch Matrix

Wakeup Timer

Multi-Rate Timer

Watchdog Osc → Windowed WDT

Note: Yellow shaded blocks support general purpose DMA

aaa-022793

Note: Yellow shaded blocks support general purpose DMA

Fig 1.   LPC84x block diagram

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **8 of 530**

2.1 How to read this chapter

The memory mapping is identical for all LPC84x parts. Different LPC84x parts support different flash and SRAM memory sizes.

2.2 General description

The LPC84x 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
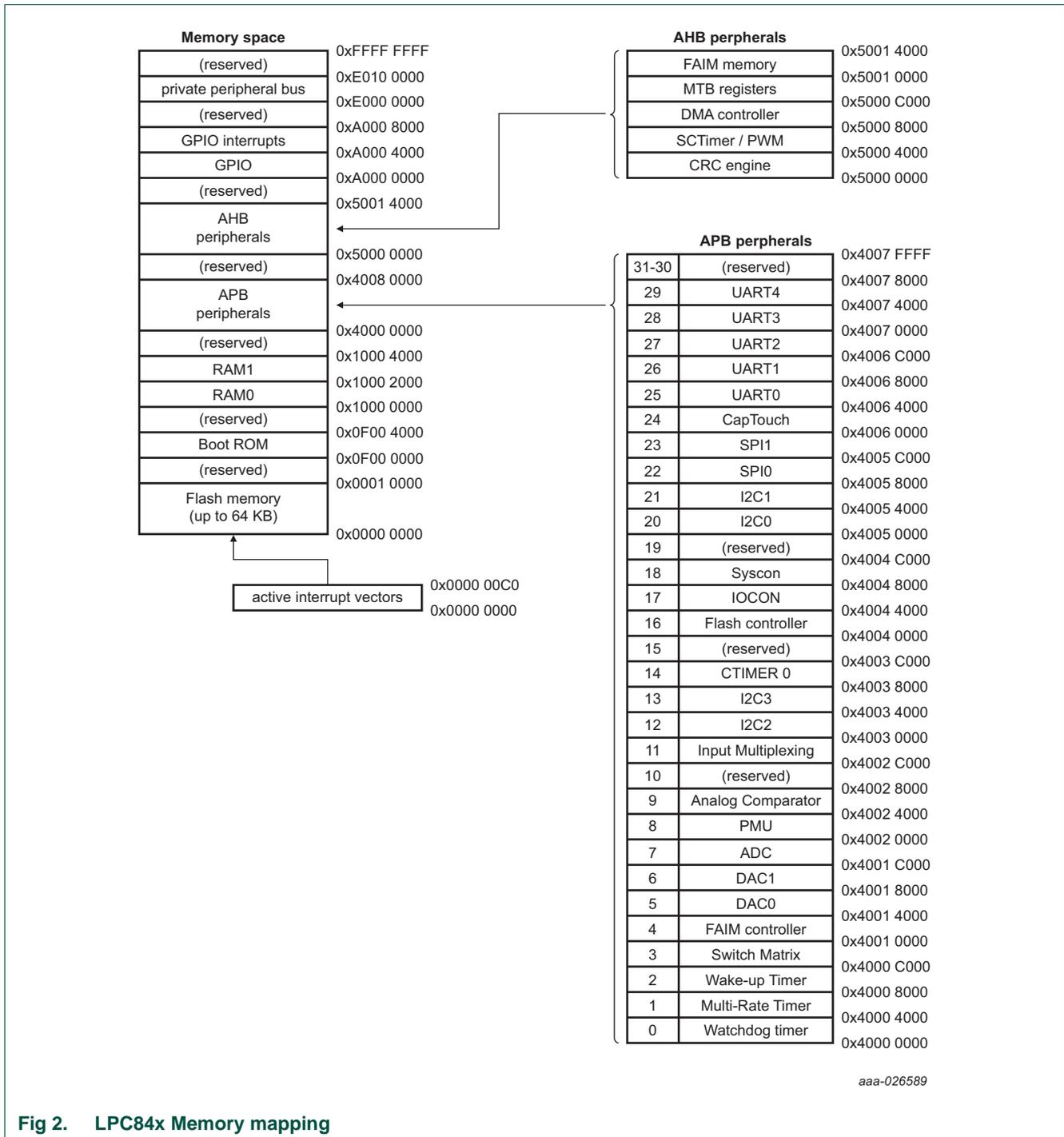
Fig 2.   LPC84x Memory mapping

2.2.2 Micro Trace Buffer (MTB)

The LPC84x supports the ARM Cortex-M0+ Micro Trace Buffer. See Section 32.5.4.

3.1 How to read this chapter

The bootloader is identical for all parts.

3.2 Features

- 16 KB on-chip boot ROM
- Contains the bootloader with In-System Programming (ISP) facility over multiple peripheral communication (UART, I^{2}C, and SPI) and the following API:
  - In-Application Programming (IAP) of flash memory.
  - Integer divide routines.
  - FAIM API.
  - FRO API.

3.3 Basic configuration

Depending on the FAIM configuration, the boot ROM sets the FRO control register to select the operating frequency accordingly. If FAIM is not programmed or contains an invalid value, the ROM begins at 12 MHz.

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.

Table 3.   Pin location in ISP mode

ISP mode	Default FAIM configuration	FAIM ISP selection (FAIM word0, bit 30 and 31), Table 5	FAIM ISP pin function (FAIM word0, bit 30 and 31)
USART ISP	PIO0_25 is UART0 TX PIO0_24 is UART0 RX Applicable when FAIM content is invalid.	0x00	See: Table 6
I^{2}C ISP	PIO0_11 is I2C0 SDA PIO0_10 is I2C0 SCL Applicable when FAIM content is invalid.	0x01	-
SPI ISP	PIO0_15 is SPI0 SCK PIO0_22 is SPI0 SSEL0 PIO0_26 is SPI0 MISO PIO0_27 is SPI0 MOSI	0x02	See: Table 6

UM11029

User manual   **Rev. 1.6 — 8 December 2017**   **11 of 530**

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, I2C slave, or SPI slave bus. The LPC84x device must be connected to a host system that provides the UART, I^{2}C or SPI 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, I^{2}C, or SPI interface. Otherwise, the bootloader checks if there is valid user code in flash. If the valid user code is not found, the bootloader checks the FAIM configuration and enters one of the ISP modes. Auto detect is selected if FAIM is invalid.

Remark: The sampling of pin the ISP entry pin can be disabled through programming flash location 0x0000 02FC (see Section 5.3.6 "Code Read Protection (CRP)").

See Chapter 5 "LPC84x 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:

- Boot loader.
- Flash In-Application Programming (IAP).
- In-System Programming (ISP) through USART, SPI, and I^{2}C.
- On-chip ROM APIs for integer divide.
- FAIM API.
- FRO API.

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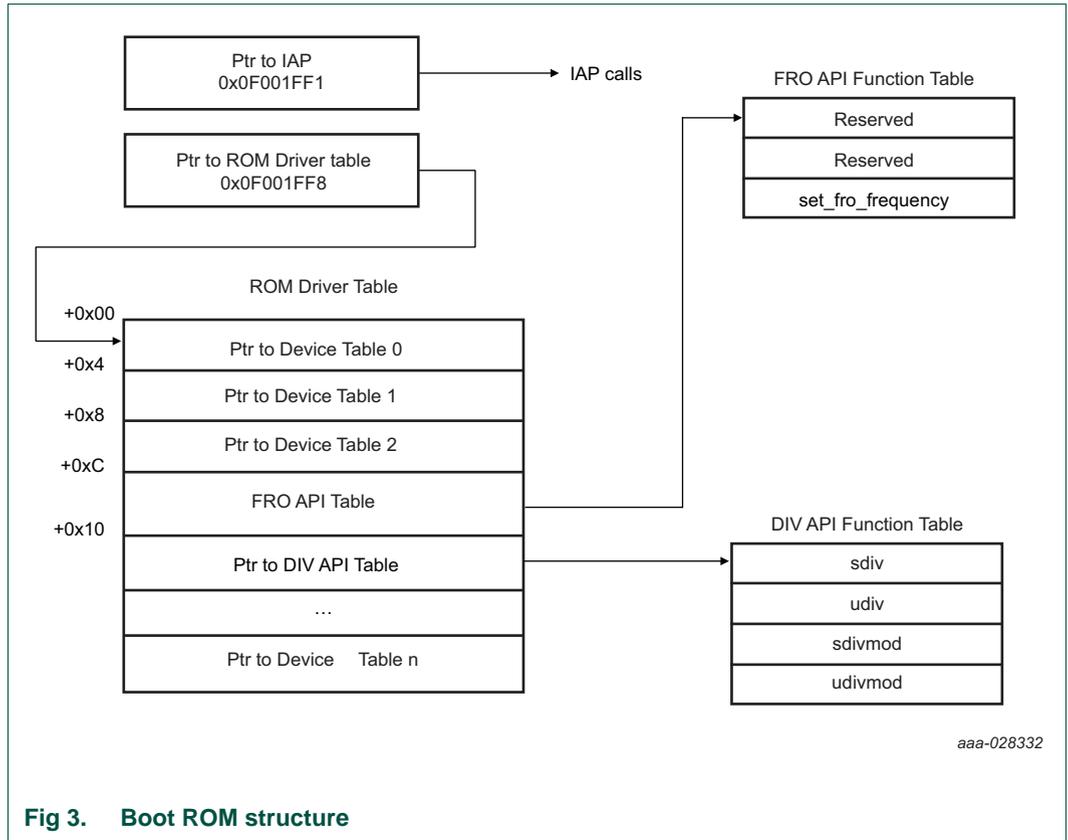
Fig 3.   **Boot ROM structure**

The boot ROM structure should be included as follows:

```
typedef struct {
    const uint32_t reserved0; /*!< Reserved */
    const uint32_t reserved1; /*!< Reserved */
    const uint32_t reserved2; /*!< Reserved */
    const uint32_t reserved2; /*!< Reserved */
    const ROM_DIV_API_T *divApiBase; /*!< Divider API function table base address */
} LPC_ROM_API_T;

#define ROM_DRIVER_BASE (0x0F001FF8)
```

Table 4.   API calls

| API	Description	Reference
Flash IAP	Flash In-Application programming	Table 37
Integer divider API	32-bit integer divide routines	Table 482
FAIM API	Fast Initialization memory programming	Section 4.2
FRO API	Free Running Oscillator configuration	Section 9.4

3.6 Functional description

3.6.1 Memory map after any reset

The boot ROM block is 16 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 5.3.7 "ISP interrupt and SRAM use". The interrupt vectors residing in the boot block of the on-chip flash memory also become active after reset, i.e., 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
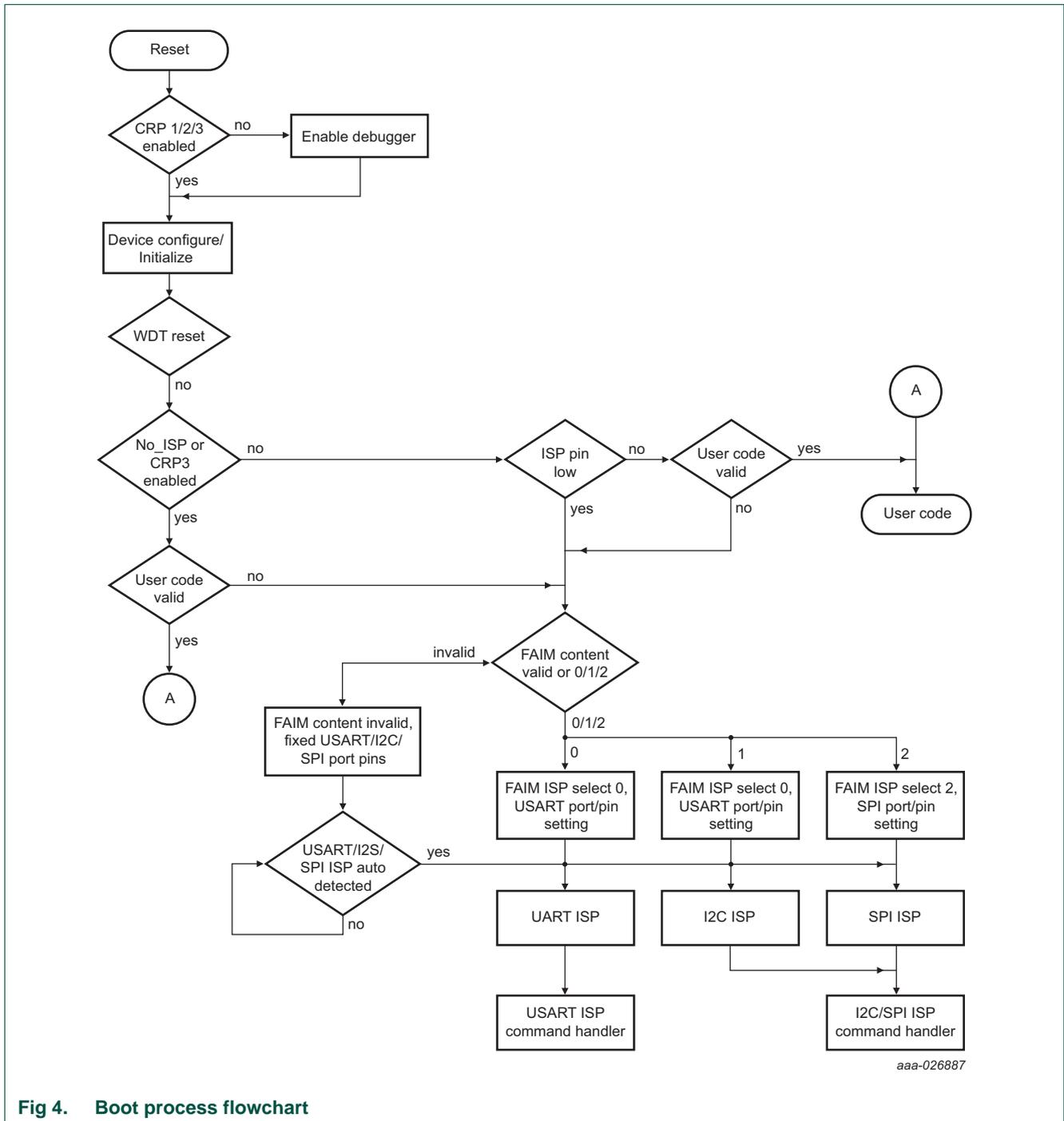
Fig 4.   Boot process flowchart

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual   **Rev. 1.6 — 8 December 2017**   **15 of 530**

4.1 How to read this chapter

The Fast Initialization Memory (FAIM) is a 256-bit memory configured as eight words (or rows) of 32-bits per word.

4.2 General description

The FAIM is a multiple time programmable (MTP), ultra low power memory, the full contents of which are read and latched immediately after reset, with no clocks required.

The FAIM contents provide a user-programmable initial configuration for aspects of the microcontroller, which take effect immediately after reset, before code begins to run. For instance, the standard I/O pads normally come out of reset with the internal pull-ups enabled. In some systems this may cause excess current to flow, until software can reconfigure the pads. However, by programming the FAIM appropriately, every pad's reset configuration can be customized. Other aspects which can be controlled by the FAIM are initial FRO divider value (low power start), serial wire debug disable, default ISP interface and pins, etc. One 32-bit FAIM row can be programmed, or read, using the ROM IAP calls FAIMWrite and FAIMRead (see Chapter 5 "LPC84x ISP and IAP" for details).

After a FAIMMWrite, a FAIMRead is required to update the output of the FAIM. Once a read has been performed, the FAIM contents are visible in the AHB Peripheral address space starting at 0x5001_0000.

For the pull-up, pull-down, and HI-Z IOCON pin configuration settings, a reset is needed to transfer the newly programmed FAIM values into the IOCON pin configuration registers. Executing a FAIMWrite followed by a FAIMRead does not update the current configuration in the IOCON pin configuration registers. Only a reset can do so. Software can at any time rewrite the IOCON pin configuration. Similarly, for SWD Disable, Low Power Start configuration, ISP interface and pin select, a reset is needed to update the actual startup configuration.

The FAIM is limited to 200 program cycles, so care must be taken to write this memory only when necessary during an end-product's development. Also, the FAIM programming voltage range is $3.0\ V \leq Vdd \leq 3.6\ V$.

Remark: If internal pull-down is enabled via FAIM on the ISP pin (PIO0_12) or the reset pin (PIO0_5), ensure to have a strong external pull-up to avoid going into ISP mode or into reset after boot-up.

4.2.1 FAIM bit definitions

The functions of FAIM bits are described in the following tables.

Table 5.   FAIM word 0 bit description

Bit		Description	Default value
0		SWD disable	0
	0	SWD enabled	
	1	SWD disabled	
1		Low Power Boot	0
	0	Normal boot (12 MHz)	
	1	Low Power Boot (1.5 MHz)	
26:2		Reserved	0
28		Reserved	0
29, 27		FAIM content valid bits	11
	00	FAIM content invalid	
	01	FAIM content invalid	
	10	FAIM content invalid	
	11	FAIM content valid	
31:30		ISP interface select	0
	00	USART0	
	01	I2C0	
	10	SPI0	
	11	Reserved	

Table 6.   FAIM word 1 bit description

Bit		Description	Default value
4:0		ISP Rx pin select (USART0 Rx, SPI MOSI)	0x18
	0x0 0x1 ... 0x1F	PIOn_0 PIOn_1 ... PIOn_31	
7:5		ISP Rx port select (USART0 Rx, SPI MOSI)	0
	0x0	Port 0	
	0x1	Port 1	
12:8		ISP Tx pin select (USART0 Tx, SPI MISO)	0x19
	0x0 0x1 ... 0x1F	PIOn_0 PIOn_1 ... PIOn_31	
15:13		ISP Tx port select (USART0 Tx, SPI MISO)	0
	0x0	Port 0	
	0x1	Port 1	
20:16		ISP clock pin select (SPI SCK)	0
	0x0 0x1 ... 0x1F	PIOnPIOn_0 PIOn_1 ... PIOn_31	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **17 of 530**

Table 6. **FAIM word 1 bit description**

Bit		Description	Default value
23:21		ISP clock port select (SPI SCK)	0
	0x0	Port 0	
	0x1	Port 1	
28:24		ISP SPI0 SSELN0 pin select	0
	0x0	PIOn_0	
	0x1	PIOn_1	
	...	...	
	0x1F	PIOn_31	
31:29		ISP SPI0 SSELN0 port select	0
	0x0	Port 0	
	0x1	Port 1	

Table 7. **FAIM word 2 bit description**

Bit		Description	Default value
31:0		Reserved	0

Table 8. **FAIM word 3 bit description**

Bit		Description	Default value
31:0		Reserved	0

Table 9. **FAIM word 4 bit description**

Bit		Description	Default value
19:0		Reserved	0
21:20		PIO1_21	0x2
	0x0	Hi-Z	
	0x1	Pull-down	
	0x2	Pull-up	
	0x3	Repeater	
23:22		PIO1_20. See description of bits of bits 21:20.	0x2
25:24		PIO1_19. See description of bits 21:20.	0x2
27:26		PIO1_18. See description of bits 21:20.	0x2
29:28		PIO1_17. See description of bits 21:20.	0x2
31:30		PIO1_16. See description of bits 21:20.	0x2

Table 10. **FAIM word 5 bit description**

Bit		Description	Default value
1:0		PIO1_15	
	0x0	Hi-Z	0x2
	0x1	Pull-down	
	0x2	Pull-up	
	0x3	Repeater	
3:2		PIO1_14. See description of bits 1:0.	0x2

Table 10.   FAIM word 5 bit description

Bit		Description	Default value
5:4		PIO1_13. See description of bits 1:0.	0x2
7:6		PIO1_12. See description of bits 1:0.	0x2
9:8		PIO1_11. See description of bits 1:0.	0x2
11:10		PIO1_10. See description of bits 1:0.	0x2
13:12		PIO1_9. See description of bits 1:0.	0x2
15:14		PIO1_8. See description of bits 1:0.	0x2
17:16		PIO1_7. See description of bits 1:0.	0x2
19:18		PIO1_6. See description of bits 1:0.	0x2
21:20		PIO1_5. See description of bits 1:0.	0x2
23:22		PIO1_4. See description of bits 1:0.	0x2
25:24		PIO1_3. See description of bits 1:0.	0x2
27:26		PIO1_2. See description of bits 1:0.	0x2
29:28		PIO1_1. See description of bits 1:0.	0x2
31:30		PIO1_0. See description of bits 1:0.	0x2

Table 11.   FAIM word 6 bit description

Bit		Description	Default value
1:0		PIO0_31	0
	0x0	Hi-Z	0x2
	0x1	Pull-down	
	0x2	Pull-up	
	0x3	Repeater	
3:2		PIO0_30. See description of bits 1:0.	0x2
5:4		PIO0_29. See description of bits 1:0.	0x2
7:6		PIO0_28. See description of bits 1:0.	0x2
9:8		PIO0_27. See description of bits 1:0.	0x2
11:10		PIO0_26. See description of bits 1:0.	0x2
13:12		PIO0_25. See description of bits 1:0.	0x2
15:14		PIO0_24. See description of bits 1:0.	0x2
17:16		PIO0_23. See description of bits 1:0.	0x2
19:18		PIO0_22. See description of bits 1:0.	0x2
21:20		PIO0_21. See description of bits 1:0.	0x2
23:22		PIO0_20. See description of bits 1:0.	0x2
25:24		PIO0_19. See description of bits 1:0.	0x2
27:26		PIO0_18. See description of bits 1:0.	0x2
29:28		PIO0_17. See description of bits 1:0.	0x2
31:30		PIO0_16. See description of bits 1:0.	0x2

Table 12.   FAIM word 7 bit description

Bit		Description	Default value
1:0		PIO0_15	0
	0x0	Hi-Z	0x2
	0x1	Pull-down	
	0x2	Pull-up	
	0x3	Repeater	
3:2		PIO0_14. See description of bits 1:0.	0x2
5:4		PIO0_13. See description of bits 1:0.	0x2
7:6		PIO0_12. See description of bits 1:0.	0x2
9:8		PIO0_11. See description of bits 1:0.	0x2
11:10		PIO0_10. See description of bits 1:0.	0x2
13:12		PIO0_9. See description of bits 1:0.	0x2
15:14		PIO0_8. See description of bits 1:0.	0x2
17:16		PIO0_7. See description of bits 1:0.	0x2
19:18		PIO0_6. See description of bits 1:0.	0x2
21:20		PIO0_5. See description of bits 1:0.	0x2
23:22		PIO0_4. See description of bits 1:0.	0x2
25:24		PIO0_3. See description of bits 1:0.	0x2
27:26		PIO0_2. See description of bits 1:0.	0x2
29:28		PIO0_1. See description of bits 1:0.	0x2
31:30		PIO0_0. See description of bits 1:0.	0x2

5.1 How to read this chapter

All LPC84x 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 6.4.1.

5.2 Features

- In-System Programming: In-System programming (ISP) is programming or reprogramming the on-chip flash memory, using the boot loader software and USART, I^{2}C, or SPI serial port. 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.
- Flexible ISP mode and port pin selection through FAIM memory configuration.
- Small size (64 byte) page erase programming.

5.3 General description

5.3.1 Boot loader

For the boot loader operation and boot pin, see Chapter 3 "LPC84x Boot Process".

The boot loader version can be read by ISP/IAP calls (see Section 5.5.13 or Section 5.6.6).

5.3.2 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 5.3.7.

UM11029

© NXP B.V. 2017. All rights reserved.

User manual   **Rev. 1.6 — 8 December 2017**   **21 of 530**

5.3.3 Flash content protection mechanism

The LPC84x 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.

5.3.4 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 will check the FAIM configuration and enter UART/I2C/SPI ISP mode automatically.

5.3.5 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 13. LPC84x flash configuration

Sector number	Sector size [KB]	Page number	Address range	16 KB flash	32 KB flash
0	1	0 -15	0x0000 0000 - 0x0000 03FF	yes	yes
1	1	16 - 31	0x0000 0400 - 0x0000 07FF	yes	yes
2	1	32 - 47	0x0000 0800 - 0x0000 0BFF	yes	yes
3	1	48 - 63	0x0000 0C00 - 0x0000 0FFF	yes	yes
4	1	64 - 79	0x0000 1000 - 0x0000 13FF	yes	yes
5	1	80 - 95	0x0000 1400 - 0x0000 17FF	yes	yes

Table 13. **LPC84x flash configuration**

Sector number	Sector size [KB]	Page number	Address range	16 KB flash	32 KB flash
6	1	96 - 111	0x0000 1800 - 0x0000 1BFF	yes	yes
7	1	112 - 127	0x0000 1C00 - 0x0000 1FFF	yes	yes
8	1	128 - 143	0x0000 2000 - 0x0000 23FF	yes	yes
9	1	144 - 159	0x0000 2400 - 0x0000 27FF	yes	yes
10	1	160 - 175	0x0000 2800 - 0x0000 2BFF	yes	yes
11	1	176 - 191	0x0000 2C00 - 0x0000 2FFF	yes	yes
12	1	192 - 207	0x0000 3000 - 0x0000 33FF	yes	yes
13	1	208 - 223	0x0000 3400 - 0x0000 37FF	yes	yes
14	1	224 - 239	0x0000 3800 - 0x0000 3BFF	yes	yes
15	1	240 - 255	0x0000 3C00 - 0x0000 3FFF	yes	yes
16	1	256 - 271	0x0000 4000 - 0x0000 43FF	-	yes
17	1	272 - 287	0x0000 4400 - 0x0000 47FF	-	yes
18	1	288 - 303	0x0000 4800 - 0x0000 4BFF	-	yes
19	1	304 - 319	0x0000 4C00 - 0x0000 4FFF	-	yes
20	1	320 - 335	0x0000 5000 - 0x0000 53FF	-	yes
21	1	336 - 351	0x0000 5400 - 0x0000 57FF	-	yes
22	1	352 - 367	0x0000 5800 - 0x0000 5BFF	-	yes
23	1	368 - 383	0x0000 5C00 - 0x0000 5FFF	-	yes
24	1	384 - 399	0x0000 6000 - 0x0000 63FF	-	yes
25	1	400 - 415	0x0000 6400 - 0x0000 67FF	-	yes
26	1	416 - 431	0x0000 6800 - 0x0000 6BFF	-	yes
27	1	432 - 447	0x0000 6C00 - 0x0000 6FFF	-	yes
28	1	448 - 463	0x0000 7000 - 0x0000 73FF	-	yes
29	1	464 - 479	0x0000 7400 - 0x0000 77FF	-	yes
30	1	480 - 495	0x0000 7800 - 0x0000 7BFF	-	yes
31	1	496 - 511	0x0000 7C00 - 0x0000 7FFF	-	yes

5.3.6 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.

User manual   **Rev. 1.6 — 8 December 2017**   **23 of 530**

Table 14. USART ISP command limitations in CRP modes

Name	Pattern programmed in 0x0000 02FC	Description
NO_ISP	0x4E69 7370	Prevents sampling of the pins for entering ISP mode. ISP sampling pin is available for other applications.
CRP1	0x1234 5678	Access to 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.
CRP2	0x8765 4321	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	0x4321 8765	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.**

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.

5.3.6.1 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.

5.3.6.2 ISP entry configuration and detection

The LPC84x UART/I2C/SPI ISP mode allows programming and reprogramming of the internal FLASH via a set of commands on the UART, I2C slave, or SPI slave buses. Two bits of the FAIM flash are used for ISP mode selection.

If FAIM content is invalid, the default ISP selection is USART/I2C/SPI or auto detection mode. In auto detection mode, the LPC84x enables all three interfaces on the fixed GPIO port and pins, and selects the first one that has either a successful auto baud detection on USART or a valid probe message response on I2C or SPI.

If FAIM content is valid, USART, I2C, or SPI ISP mode is configured in the FAIM. Additional SWM configuration for the interface and pins through FAIM is required. For example, if FAIM ISP selection is 0x02, the GPIO port and pin information for SPI0, such as SSEL, SCK, MOSI, and MISO is provided by the FAIM. The boot ROM reads the GPIO port and pin information from the FAIM and writes to the SWM and IOCON registers accordingly as part of the SPI initialization.

5.3.7 ISP interrupt and SRAM use

5.3.7.1 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.

5.3.7.2 RAM used by ISP command handlers

The stack of UART ISP commands is located at address 0x1000 0600. The maximum stack usage is 1280 bytes (0x500) and grows downwards.

The DMA is used by the SPI ISP mode. The DMA descriptor table location is located at address 0x1000 0600. The DMA table size is 512 bytes (0x200) and grows upwards. Therefore, depending on the ISP mode entered, the maximum RAM used by ISP mode is 2 K starting from the address 0x1000 0000.

Memory for the USART and I2C/SPI ISP commands is allocated dynamically.

UM11029

User manual

Rev. 1.6 — 8 December 2017

25 of 530

5.4 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.

5.4.1 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. In USART ISP mode, the part is clocked by the FAIM configuration and the crystal frequency is ignored.

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 5.5 "USART ISP commands".

5.4.2 USART ISP command format

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

5.4.3 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).

5.4.4 USART ISP data format

The data stream is in plain binary format.

5.5 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 15. USART ISP command summary

ISP Command	Usage	Section
Unlock	U <Unlock Code>	5.5.1
Set Baud Rate	B <Baud Rate> <stop bit>	5.5.2
Echo	A <setting>	5.5.3
Write to RAM	W <start address> <number of bytes>	5.5.4
Read Memory	R <address> <number of bytes>	5.5.5
Prepare sectors for write operation	P <start sector number> <end sector number>	5.5.6
Copy RAM to flash	C <Flash address> <RAM address> <number of bytes>	5.5.7
Go	G <address> <Mode>	5.5.8
Erase sector(s)	E <start sector number> <end sector number>	5.5.9
Erase page(s)	X <start page number> <end page number>	5.5.10
Blank check sector(s)	I <start sector number> <end sector number>	5.5.11
Read Part ID	J	5.5.12
Read Boot code version	K	5.5.13
Compare	M <address1> <address2> <number of bytes>	5.5.14
ReadUID	N	5.5.15
Read CRC checksum	S <address> <number of bytes>	5.5.16
Read flash signature	Z	5.5.17
Read/Write FAIM Page	O	5.5.18

Table 16 lists the supported USART ISP commands for each CRP level.

Table 16. 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

Table 16. ISP commands allowed for different CRP levels

ISP command	CRP1	CRP2	CRP3 (no entry in ISP mode allowed)
Erase sector(s)	yes; sector 0 can only be erased when all sectors are erased.	yes; all sectors only	n/a
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	no	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
Read/Write FAIM page	yes	yes	n/a

5.5.1 Unlock

Table 17. 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.

5.5.2 Set Baud Rate

Table 18. USART ISP Set Baud Rate command

Command	B
Input	Baud Rate, FAIM configuration dependant: 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.

5.5.3 Echo

Table 19. USART ISP Echo command

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

5.5.4 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 20. 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.

5.5.5 Read Memory

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

Table 21. 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.

5.5.6 Prepare sectors for write operation

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

Table 22. 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.

5.5.7 Copy RAM to flash

When writing to the flash, the following limitations apply:

1. 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).

2. One page consists of 16 flash words (lines), and the smallest amount that can be modified per flash write is one flash word (one line). This limitation exists because ECC is applied during the flash write operation, see Section 5.3.3.

3. To avoid write disturbance (a mechanism intrinsic to flash memories), an erase should be performed after 16 consecutive writes inside the same page. Note that the erase operation then erases the entire sector.

   Remark: Once a page has been written to 16 times, it is still possible to write to other pages within the same sector without performing a sector erase (assuming that those pages have been erased previously).

Table 23. 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 5.3.3 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.

5.5.8 Go

Table 24. 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.

5.5.9 Erase sectors

Table 25.   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.

5.5.10 Erase pages

Table 26.   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.

5.5.11 Blank check sectors

Table 27.   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.

5.5.12 Read Part Identification number

Table 28. **USART ISP Read Part Identification command**

Command	J
Input	None.
Return Code	CMD_SUCCESS followed by part identification number (see Table 29).
Description	This command is used to read the part identification number.

Table 29. **LPC84x device ID register values**

Part number	Part ID
LPC844M201JBD48	0x00008442
LPC844M201JBD64	0x00008441
LPC844M201JHI33	0x00008444
LPC844M201JHI48	0x00008442
LPC845M301JBD48	0x00008452
LPC845M301JBD64	0x00008451
LPC845M301JHI33	0x00008454
LPC845M301JHI48	0x00008453

5.5.13 Read Boot code version number

Table 30. **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.

5.5.14 Compare

Table 31. **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.

5.5.15 ReadUID

Table 32. 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.

5.5.16 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 33. 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>"

5.5.17 Read flash signature

Get the signature for the flash memory, using an internal flash signature generator (see Chapter 6 "LPC84x Flash signature generator"). When CRP1 is enabled, only the signature of the entire flash can be read and no parameters can be passed.

Table 34.   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.

UM11029

© NXP B.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **35 of 530**

5.5.18 Read/write FAIM page

This command is used to read/write FAIM page.

Table 35. USART ISP Read/Write FAIM Page command

Command	O
Input	**FAIM access:** 0 = FAIM Page read 1 = FAIM Page write **FAIM page number**: 0 - 7 **FAIM configuration value:** user defined (if FAIM access is page write)
Return Code	CMD_SUCCESS | PARAM_ERROR | INVALID_COMMAND | INVALID_PAGE
Description	This command is used to read/write FAIM page
Example	"O 0 5<CR><LF>" reads the FAIM page 5. "O 1 7<CR><LF>" initiates a write to FAIM page 7. When the system is ready to receive the data it responds with 0<CR><LF> [CMD_SUCCESS]. The host should send a raw binary data of 4 bytes. FAIM programming starts when the fourth byte is received and the ISP command responds with 0<CR><LF> [CMD_SUCCESS] error code after the programming is completed successfully.

5.5.19 ISP/IAP Error codes

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

Table 36. 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.

Table 36. ISP/IAP Error codes

Return Code	Error code	Description
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
0x1E	FAIM_NO_POWER	FAIM not turned on in the PDRUNCFG register.
0x1F	FAIM_NO_CLOCK	FAIM clock disabled in the AHBCLKCTRL register.

5.6 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 0x0F001FF0 and it is thumb code, therefore called as 0x0F001FF1 by the Cortex-M4 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-M4.

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 *)(0x0F001FF1)
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.

The flash memory is not accessible during a write or erase operation. IAP commands, which results in a flash write/erase operation, use 32 bytes of space in the top portion of the on-chip RAM for execution. The user program should not be use this space if IAP flash programming is permitted in the application.

Table 37.   IAP Command Summary

IAP Command	Command code	Section
Prepare sector(s) for write operation	50 (decimal)	5.6.1
Copy RAM to flash	51 (decimal)	5.6.2
Erase sector(s)	52 (decimal)	5.6.3
Blank check sector(s)	53 (decimal)	5.6.4
Read Part ID	54 (decimal)	5.6.5
Read Boot code version	55 (decimal)	5.6.6
Compare	56 (decimal)	5.6.7
Reinvoke ISP	57 (decimal)	5.6.8
Read UID	58 (decimal)	5.6.9
Erase page(s)	59 (decimal)	5.6.10
Read Signature	73 (decimal)	5.6.11
Read FAIM Page	80 (decimal)	5.6.12
Write FAIM Page	81 (decimal)	5.6.13
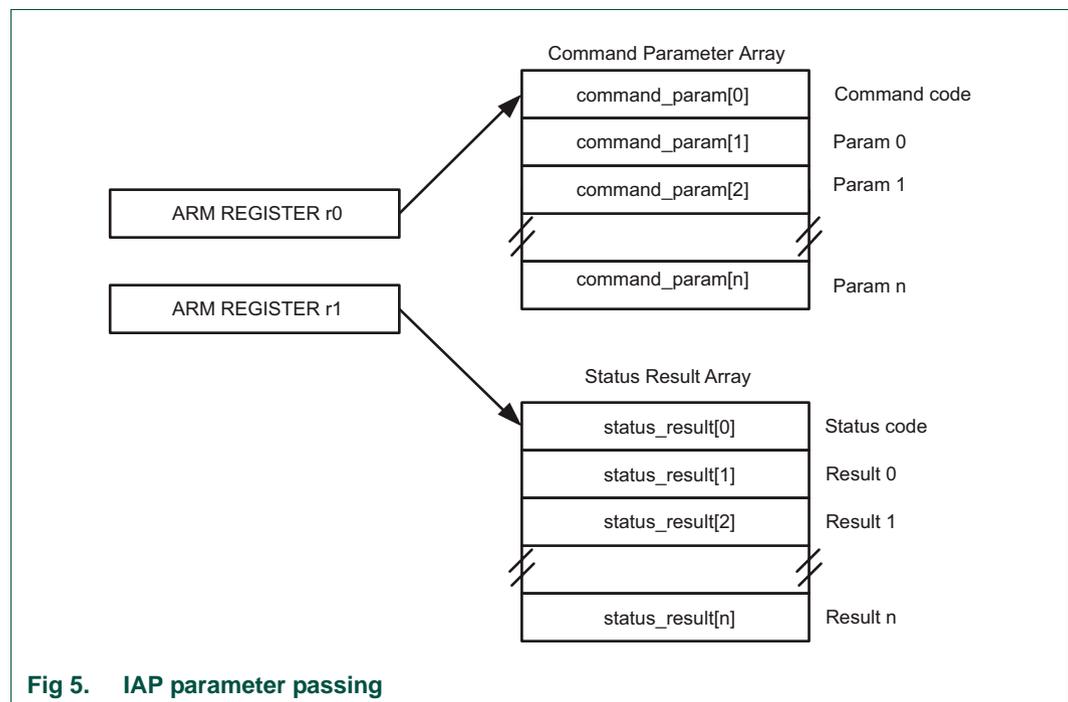
Fig 5.   IAP parameter passing

5.6.1 Prepare sector(s) for write operation

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

Table 38. 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.

5.6.2 Copy RAM to flash

See Section 5.5.7 for limitations on the write-to-flash process.

Table 39. 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 5.3.3 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.

5.6.3 Erase Sector(s)

Table 40. 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.

5.6.4 Blank check sector(s)

Table 41. 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.

5.6.5 Read Part Identification number

Table 42. 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.

5.6.6 Read Boot code version number

Table 43. 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.

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

Table 44. 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.

5.6.8 Reinvoke ISP

Table 45. Reinvoke ISP

Command	Compare
Input	**Command code: 57 (decimal)**
	Param0(mode): ISP interface selection
	0 - Auto or UIS ISP (only applicable when FAIM configuration is invalid.)
	1 - USART ISP (Match FAIM ISP configuration if it is valid.)
	2 - I2C ISP (Match FAIM ISP configuration if it is valid.)
	3 - SPI ISP (Match FAIM ISP configuration if it is valid.)
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.
	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.
	If using USART ISP mode, enable the clocks to the default before calling this command.

5.6.9 ReadUID

Table 46. 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.

5.6.10 Erase page

Table 47. 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.

5.6.11 Read Signature

Table 48. 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 5.5.17 "Read flash signature" and Chapter 6 for more information. **Remark:** See Section 6.5.1.2 "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.

5.6.12 Read FAIM Page

Table 49. IAP Read FAIM page command

Command	Read FAIM page
Input	**Command code: 80 (decimal)** **Param0:** FAIM page number. **Param1:** Memory address to store the value read from FAIM.
Status code	CMD_SUCCESS | ADDR_NOT_MAPPED
Result	None.
Description	This command reads a given page of FAIM into the memory provided.

5.6.13 Write FAIM Page

Table 50. IAP Write FAIM page command

Command	Write FAIM page
Input	**Command code: 81 (decimal)** **Param0:** FAIM page number. **Param1:** Pointer to the memory address holding data to be stored on to FAIM page.
Status code	CMD_SUCCESS | ADDR_NOT_MAPPED
Result	None.
Description	This command retrieves data from memory at the address given by Param1 and written to the FAIM page.

5.6.14 IAP Error Codes

See .

UM11029 All information provided in this document is subject to legal disclaimers.

User manual **Rev. 1.6 — 8 December 2017** **44 of 530**

5.7 I2C and SPI ISP commands

The I2C/SPI ISP allows programming and reprogramming of the internal flash via a set of commands on the I2C slave or SPI slave buses of the LPC84x. These need to be connected to a host system that provides the I2C or SPI master connections to the LPC84x.

5.7.1 Dual purpose of the ISP/ISP_IRQ pin

The ISP pin is a special function pin that is switches function state once I2C/SPI ISP mode is entered and the host interface has been selected. Once the host interface has been selected, the ISP pin becomes an output pin used for indicating to the host system that a command can be serviced. During this phase, the pin is called ISP_IRQ. A low state on this pin indicates to the host that the LPC84x needs servicing.

Notes on ISP input to ISP_IRQ output switching

After reset, ISP is set to an input. When the FAIM configuration is invalid and host interface is not selected, on entry to the USART/I2C/SPI ISP mode from device reset, the interface is in auto-detection mode and the pin remains an input. Once the host interface sends the first probe command via I2C or SPI interface, and it is accepted by the LPC84x, then the interface is detected, the ISP pin switches to an output high as ISP_IRQ.

When the FAIM configuration is valid and interface selection is either I2C or SPI ISP mode, this pin switches to an output high immediately without a probe message.

When using the 'Reinvoke ISP' command, the host interface is selected as part of the IAP command, the pin switches to an output high immediately without a probe message.

Care must be used that any external signal applied to the ISP pin to drive the ISP pin low should be done with a pull-down resistor and not tied to ground, so the ISP_IRQ output does not short directly to a low state when driving high. Host systems that drive this pin to selectively enable I2C/SPI ISP mode should drive this pin via a resistor.

5.7.2 I2C/SPI ISP mode transaction protocol

This section explains the high-level protocol used with the I2C and SPI interfaces. A typical transaction starts with the host sending a command packet, the LPC84x processing the command packet, the LPC84x optionally asserting the ISP_IRQ line low when processing is complete, and then the host system getting the response packet. The LPC84x will hold the ISP_IRQ pin asserted (low) until the host system requests the response packet. The process is shown in Figure 6. Not all commands may follow this protocol – some commands may have no response or may not assert the ISP_IRQ pin.
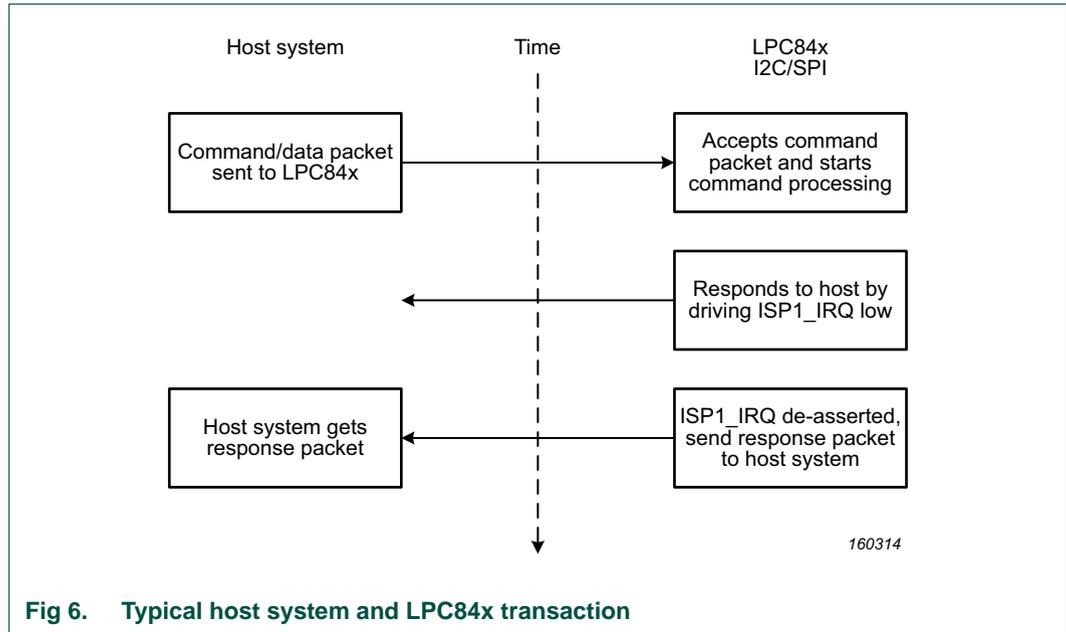
Fig 6. **Typical host system and LPC84x transaction**

5.7.3 I2C ISP mode transaction protocol

The LPC84x will respond to a host system on I2C addresses 0x18, 0x1C, 0x30, and 0x38. The host system's I2C master clock rate can be as high as 400 kHz. The LPC84x may extend the I2C clock to delay the I2C master if it needs more time to perform an operation.

When using I2C, the command is given to the LPC84x by an I2C write transaction. The entire command packet is accepted by the LPC84x and then processing starts. Once the LPC84x has completed processing, it may drive the ISP1_IRQ line low until the host system issues an I2C read transaction to get the response.

5.7.4 SPI ISP mode transaction protocol

The LPC84x will respond to a host system on the configured SPI interface. A transfer is started once SSEL goes low on the LPC84x. The SPI clock to the LPC84x should not exceed 2MHz. SPI SSEL to first clock timing should not be less than 100uS. SPI transfer configuration should be SPI Mode 0 with 8 data bits.

SPI transactions are bi-directional. During the command packet phase, the host system should ignore the read data (send data from LPC84x). During the response phase, the LPC84x will ignore the read data (send data from host).

Although SPI is bi-directional, the command and response packet phases only send data one way for each phase. During the command packet phase, a single SPI transfer occurs where the command and data is sent from the host system. In this phase, SSEL is asserted low, the command packet is sent, and then SSEL is deasserted. The host system then waits for ISP1_IRQ to go low from the LPC84x. Once ISP1_IRQ goes low, the host system then asserts SSEL low, the response packet is read, and then SSEL is deasserted. The LPC84x will deassert ISP1_IRQ at the start of the response packet prior to the response packet completing transfer.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **46 of 530**

5.7.5 I2C/SPI operations allowed for CRP systems

If CRP is enabled (CRP1, CRP2, or CRP3) then the I2C/SPI commands may be limited in functionality. See Table 51 for limitations of I2C/SPI ISP commands when CRP is enabled.

If NO_ISP mode is used in the application, the I2C/SPI ISP mode cannot be started when the ISP0 pin is asserted. The only way to recover a system when this happens is to erase flash using the SWD interface or via the IAP commands. The application can also use the 'Re-invoke ISP' command to get back into the I2C/SPI ISP mode.

Table 51. ISP commands allowed for different CRP levels

ISP command	CRP disabled	CRP1	CRP2	CRP3 (no entry in ISP mode allowed)
SH_CMD_GET_VERSION	yes	yes	yes	n/a
SH_CMD_RESET	yes	yes	yes	n/a
SH_CMD_BOOT	yes	yes	yes	n/a
SH_CMD_CHECK_IMAGE	yes	yes	yes	n/a
SH_CMD_PROBE	yes	yes	yes	n/a
SH_CMD_WRITE_BLOCK	yes	yes; not sector 0	no	n/a
SH_CMD_READ_BLOCK	yes	no	no	n/a
SH_CMD_SECTOR_ERASE	yes	yes; not sector 0	no	n/a
SH_CMD_PAGE_ERASE	yes	yes; not sector 0	no	n/a
SH_CMD_PAGE_WRITE	yes	yes; not sector 0	no	n/a
SH_CMD_PAGE_READ	yes	no	no	n/a
SH_CMD_WRITE_SUBBLOCK	yes	yes; not sector 0	no	n/a
SH_CMD_READ_SUBBLOCK	yes	no	no	n/a
SH_CMD_BULK_ERASE	yes	yes; sector 0 can be erased only if all sectors are erased.	yes; full device only	n/a
SH_CMD_WRITE_RAM	yes	yes; partially $\geq$ 0x1000 0800	no	n/a
SH_CMD_GOTO	yes	no	no	n/a
SH_CMD_FAIM_READ	yes	yes	yes	n/a
SH_CMD_FAIM_WRITE	yes	yes	yes	n/a

UM11029

User manual **Rev. 1.6 — 8 December 2017** **47 of 530**

5.8 I2C/SPI ISP mode commands, data, and responses

All of the supported commands, associated structures and data formats for those commands, and responses are explained in this section

Table 52.   I2C/SPI ISP command summary

ISP Command	Command #	Section
Get Version	0xA1	5.8.1
Reset device	0xA2	5.8.2
Boot image	0xA3	5.8.3
Check image	0xA4	5.8.4
Probe	0xA5	5.8.5
Write block	0xA6	5.8.6
Read block	0xA7	5.8.7
Sector erase	0xA8	5.8.8
Page erase	0xA9	5.8.9
Page write	0xAA	5.8.10
Page read	0xAB	5.8.11
Write sub-block	0xAC	5.8.12
Read sub-block	0xAD	5.8.13
Bulk erase	0xAE	5.8.14
Write RAM	0xB0	5.8.15
GOTO	0xB1	5.8.16
FAIM read	0xBE	5.8.17
FAIM write	0xBF	5.8.18

5.8.1 SH_CMD_GET_VERSION (0xA1) command

This command can be used to get the version number of the I2C/SPI ISP processor.

Table 53.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA1	'Get Version' command identifier

Table 54.   Response packet

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA1	Processed command identifier
length	0x2	0x2	0x2	Length of the response packet
major	0x4	0x1	Version	Major version
minor	0x5	0x1	Version	Minor version

5.8.2 SH_CMD_RESET (0xA2) command

This command can be used to reset the LPC84x. This command has no response.

Table 55. Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA2	'Reset' command identifier

5.8.3 SH_CMD_BOOT (0xA3) command

This command can be used to boot the application currently programmed into flash, boot address starting at 0x0. This command has a response only if the boot image does not have a valid checksum.

Table 56. Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA3	'Boot' command identifier

Table 57. Response packet

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA3	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet on an error
errorCode	0x4	0x4	Error code	Boot prevention due to invalid checksum.

5.8.4 SH_CMD_CHECK_IMAGE (0xA4) command

This command can be used to check the checksum value of the application currently programmed into FLASH. The command verifies the checksum of the image. If the checksum is valid, a response value of 0 is returned, otherwise, the checksum value is returned.

Table 58. Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA4	'Check image' command identifier

Table 59. Response packet

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA4	Processed command identifier
length	0x2	0x2	0x04	Length of the response packet
checksum	0x4	0x4	CRC32 value	0x0 if match, otherwise the checksum value.

5.8.5 SH_CMD_PROBE (0xA5) command

The probe command is used to select either the I2C or SPI interface when in auto-detection mode. This command is required when booting into the I2C/SPI ISP mode from a reset condition. The probe command data is accepted on the supported I2C or SPI

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP B.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **49 of 530**

ISP pins. Once the data is checked, the ISP1_IRQ line is driven LOW. The interface that was used for the probe command becomes the active interface and the other interface is disabled.

The probe command is optional when the I2C or SPI ISP mode is re-invoked from an application using the 'Re-invoke ISP' IAP command. Only the interface selected with the 'Re-invoke ISP' IAP command will be active for the optional probe command.

The host system should repeatedly send the probe command to the LCP84x via one of the supported interfaces until the LPC84x asserts the ISP1_IRQ pin LOW.

Table 60. Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA5	'Probe' command identifier
ifSel	0x1	0x1		Host interface type and port. Must match selected interface used to the LPC84x. 1 – I2C0 port 4 – SPI0 port
Reserved0	0x2	0x1		Set to 0x00
Reserved1	0x3	0x1		Set to 0x00
Reserved2	0x4	0x1		Set to 0x00
Reserved3	0x5	0x1		Set to 0x00
Reserved4	0x6	0x1		Set to 0x00
checksum	0x7	0x1		XOR of all the 7 bytes above.

Table 61. Response packet success

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA5	Processed command identifier
length	0x2	0x2	0x00	On Success, the XOR of the 7 bytes is zero and the interface number matches. Set to 0x00.

Table 62. Response packet error

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA5	Processed command identifier
length	0x2	0x2	0x04	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.6 SH_CMD_WRITE_BLOCK (0xA6) command

The write block command is used to write a block of data to flash. A block of data is 128 bytes. If a block number crosses a sector boundary, the LPC84x automatically erases the sector prior to the write operation.

Table 63.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA6	'Write block' command identifier
crcCheck	0x1	0x1		0 – Do CRC check for this packet 1 - Ignore CRC field for this packet
blockNum	0x2	0x2		Flash block number in which the appended data to be programmed. For example to program flash block 0x8000, this parameter should be set to 256.
data	0x4	Block_size		Data to be programmed in flash.
checksum	Block_size + 4	0x4	CRC32	CRC32 of the packet excluding this field. Set this field to 0 if crcCheck is set to 1.

Table 64.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA6	Processed command identifier
length	0x2	0x2	0x00	On success this field is set to 0.

Table 65.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA6	Processed command identifier
length	0x2	0x2	0x00	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.7   SH_CMD_READ_BLOCK (0xA7) command

The read block command is used to read a block of data from flash. A block of data is 128 bytes.

Table 66.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA7	'Read block' command identifier
reserved	0x1	0x1		Should be zero
blockNum	0x2	0x2		Flash block number to read. For example to read flash block 0x8000, this parameter should be set to 256.

Table 67.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA7	Processed command identifier
length	0x2	0x2	Block_size + 4	Length of the response packet. Block_size + 4 = success
data	0x4	Block_size		Flash block content
checksum	Block_size + 4	0x4	CRC32	CRC32 of the packet excluding this field

Table 68.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA7	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet. 0x4 = failure
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.8 SH_CMD_SECTOR_ERASE (0xA8) command

The sector erase command is used to erase a sector in flash. A sector size is 1K bytes.

Table 69.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA8	'Erase sector' command identifier
reserved	0x1	0x1		Should be zero.
sectorNum	0x2	0x2		Flash sector number to be erased. For example to erase flash sector at 0x2000, this parameter should be set to 8.

Table 70.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA8	Processed command identifier
length	0x2	0x2	0x00	On success this field is set to 0.

Table 71.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA8	Processed command identifier
length	0x2	0x2	0x4	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.9 SH_CMD_PAGE_ERASE (0xA9) command

The page erase command is used to erase a page in flash. A page size is 64 bytes.

Table 72.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xA9	'Page erase' command identifier
reserved	0x1	0x1		Should be zero
pageNum	0x2	0x2		Flash page number to be erased. For example to erase flash page at 0x8000, this parameter should be set to 512.

Table 73.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA9	Processed command identifier
length	0x2	0x2	0x00	Length of the response packet. 0x0 = success

UM11029

User manual **Rev. 1.6 — 8 December 2017** **52 of 530**

Table 74.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xA9	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet. 0x4 = failure
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.10   SH_CMD_PAGE_WRITE (0xAA) command

The page write command is used to write a page in flash. A page size is 64 bytes.

Table 75.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xAA	'Page write' command identifier
crcCheck	0x1	0x1		0 – Do CRC check for this packet. 1 - Ignore CRC field for this packet.
pageNum	0x2	0x2		Flash page number in which the appended data to be programmed. For example to program flash page at 0x8000, this parameter should be set to 512.
data	0x4	Page_size		Data to be programmed in flash.
checkSum	Page_size+4	0x4	CRC32	CRC32 of the packet excluding this field. Set this field to 0 if crcCheck is set to 1.

Table 76.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAA	Processed command identifier
length	0x2	0x2	0x00	On success this field is set to 0

Table 77.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAA	Processed command identifier
length	0x2	0x2	0x4	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.11   SH_CMD_PAGE_READ (0xAB) command

The page read command is used to read a page in flash. A page size is 64 bytes.

Table 78.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xAB	'Page read' command identifier
crcCheck	0x1	0x1		0 – Do CRC check for this packet. 1 - Ignore CRC field for this packet.
pageNum	0x2	0x2		Flash page number from which the data to be read.

Table 79.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAB	Processed command identifier
length	0x2	0x2	Page_size+4	On success this field is set to Page_size+4
data	0x4	Page_size		Flash page content.
checkSum	Page_size+4	0x4	CRC32	CRC32 of the packet excluding this field

Table 80.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAB	Processed command identifier
length	0x2	0x2	0x04	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.12 SH_CMD_WRITE_SUBBLOCK (0xAC) command

The write sub-block command is used for queuing data for a full flash block write. It is used when the host system cannot send the entire data block to the LPC84x in a single I2C transfer using the 'Write block' command. When using this command, multiple sub-blocks are sent to the LPC84x in sequential order for the block. The LPC84x collects all the packets and performs the flash write once the last packet is received. If any other commands are sent between 'Write sub-block' commands, the collected buffers are discarded and the operations needs to restart. If a sub-block number crosses a sector boundary, the LPC84x automatically erases the sector prior to the write operation.

Table 81.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xAC	'Write sub-block' command identifier
subBlock	0x1	0x1		Bit 0: If set, CRC check is not done for this packet. Bits [5:1]: Specifies the sub-block number. Bits [7:6]: Specifies the sub-block size. 00 – 32 bytes 01 – 64 bytes 10 – reserved 11 – reserved
blockNum	0x2	0x2		Flash block number in which the appended data to be programed. For example to program sub-block at 0x8000 this parameter should be set to 256.
data	0x4	Sub-block size		Data to be programmed in flash.
checkSum	Sub-block size + 4	0x4		CRC32 of the packet excluding this field. Set this field to 0 if crcCheck is set to 1.

Table 82.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAC	Processed command identifier
length	0x2	0x2	0x00	On success this field is set to 0

UM11029

© NXP B.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **54 of 530**

Table 83. Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAC	Processed command identifier
length	0x2	0x2	0x4	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.13 SH_CMD_ READ_SUBBLOCK (0xAD) command

The read sub-block command is used for reading partial data from a full flash block write. It is used when the host system cannot receive the entire data block to the LPC84x in a single I2C transfer using the 'Read block' command.

Table 84. Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xAD	'Read subblock' command identifier
subBlock	0x1	0x1		Bit 0: If set, CRC check is not done for this packet. Bits [5:1]: Specifies the sub-block number Bits [7:6]: Specifies the sub-block size 00 – 32 bytes 01 – 64 bytes 10 – reserved 11 – reserved
blockNum	0x2	0x2		Flash block number to read. For example to read sub-block at 0x8000 this parameter should be set to 256.

Table 85. Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAD	Processed command identifier
length	0x2	0x2	Sub-block size + 4	On success this field is set to (Sub-block size + 4)
data	0x4	0x2	Sub-block size	Flash data
checksum	Sub-block size + 4	0x4		CRC32 of the packet excluding this field

Table 86. Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAD	Processed command identifier
length	0x2	0x2	0x4	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.14 SH_CMD_BULK_ERASE (0xAE) command

The bulk erase command is used to erase more than one sector (from a start sector to an end sector).

UM11029

© NXP B.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **55 of 530**

Table 87.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xAE	'Bulk erase' command identifier
reserved	0x1	0x1	-	-
startSec	0x2	0x1	-	Starting sector number to erase
endsec	0x3	0x1	-	End sector number to erase

Table 88.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAE	Processed command identifier
length	0x2	0x2	0x00	On success this field is set to 0

Table 89.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xAE	Processed command identifier
length	0x2	0x2	0x4	On error this field is set to 4
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.15 SH_CMD_WRITE_RAM (0xB0) command

This command is used to write a block of data to the RAM. The RAM between 0x1000000 and 0x10007FF is reserved by the I2C/SPI ISP.

Table 90.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xB0	'Write RAM' command identifier
crcCheck	0x1	0x1		0 – CRC check not enabled 1 – CRC check enabled
size	0x2	0x2	length	Number of bytes to be written to the RAM. The size must be word aligned. The maximum size is 128 bytes.
addr	0x4	0x4		Beginning address to be written in RAM, must be >= 0x10000800 if CRP1 is enabled.
checksum	0x8	0x4	CRC32	CRC32 of the packet excluding this field. Set this field to 0 if crcCheck is set to 1.
data	0xC	length		Data string to be written to the RAM

Table 91.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xB0	Processed command identifier
length	0x2	0x2	0x0	Length of the response packet. 0x0 = success

Table 92.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xB0	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet. 0x4 = failure
errorCode	0x4	0x4	Error code	Error code specified in error.h parameters

5.8.16 SH_CMD_GOTO (0xB1) command

This command is used to jump to user code based on the address given in the command packet.

Table 93.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xB1	'GoTo' command identifier
address	0x4	0x4		Code address to jump to user code

If there is an error in the command, there is a response packet as shown in Table 94.

Table 94.   Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xB1	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet. 0x4 = failure
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.17 SH_CMD_FAIM_READ (0xBE) command

This command is used to read one page from the FAIM memory. One page of FAIM is 4 bytes.

Table 95.   Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xBE	'Read block' command identifier
reserved	0x1	0x1		zero
PageNum	0x2	0x2		FAIM page number to read

Table 96.   Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xBE	Processed command identifier
length	0x2	0x2	0x08	One page is 4 bytes. Length of the response packet. 0x8 = success
data	0x4	0x4		FAIM page content
checksum	0x8	0x4	CRC32	CRC32 of the packet excluding this field.

Table 97. Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xBE	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet. 0x4 = failure
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.18 SH_CMD_FAIM_WRITE (0xBF) command

This command is used to write one page to the FAIM memory. One page of FAIM is 4 bytes.

Table 98. Command packet

Field	Offset	Size (bytes)	Value	Description
command	0x0	0x1	0xBF	'FAIM write' command identifier
crcCheck	0x1	0x1		00 – Do CRC check for this packet 01 – Ignore CRC field for this packet
PageNum	0x2	0x2		Page number to program data
data	0x4	0x4		Data to be programmed in FAIM
checksum	0x8	0x4	CRC32	CRC32 of the packet excluding this field. Set this field to 0 if crcCheck is set to 1.

Table 99. Response packet (success)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xBF	Processed command identifier
length	0x2	0x2	0x00	On success this field is set to 0

Table 100. Response packet (error)

Field	Offset	Size (bytes)	Value	Description
sop	0x0	0x1	0x55	Start of packet identifier
command	0x1	0x1	0xBF	Processed command identifier
length	0x2	0x2	0x4	Length of the response packet. 0x4 = failure
errorCode	0x4	0x4	Error code	Error code specified in error.h

5.8.19 I2C/SPI ISP Error codes

See Table 36 "ISP/IAP Error codes".

5.8.20 I2C/SPI ISP mode protocol software support

This section includes predefined command definitions and software structures used for communication for the I2C/SPI ISP protocol. These may be used with a host application that talks to the I2C/SPI ISP interface via the supported I2C or SPI slave interfaces.

```
/* I2C/SPI IMSP mode protocol commands */
#define SH_CMD_GET_VERSION 0xA1
#define SH_CMD_RESET     0xA2
#define SH_CMD_BOOT      0xA3
```

UM11029
© NXP B.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **58 of 530**

```
#define SH_CMD_CHECK_IMAGE 0xA4
#define SH_CMD_PROBE  0xA5
#define SH_CMD_WRITE_BLOCK 0xA6
#define SH_CMD_READ_BLOCK 0xA7
#define SH_CMD_SECTOR_ERASE 0xA8
#define SH_CMD_PAGE_ERASE 0xA9
#define SH_CMD_PAGE_WRITE 0xAA
#define SH_CMD_PAGE_READ 0xAB
#define SH_CMD_WRITE_SUBBLOCK 0xAC
#define SH_CMD_READ_SUBBLOCK 0xAD
#define SH_CMD_BULK_ERASE 0xAE
#define SH_CMD_WRITE_RAM 0xB0
#define SH_CMD_GOTO 0xB1
#define SH_CMD_FAIM_READ 0xBE
#define SH_CMD_FAIM_WRITE 0xBF
/** Structure describing response packet format. */
typedef struct {
    uint8_t sop;     /*!< Start of packet = 0x55 for bootloader */
    uint8_t cmd;     /*!< Response to the Command ID. */
    uint16_t length; /*!< Response data length not including this header. */
} CmdResponse_t;

/** Structure describing Read/Write block command packet format. */
typedef struct {
    uint8_t cmd;       /*!< Command ID */
    uint8_t crc_check; /*!< specifies if we need to do CRC check before processing */
    uint16_t block_nr; /*!< Block number.*/
    uint32_t data[SL_FLASH_BLOCK_SZ/4];    /*!< Data */
    uint32_t crc32;    /*!< CRC32 of command header and data */
} CmdRWBlockParam_t;
    .            .      .    .     .     .      .
/** Structure describing Read/Write page command packet format. */
typedef struct {
    uint8_t cmd;       /*!< Command ID */
    uint8_t crc_check; /*!< specifies if we need to do CRC check before processing */
    uint16_t page_nr;  /*!< page number.*/
    uint32_t data[SL_FLASH_PAGE_SZ/4];     /*!< Data */
    uint32_t crc32;    /*!< CRC32 of command header and data */
} CmdRWPageParam_t;


/** Structure describing Read sub-block command packet format. */
typedef struct {
    uint8_t cmd;            /*!< Command ID */
    uint8_t sub_block_nr;   /*!< specifies the sub-block number. Bits below:
                            0 - Skip crc;
                            5-1: sub block nr;
                            7-6: sub-block size. 0 - 32, 1 - 64, 2 - 128, 3 - 256 */
    uint16_t block_nr;      /*!< block number.*/
    uint32_t data[SL_FLASH_BLOCK_SZ/4];    /*!< Data */
} CmdReadSubBlockParam_t;

/** Structure describing Sector erase command packet format. */
```

```
typedef struct {
    uint8_t cmd;       /*!< Command ID */
    uint8_t reserved; /*!< Should be zero. */
    uint16_t sec_nr;  /*!< Sector number.*/
} CmdEraseSectorParam_t;

/** Structure describing Bulk erase command packet format. */
typedef struct {
    uint8_t cmd;            /*!< Command ID */
    uint8_t reserved;       /*!< Should be zero. */
    uint8_t start_sec;      /*!< Start Sector number.*/
    uint8_t end_sec;        /*!< End Sector number.*/
} CmdBulkEraseParam_t;

/** Structure describing response packet with data. */
typedef struct {
    CmdResponse_t hdr;                   /*!< Response header. */
    uint32_t data[SL_FLASH_BLOCK_SZ/4]; /*!< Data */
    uint32_t crc32;                      /*!< CRC32 of response packet. */
} CmdDataResp_t;
typedef enum
{
```

6.1 How to read this chapter

The flash signature generator is identical on all LPC84x parts.

6.2 Features

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

6.3 General description

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

6.4 Register description

Table 101. Register overview: FMC (base address 0x4004 0000)

Name	Access	Address offset	Description	Reset value	Reference
FLASHCFG	R/W	0x010	Flash configuration register	-	Section 6.4.1
FMSSTART	R/W	0x020	Signature start address register	0	Section 6.4.2
FMSSTOP	R/W	0x024	Signature stop-address register	0	Section 6.4.3
FMSW0	R	0x02C	Signature word	-	Section 6.4.4
FMSTAT	R	0xFE0	Signature generation status register	0	Section 6.4.5
FMSTATCLR	W	0xFE8	Signature generation status clear register.	-	Section 6.4.6

6.4.1 Flash configuration register

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

Table 102. 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.	
		0x1	2 system clocks flash access time.	
		0x2	3 system clocks flash access time.	
		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**.	-

6.4.2 Flash signature start address register

Table 103. 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

6.4.3 Flash signature stop address register

Table 104. 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

6.4.4 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 6.5.1.

Table 105. FMSW0 register bit description (FMSW0, address: 0x4004 002C)

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

6.4.5 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **62 of 530**

Table 106. Flash module signature status register (FMSTAT, offset 0x0FE0) bit description

Bit	Symbol	Description	Reset value
1:0	-	Reserved. Read value is undefined, only zero should be written.	NA
2	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

6.4.6 Flash module signature status clear register

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

Table 107. Flash module signature status clear register (FMSTATCLR, offset 0x0FE8) bit description

Bit	Symbol	Description	Reset value
1:0	-	Reserved. Read value is undefined, only zero should be written.	NA
2	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

6.5 Functional description

6.5.1 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.

6.5.1.1 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 103 and Table 104 show the bit assignments in the FMSSTART and FMSSTOP registers respectively.

6.5.1.2 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.

6.5.1.3 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
```

7.1 How to read this chapter

The NVIC is identical on all LPC84x parts.

7.2 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.

7.3 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.

7.3.1 Interrupt sources

Table 108 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 108. Connection of interrupt sources to the NVIC

Interrupt number	Name	Description	Flags
0	SPI0_IRQ	SPI0 interrupt	See Table 342 "SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI0), 0x4005 C00C (SPI1)) bit description".
1	SPI1_IRQ	SPI1 interrupt	Same as SPI0_IRQ
2	DAC0_IRQ	DAC0 interrupt	-

UM11029
© NXP Semiconductors N.V. 2017. All rights reserved.

User manual   **Rev. 1.6 — 8 December 2017**   **66 of 530**

Table 108. Connection of interrupt sources to the NVIC

Interrupt number	Name	Description	Flags
3	UART0_IRQ	USART0 interrupt	See Table 328 "USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1), 0x4006C00C (USART2), 0x4007 000C (USART3), 0x4007 400C (USART4)) bit description"
4	UART1_IRQ	USART1 interrupt	Same as UART0_IRQ
5	UART2_IRQ	USART2 interrupt	Same as UART0_IRQ
6	-	Reserved	-
7	I2C1_IRQ	I2C1 interrupt	See Table 358 "Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0), 0x4005 400C (I2C1), 0x4003 000C (I2C2), 0x4003 400C (I2C3)) bit description".
8	I2C0_IRQ	I2C0 interrupt	See Table 358 "Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0), 0x4005 400C (I2C1), 0x4003 000C (I2C2), 0x4003 400C (I2C3)) bit description".
9	SCT_IRQ	State configurable timer interrupt	EVFLAG SCT event
10	MRT_IRQ	Multi-rate timer interrupt	Global MRT interrupt. GFLAG0 GFLAG1 GFLAG2 GFLAG3
11	CMP_IRQ or CAPT_IRQ	Analog comparator interrupt or Capacitive Touch interrupt	COMPEDGE - rising, falling, or both edges can set the bit. Capacitive Touch interrupt.
12	WDT_IRQ	Windowed watchdog timer interrupt	WARNINT - watchdog warning interrupt
13	BOD_IRQ	BOD interrupts	BODINTVAL - BOD interrupt level
14	FLASH_IRQ	flash interrupt	-
15	WKT_IRQ	Self-wake-up timer interrupt	ALARMFLAG
16	ADC_SEQA_IRQ	ADC sequence A completion	-
17	ADC_SEQB_IRQ	ADC sequence B completion	-
18	ADC_THCMP_IRQ	ADC threshold compare	-
19	ADC_OVR_IRQ	ADC overrun	-
20	DMA_IRQ	DMA interrupt	-
21	I2C2_IRQ	I2C2 interrupt	See Table 358 "Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0), 0x4005 400C (I2C1), 0x4003 000C (I2C2), 0x4003 400C (I2C3)) bit description".
22	I2C3_IRQ	I2C3 interrupt	See Table 358 "Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0), 0x4005 400C (I2C1), 0x4003 000C (I2C2), 0x4003 400C (I2C3)) bit description".

Table 108. Connection of interrupt sources to the NVIC

Interrupt number	Name	Description	Flags
23	CT32B0_IRQ	Timer interrupt	-
24	PININT0_IRQ	Pin interrupt 0 or pattern match engine slice 0 interrupt	PSTAT - pin interrupt status
25	PININT1_IRQ	Pin interrupt 1 or pattern match engine slice 1 interrupt	PSTAT - pin interrupt status
26	PININT2_IRQ	Pin interrupt 2 or pattern match engine slice 2 interrupt	PSTAT - pin interrupt status
27	PININT3_IRQ	Pin interrupt 3 or pattern match engine slice 3 interrupt	PSTAT - pin interrupt status
28	PININT4_IRQ	Pin interrupt 4 or pattern match engine slice 4 interrupt	PSTAT - pin interrupt status
29	PININT5_IRQ or DAC1_IRQ	Pin interrupt 5 or pattern match engine slice 5 interrupt or DAC1 interrupt	PSTAT - pin interrupt status
30	PININT6_IRQ r USART3_IRQ	Pin interrupt 6 or pattern match engine slice 6 interrupt or USART3 interrupt	PSTAT - pin interrupt status
31	PININT7_IRQ or USART4_IRQ	Pin interrupt 7 or pattern match engine slice 7 interrupt or USART4 interrupt	PSTAT - pin interrupt status

7.3.2 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 108 as NMI using the NMISRC register in the SYSCON block (Table 167). 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.

7.3.3 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).

7.4 Register description

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

Table 109. 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 110
-	-	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 111
-	-	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 112
-	-	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 113
-	-	0x284	Reserved.	0	-
IABR0	RO	0x300	Interrupt Active Bit Register 0. This register allows reading the current interrupt active state for specific peripheral functions.	0	Table 114
-	-	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 115
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 116
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 117
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 118
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 119
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 120
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 121
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 122

UM11029

User manual **Rev. 1.6 — 8 December 2017** **69 of 530**

7.4.1 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 7.4.2).

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 110. 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	ISE_DAC0	Interrupt enable.	0
3	ISE_UART0	Interrupt enable.	0
4	ISE_UART1	Interrupt enable.	0
5	ISE_UART2	Interrupt enable.	0
6	-	Reserved	0
7	ISE_I2C1	Interrupt enable.	0
8	ISE_I2C0	Interrupt enable.	0
9	ISE_SCT	Interrupt enable.	0
10	ISE_MRT	Interrupt enable.	0
11	ISE_CMP or ISE_CAPT	Interrupt enable for both comparator and Capacitive Touch.	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_SDMA	Interrupt enable.	0
21	ISE_I2C2	Interrupt enable.	0
22	ISE_I2C3	Interrupt enable.	0
23	ISE_CT32b0	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 or ISE_DAC1	Interrupt enable for both pinint5 and DAC1.	0
30	ISE_PININT6 or ISE_USART3	Interrupt enable for both pinint6 and USART3.	0
31	ISE_PININT7or ISE_USART4	Interrupt enable for both pinint7 and USART4.	0

7.4.2 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 7.4.1).

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 111. 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	ICE_DAC0	Interrupt disable.	0
3	ICE_UART0	Interrupt disable.	0
4	ICE_UART1	Interrupt disable.	0
5	ICE_UART2	Interrupt disable.	0
6	-	Reserved.	0
7	ICE_I2C1	Interrupt disable.	0
8	ICE_I2C0	Interrupt disable.	0
9	ICE_SCT	Interrupt disable.	0
10	ICE_MRT	Interrupt disable.	0
11	ICE_CMP or ICE_CAPT	Interrupt disable for both comparator and Capacitive Touch.	ISE_CMP or ISE_CAPT
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_SDMA	Interrupt disable.	0
21	ICE_I2C2	Interrupt disable.	0
22	ICE_I2C3	Interrupt disable.	0
23	ICE_CT32B0	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

Table 111. Interrupt clear enable register 0 (ICER0, address 0xE000 E180) *…continued*

Bit	Symbol	Description	Reset value
29	ICE_PININT5 or ICE_DAC1	Interrupt disable for both pinint5 and DAC1.	0
30	ICE_PININT6 or ICE_USART3	Interrupt disable for both pinint6 and USART3.	0
31	ICE_PININT7 or ICE_USART4	Interrupt disable for both pinint7 and USART4.	0

7.4.3 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 7.4.4).

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 112. 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	ISP_DAC0	Interrupt pending set.	0
3	ISP_UART0	Interrupt pending set.	0
4	ISP_UART1	Interrupt pending set.	0
5	ICE_UART2	Interrupt pending set.	0
6	-	Reserved	0
7	ISP_I2C1	Interrupt pending set.	0
8	ISP_I2C0	Interrupt pending set.	0
9	ISP_SCT	Interrupt pending set.	0
10	ISP_MRT	Interrupt pending set.	0
11	ISP_CMP or ISP_CAPT	Interrupt pending set for both comparator and Capacitive Touch.	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_I2C2	Interrupt pending set.	0
22	ISP_I2C3	Interrupt pending set.	0

UM11029

User manual **Rev. 1.6 — 8 December 2017** **72 of 530**

Table 112. Interrupt set pending register 0 register (ISPR0, address 0xE000 E200) bit description *…continued*

Bit	Symbol	Description	Reset value
23	ISP_CT32B0	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 or ISP_DAC1	Interrupt pending set for both pinint5 and DAC1.	0
30	ISP_PININT6 or ISP_UART3	Interrupt pending set for both pinint6 and USART3.	0
31	ISP_PININT7 or ISP_USART4	Interrupt pending set for both pinint7 and USART4.	0

7.4.4 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 7.4.3).

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 113. 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	ICP_DAC0	Interrupt pending clear.	0
3	ICP_UART0	Interrupt pending clear.	0
4	ICP_UART1	Interrupt pending clear.	0
5	ICP_UART2	Interrupt pending clear.	0
6	-	Reserved	0
7	ICP_I2C1	Interrupt pending clear.	0
8	ICP_I2C0	Interrupt pending clear.	0
9	ICP_SCT	Interrupt pending clear.	0
10	ICP_MRT	Interrupt pending clear.	0
11	ICP_CMP or ICP_CAPT	Interrupt pending clear for both comparator and Capacitive Touch.	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

Table 113. Interrupt clear pending register 0 register (ICPR0, address 0xE000 E280) bit description *…continued*

Bit	Symbol	Function	Reset value
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_I2C2	Interrupt pending clear.	0
22	ICP_I2C3	Interrupt pending clear.	0
23	ICICP_CT32B0	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 or ICP_DAC1	Interrupt pending clear for both pinint5 and DAC1.	0
30	ICP_PININT6 or ICP_USART3	Interrupt pending clear for both pinint6 and USART3.	0
31	ICP_PININT7 or ICP_USART4	Interrupt pending clear for both pinint7 and USART4.	0

7.4.5 Interrupt Active Bit Register 0

The IABR0 register is a read-only register that allows reading the active state of the peripheral interrupts. Use this register to determine which peripherals are asserting an interrupt to the NVIC and may also be pending if there are enabled.

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

Write — n/a.

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

Table 114. Interrupt Active Bit Register 0 (IABR0, address 0xE000 E300) bit description

Bit	Symbol	Function	Reset value
0	IAB_SPI0	Interrupt active.	0
1	IAB_SPI1	Interrupt active.	0
2	IAB_DAC0	Interrupt active.	0
3	IAB_UART0	Interrupt active.	0
4	IAB_UART1	Interrupt active.	0
5	IAB_UART2	Interrupt active.	0
6	-	Reserved	0
7	IAB_I2C1	Interrupt active.	0
8	IAB_I2C0	Interrupt active.	0
9	IAB_SCT	Interrupt active.	0
10	IAB_MRT	Interrupt active.	0
11	IAB_CMP or IAB_CAPT	Interrupt active for both comparator and Capacitive Touch.	0
12	IAB_WDT	Interrupt active.	0
13	IAB_BOD	Interrupt active.	0

Table 114. Interrupt Active Bit Register 0 (IABR0, address 0xE000 E300) bit description *...continued*

Bit	Symbol	Function	Reset value
14	IAB_FLASH	Interrupt active.	0
15	IAB_WKT	Interrupt active.	0
16	IAB_ADC_SEQA	Interrupt active.	0
17	IAB_ADC_SEQB	Interrupt active.	0
18	IAB_ADC_THCMP	Interrupt active.	0
19	IAB_ADC_OVR	Interrupt active.	0
20	IAB_SDMA	Interrupt active.	0
21	IAB_I2C2	Interrupt active.	0
22	IAB_I2C3	Interrupt active.	0
23	IAB_CT32B0	Interrupt active.	0
24	IAB_PININT0	Interrupt active.	0
25	IAB_PININT1	Interrupt active.	0
26	IAB_PININT2	Interrupt active.	0
27	IAB_PININT3	Interrupt active.	0
28	IAB_PININT4	Interrupt active.	0
29	IAB_PININT5 or IAB_DAC1	Interrupt active for both pinint5 and DAC1.	0
30	IAB_PININT6 or IAB_USART3	Interrupt active for both pinint6 and USART3.	0
31	IAB_PININT7 or IAB_USART4	Interrupt active for both pinint7 and USART4.	0

7.4.6 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 115. 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	IP_DAC0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	Reserved.
31:30	IP_UART0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

7.4.7 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 116. 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.
29:22	-	Reserved.
31:30	IP_I2C1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

7.4.8 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 117. 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	IP_SCT	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
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_CMP or IP_CAPT	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

7.4.9 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 118. 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.

7.4.10 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 119. 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.

7.4.11 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 120. 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.
15:14	IP_I2C2	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_I2C3	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	Reserved.
31:30	IP_CT32B0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

7.4.12 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 121. 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.

7.4.13 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 122. 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 or IP_DAC1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
21:16	-	These bits ignore writes, and read as 0.
23:22	IP_PININT6 or IP_USART3	Interrupt Priority. 0 = highest priority. 3 = lowest priority.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_PININT7 or IP_USART4	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

8.1 How to read this chapter

The system configuration block is identical for all LPC84x parts.

8.2 Features

- Clock control
  - Configure the system PLL.
  - Configure system oscillator, watchdog 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 Capacitive Touch, ADC, SCT 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.
- MTB trace start and stop.
- 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, $\overline{RESET}$, 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 18 MHz, 24 MHz, and 30 MHz outputs that can be used as a system clock. Also, the fro_oscout can be divided down to provide frequencies of 1.125 MHz, 1.5 MHz, 1.875 MHz, 9 MHz, 12 MHz, or 15 MHz for system clock.

By default, the fro_oscout is 24 MHz and is divided by 2 to provide a default system (CPU) clock frequency of 12 MHz.

---

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

   Section 8.6.48 "Power configuration register"

2. Select the fro_oscout (30 MHz/24 MHz/18 MHz) using the set_fro_frequency API call:

   Chapter 9 "LPC84x FRO API ROM routine" and Figure 9 "LPC84x clock generation (continued)".

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"

5. Based on the Low Power boot bit in the FAIM, default divide by 2 is replaced with divide by 16.

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: 12 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 8.6.48 "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: 12 MHz internal oscillator (default).

UM11029

User manual **Rev. 1.6 — 8 December 2017** **80 of 530**

 – External clock input: It can be external crystal oscillator using the XTALIN/XTALOUT pins or CLKIN from external pin.

 – Watchdog oscillator.

 – FRO DIV: 6 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 8.6.22 "System clock control 0 register"

 Section 8.6.23 "System clock control 1 register"

8.3.4 Set up the system oscillator using XTALIN and XTALOUT

To use the system oscillator with the LPC84x, 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 214 "PIO0_8 register (PIO0_8, address 0x4004 4038) bit description"

Table 213 "PIO0_9 register (PIO0_9, address 0x4004 4034) bit description"

Table 196 "Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description"

Table 129 "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 10.3.1 "Connect an internal signal to a package pin" to assign the CLKOUT function to a pin.

See Section 10.3.2 to enable the clock input, the oscillator pins, and the external reset input.

Table 123. SYSCON pin description

Function	Direction	Pin	Description	SWM register	Reference
CLKOUT	O	any	CLKOUT clock output.	PINASSIGN8	Table 189
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 196
XTALIN	I	PIO0_8/XTALIN	Input to the system oscillator.	PINENABLE0	Table 196
XTALOUT	O	PIO0_9/XTALOUT	Output from the system oscillator.	PINENABLE0	Table 196
RESET	I	RESET/PIO0_5	External reset input	PINENABLE0	Table 196

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^2C clock, CAPT clock, SCTimer/PWM 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 30 MHz. See Figure 7.

Remark: The main clock frequency is limited to 100 MHz.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **83 of 530**
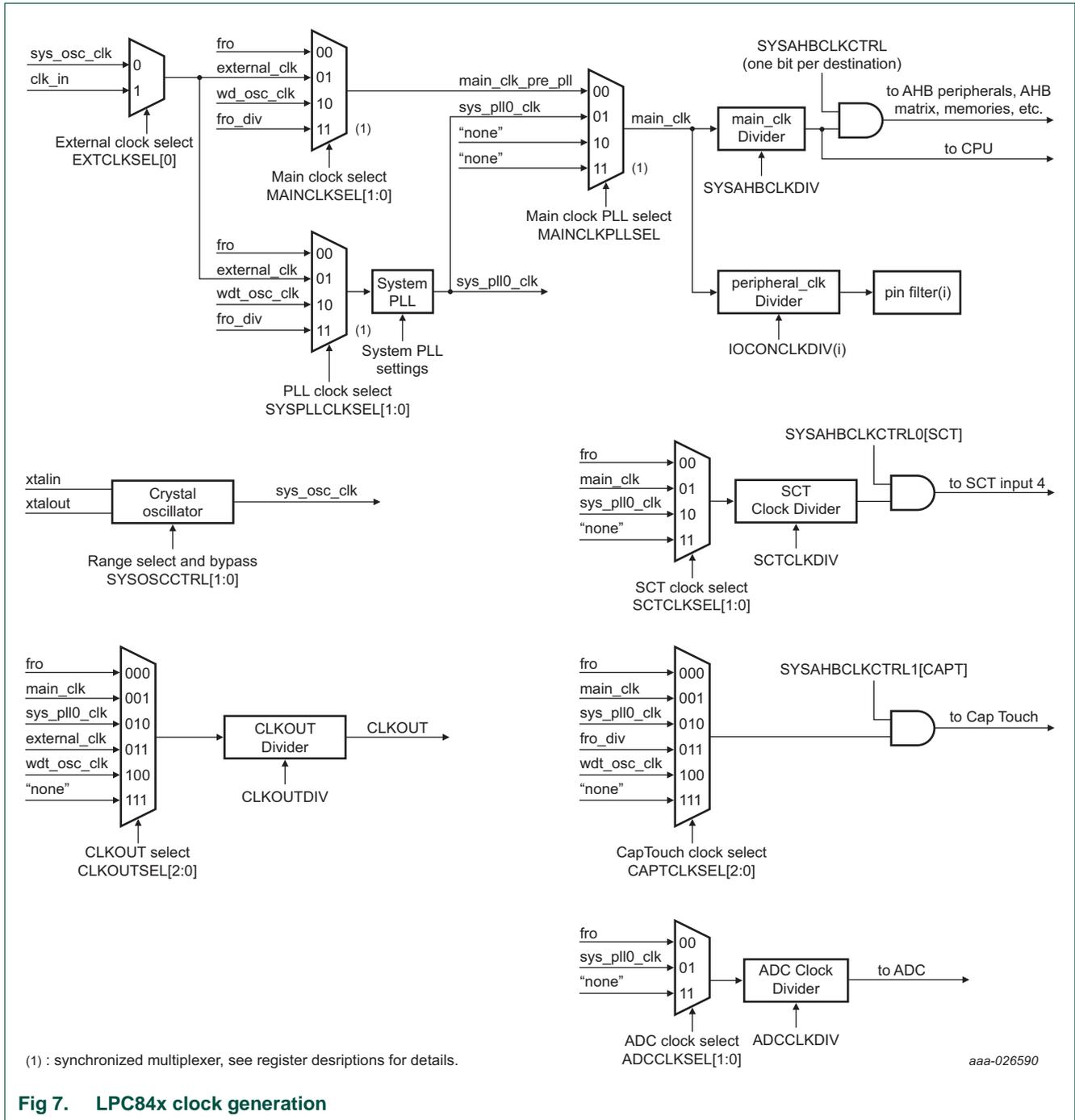
Fig 7.   LPC84x clock generation
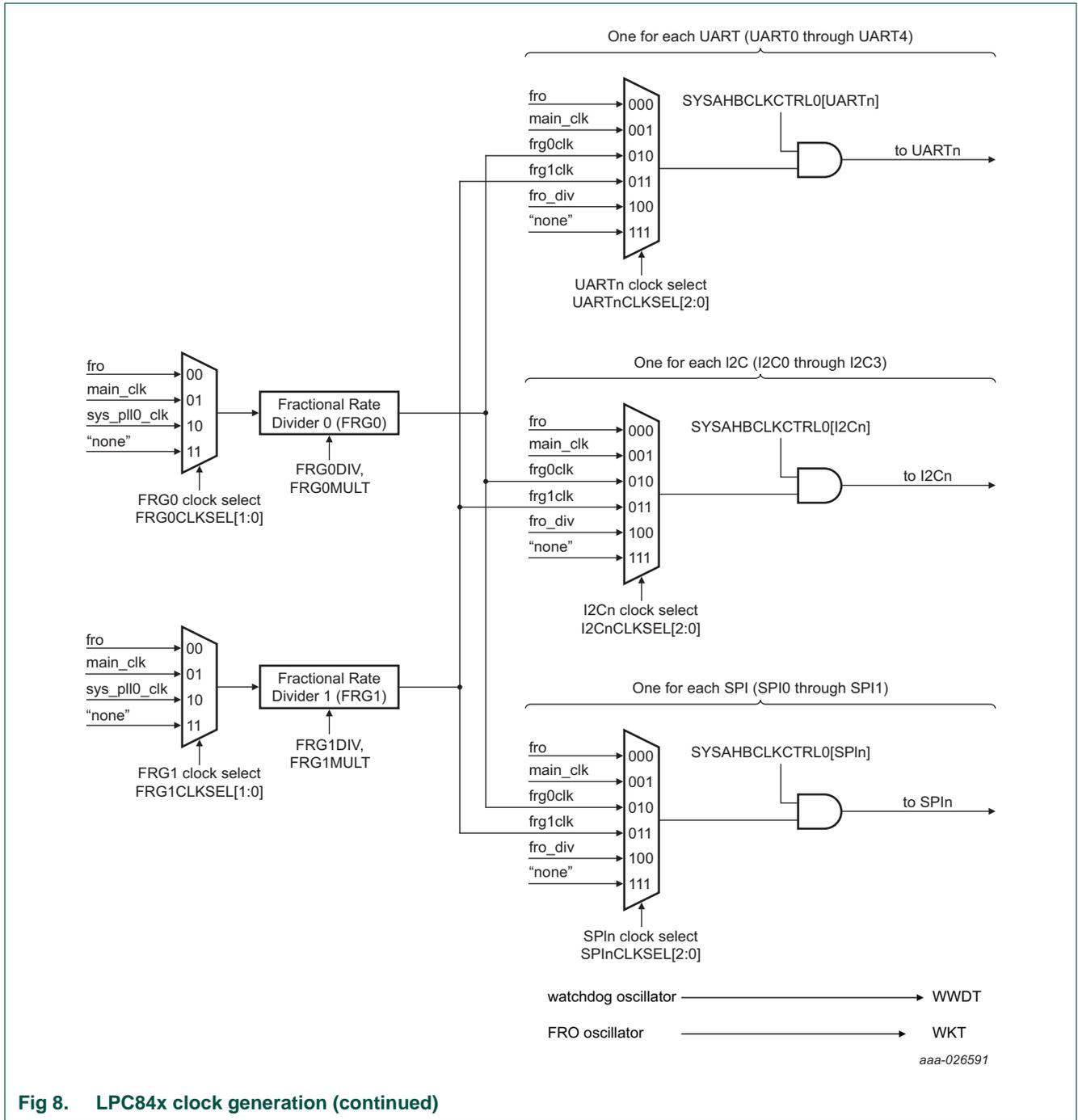
Fig 8.   LPC84x clock generation (continued)
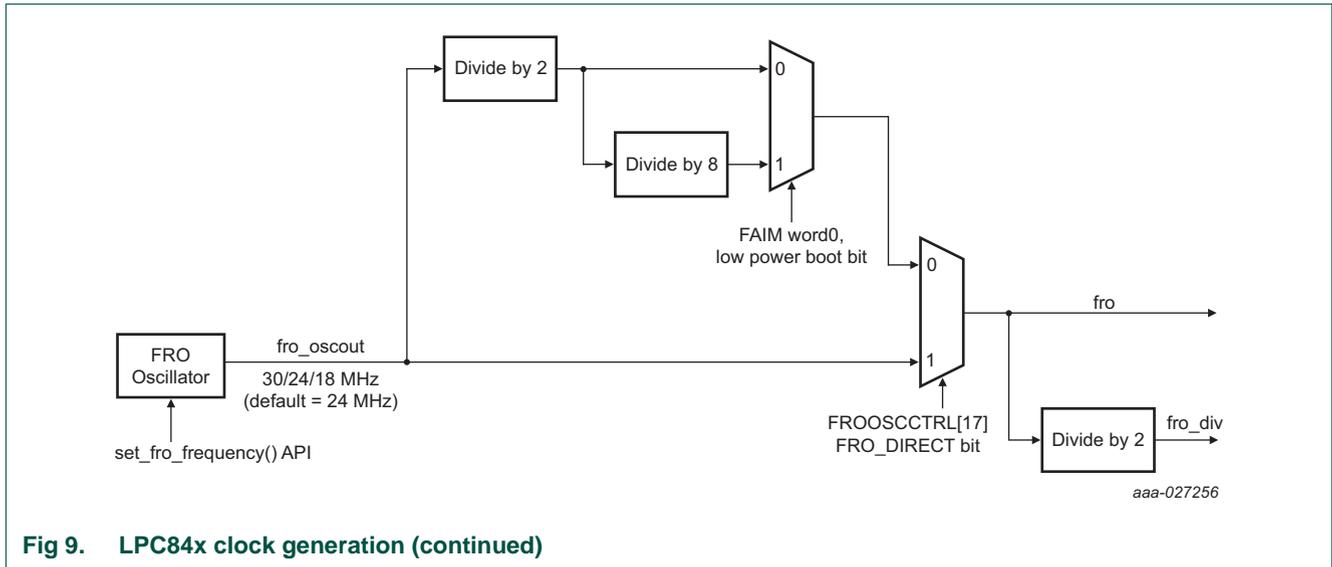
Fig 9.   LPC84x clock generation (continued)

Table 124. 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 "LPC84x clock generation (continued)".
fro	The output of the currently selected on-chip FRO oscillator. See Figure 9 "LPC84x clock generation (continued)".
fro_div	The FRO output. This may be either 15 MH, 12 MHz, or 9 MHz. See Figure 9 "LPC84x clock generation (continued)".
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 "LPC84x clock generation".
"none"	A tied-off source that should be selected to save power when the output of the related multiplexer is not used.
sys_pll0_clk	The output of the System PLL. The System PLL and its source selection are shown in Figure 7 "LPC84x clock generation".
wdt_osc_clk	The output of the watchdog oscillator, which has a selectable target frequency. It must also be enabled in the PDRINCFG0 register. See UM11029 User manual.
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.

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 8.6.46 "Deep-sleep mode configuration register"

Section 8.6.2 "System PLL control register"

Section 8.6.5 "Watchdog oscillator 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 watchdog 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 8.6.48 "Power configuration register"

Section 8.6.45 "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 8.6.43 "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 125 as reserved should not be written to.

Table 125. Register overview: System configuration (base address 0x4004 8000)

Name	Access	Offset	Description	Reset value	Section
SYSMEMREMAP	R/W	0x000	System memory remap	0x2	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
WDTOSCCTRL	R/W	0x024	Watchdog oscillator control	0x000	8.6.5
FROOSCCTRL	R/W	0x028	FRO oscillator control	0x8801	8.6.6
-	-	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	0	8.6.8
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
-	-	0x05C	Reserved	-	-
CAPTCLKSEL	R/W	0x060	CAPT clock source select	0x7	8.6.16
ADCCLKSEL	R/W	0x064	ADC clock source select	0	8.6.17
ADCCLKDIV	R/W	0x068	ADC clock divider	0	8.6.18
SCTCLKSEL	R/W	0x06C	SCT clock source select	0	8.6.19
SCTCLKDIV	R/W	0x070	SCT clock divider	0	8.6.20
EXTCLKSEL	R/W	0x074	External clock source select	0	8.6.21
-	-	0x078	Reserved	-	-
-	-	0x07C	Reserved	-	-
SYSAHBCLKCTRL0	R/W	0x080	System clock control 0	0x17	8.6.22
SYSAHBCLKCTRL1	R/W	0x084	System clock control 1	0x0	8.6.23
PRESETCTRL0	R/W	0x088	Peripheral reset control 0	0xFFFFFFFF	8.6.24

UM11029

User manual **Rev. 1.6 — 8 December 2017** **88 of 530**

Table 125. Register overview: System configuration (base address 0x4004 8000) *...continued*

Name	Access	Offset	Description	Reset value	Section
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
UART3CLKSEL	R/W	0x09C	Function clock source select for UART3	0x7	8.6.26
UART4CLKSEL	R/W	0x0A0	Function clock source select for UART4	0x7	8.6.26
I2C0CLKSEL	R/W	0x0A4	Function clock source select for I^{2}C0	0x7	8.6.26
I2C1CLKSEL	R/W	0x0A8	Function clock source select for I^{2}C1	0x7	8.6.26
I2C2CLKSEL	R/W	0x0AC	Function clock source select for I^{2}C2	0x7	8.6.26
I2C3CLKSEL	R/W	0x0B0	Function clock source select for I^{2}C3	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
-	-	0x0BC	Reserved	-	-
-	-	0x0C0	Reserved	-	-
-	-	0x0C4	Reserved	-	-
-	-	0x0C8	Reserved	-	-
-	-	0x0CC	Reserved	-	-
FRG0DIV	R/W	0x0D0	Fractional generator divider value	0x0	8.6.27
FRG0MULT	R/W	0x0D4	Fractional generator multiplier value	0x0	8.6.28
FRG0CLKSEL	R/W	0x0D8	FRG0 clock source select	0	8.6.29
-	-	0x0DC	Reserved	-	-
FRG1DIV	R/W	0x0E0	Fractional generator divider value	0x0	8.6.30
FRG1MULT	R/W	0x0E4	Fractional generator multiplier value	0x0	8.6.31
FRG1CLKSEL	R/W	0x0E8	FRG1 clock source select	0	8.6.32
-	-	0x0EC	Reserved	-	
CLKOUTSEL	R/W	0x0F0	CLKOUT clock source select	0	8.6.33
CLKOUTDIV	R/W	0x0F4	CLKOUT clock divider	0	8.6.34
-	-	0x0F8	Reserved	-	-
EXTTRACECMD	R/W	0x0FC	External trace buffer command register	0	8.6.35
PIOPORCAP0	R	0x100	POR captured PIO0 status 0	user dependent	8.6.36
PIOPORCAP1	R	0x104	POR captured PIO1 status 0	user dependent	8.6.37
-	-	0x108 - 0x130	Reserved	-	-
IOCONCLKDIV6	R/W	0x134	Peripheral clock 6 to the IOCON block for programmable glitch filter	0	8.6.38
IOCONCLKDIV5	R/W	0x138	Peripheral clock 5 to the IOCON block for programmable glitch filter	0	8.6.38
IOCONCLKDIV4	R/W	0x13C	Peripheral clock 4 to the IOCON block for programmable glitch filter	0	8.6.38
IOCONCLKDIV3	R/W	0x140	Peripheral clock 3 to the IOCON block for programmable glitch filter	0	8.6.38

Table 125. Register overview: System configuration (base address 0x4004 8000) *…continued*

Name	Access	Offset	Description	Reset value	Section
IOCONCLKDIV2	R/W	0x144	Peripheral clock 2 to the IOCON block for programmable glitch filter	0	8.6.38
IOCONCLKDIV1	R/W	0x148	Peripheral clock 1 to the IOCON block for programmable glitch filter	0	8.6.38
IOCONCLKDIV0	R/W	0x14C	Peripheral clock 0 to the IOCON block for programmable glitch filter	0	8.6.38
BODCTRL	R/W	0x150	Brown-Out Detect	0	8.6.39
SYSTCKCAL	R/W	0x154	System tick counter calibration	0	8.6.40
-	R/W	0x158 - 0x16C	Reserved	-	-
IRQLATENCY	R/W	0x170	IRQ delay. Allows trade-off between interrupt latency and determinism.	0x0000 0010	8.6.41
NMISRC	R/W	0x174	NMI Source Control	0	8.6.42
PINTSEL0	R/W	0x178	GPIO Pin Interrupt Select register 0	0	8.6.43
PINTSEL1	R/W	0x17C	GPIO Pin Interrupt Select register 1	0	8.6.43
PINTSEL2	R/W	0x180	GPIO Pin Interrupt Select register 2	0	8.6.43
PINTSEL3	R/W	0x184	GPIO Pin Interrupt Select register 3	0	8.6.43
PINTSEL4	R/W	0x188	GPIO Pin Interrupt Select register 4	0	8.6.43
PINTSEL5	R/W	0x18C	GPIO Pin Interrupt Select register 5	0	8.6.43
PINTSEL6	R/W	0x190	GPIO Pin Interrupt Select register 6	0	8.6.43
PINTSEL7	R/W	0x194	GPIO Pin Interrupt Select register 7	0	8.6.43
-	-	0x198 - 0x200	Reserved	-	-
STARTERP0	R/W	0x204	Start logic 0 pin wake-up enable register	0	8.6.44
-	-	0x208 - 0x210	Reserved	-	-
STARTERP1	R/W	0x214	Start logic 1 interrupt wake-up enable register	0	8.6.45
-	-	0x218 - 0x22C	Reserved	-	-
PDSLEEPCFG	R/W	0x230	Power-down states in deep-sleep mode	0xFFFF	8.6.46
PDAWAKECFG	R/W	0x234	Power-down states for wake-up from deep-sleep	0xEDF8	8.6.47
PDRUNCFG	R/W	0x238	Power configuration register	0xEDF8	8.6.48
-	-	0x23C - 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 126. 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.	0x2
		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 127. 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 $2 \times P$.	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).

UM11029

User manual **Rev. 1.6 — 8 December 2017** **91 of 530**

Table 128. 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 173.

Table 129. 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 Watchdog oscillator control register

This register configures the watchdog 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 watchdog oscillator can be calculated as wdt_osc_clk = Fclkana/(2 × (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 watchdog 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 watchdog oscillator is undefined after reset. The watchdog oscillator frequency must be programmed by writing to the WDTOSCCTRL register before using the watchdog oscillator.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **92 of 530**

Table 130. Watchdog oscillator control register (WDTOSCCTRL, address 0x4004 8024) bit description

Bit	Symbol	Value	Description	Reset value
4:0	DIVSEL		Select divider for Fclkana. wdt_osc_clk = Fclkana/ $(2 \times (1 + DIVSEL))$ 00000: $2 \times (1 + DIVSEL) = 2$ 00001: $2 \times (1 + DIVSEL) = 4$ to 11111: $2 \times (1 + DIVSEL) = 64$	0
8:5	FREQSEL		Select watchdog oscillator analog output frequency (Fclkana).	0x00
		0x0	0 MHz	
		0x1	0.6 MHz	
		0x2	1.05 MHz	
		0x3	1.4 MHz	
		0x4	1.75 MHz	
		0x5	2.1 MHz	
		0x6	2.4 MHz	
		0x7	2.7 MHz	
		0x8	3.0 MHz	
		0x9	3.25 MHz	
		0xA	3.5 MHz	
		0xB	3.75 MHz	
		0xC	4.0 MHz	
		0xD	4.2 MHz	
		0xE	4.4 MHz	
		0xF	4.6 MHz	
31:9	-	-	Reserved	0x00

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **93 of 530**

8.6.6 FRO oscillator control register

The FROOSCCTRL register can be used to select direct fro_oscout (30 MHz, 24 MHz, 18 MHz) or select divided fro_oscout (1.125 MHz, 1.5 MHz, 1.875 MHz, 9 MHz, 12 MHz, or 15 MHz) based on FAIM low power boot value.

The set_fro_frequency API call (Chapter 9 "LPC84x FRO API ROM routine") must be used to select desired output frequency from FRO. See Figure 9 "LPC84x clock generation (continued)".

Table 131. 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) or 16 (low power boot), depending on FAIM low power boot value. See Section 4.2.1 "FAIM bit definitions".	
		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 132. 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	-

UM11029

User manual **Rev. 1.6 — 8 December 2017** **94 of 530**

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 133 applies to the POR reset.

Table 133. System reset status register (SYSRSTSTAT, address 0x4004 8038) bit description

Bit	Symbol	Value	Description	Reset value
0	POR		POR reset status	0
		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.	0
		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 134. 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	Watchdog 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 135. 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 136. 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 137. 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, watchdog 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 138. 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	Watchdog 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 139. 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 140. 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 Capacitive Touch clock source select register

The CAPTCLKSEL register selects the CAPT clock, which can be the FRO, main clock, sys_pll, FRO_DIV, or watchdog oscillator.

Table 141. CAPT clock source select register (CAPTCLKSEL, address 0x4004 8060) bit description

Bit	Symbol	Value	Description	Reset value
2:0	SEL		Clock source for CAPT clock.	0x7
		0x0	FRO	
		0x1	main clock	
		0x2	SYS PLL	
		0x3	FRO_DIV = FRO/2	
		0x4	Watchdog oscillator	
		0x5	None	
		0x6	None	
		0x7	None	
31:3	-		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 142. 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 143. ADC clock divider register (ADCCLKDIV, address 0x4004 8068) bit description

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

8.6.19 SCT clock source select register

The SCTCLKSEL register selects the SCT clock, which can be the FRO, main clock or sys_pll.

Table 144. SCT clock source select register (SCTCLKSEL, address 0x4004 806C) bit description

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

8.6.20 SCT clock divider register

The SCTCLKDIV register controls how the SCT clock is divided to provide the SCT clock to the SCT module. The SCT clock can be shut down completely by setting the DIV field to zero.

Table 145. SCT clock divider register (SCTCLKDIV, address 0x4004 8070) bit description

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

8.6.21 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 146. 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	-

8.6.22 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 147. 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	RAM0_1		Enables clock for SRAM0 and SRAM1.	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.	1
		0	Disable	
		1	Enable	
8	SCT		Enables clock for state configurable timer SCTimer/PWM.	0
		0	Disable	
		1	Enable	
9	WKT		Enables clock for self-wake-up timer.	0
		0	Disable	
		1	Enable	
10	MRT		Enables clock for multi-rate timer.	
		0	Disable	
		1	Enable	
11	SPI0		Enables clock for SPI0.	0
		0	Disable	
		1	Enable	
12	SPI1		Enables clock for SPI1.	
		0	Disable	
		1	Enable	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **100 of 530**

Table 147. System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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	I2C1		Enables clock to I2C1.	0
		0	Disable	
		1	Enable	
22	I2C2		Enables clock to I2C2.	0
		0	Disable	
		1	Enable	
23	I2C3		Enables clock to I2C3.	0
		0	Disable	
		1	Enable	
24	ADC		Enables clock to ADC.	0
		0	Disable	
		1	Enable	
25	CTIMER0	-	Enables clock for CTIMER0	0
		0	Disable	
		1	Enable	

Table 147. System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description …continued

Bit	Symbol	Value	Description	Reset value
26	MTB		Enables clock to micro-trace buffer control registers. Turn on this clock when using the micro-trace buffer for debug purposes.	0
		0	Disable	
		1	Enable	
27	DAC0	-	Enable clock for DAC0	0
		0	Disable	
		1	Enable	
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	
30	UART3		Enable clock for UART3	0
		0	Disable	
		1	Enable	
31	UART4		Enable clock for UART4	0
		0	Disable	
		1	Enable	

8.6.23 System clock control 1 register

The SYSAHBCLKCTRL1 register enables the clocks to peripheral blocks.

Table 148. System clock control 1 register (SYSAHBCLKCTRL1, address 0x4004 8084) bit description

Bit	Symbol	Value	Description	Reset value
0	CAPT		Enables the clock for CAPT.	0
		0	Disable	
		1	Enable	
1	DAC1		Enables clock for DAC1.	0
		0	Disable	
		1	Enable	
31:2	-		Reserved	-

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.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **102 of 530**

Table 149. 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	SCT_RST_N		SCT reset control	1
		0	Assert the SCT reset.	
		1	Clear the SCT reset.	
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.	
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.	

Table 149. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
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	I2C1_RST_N		I^{2}C1 reset control	1
		0	Assert the I^{2}C1 reset.	
		1	Clear the I^{2}C1 reset.	
22	I2C2_RST_N		I^{2}C2 reset control	1
		0	Assert the I^{2}C2 reset.	
		1	Clear the I^{2}C2 reset.	
23	I2C3_RST_N		I^{2}C3 reset control	1
		0	Assert the I^{2}C3 reset.	
		1	Clear the I^{2}C3 reset.	
24	ADC_RST_N		ADC reset control	1
		0	Assert the ADC reset.	
		1	Clear the ADC reset.	
25	CTIMER0_RST_N		CTIMER reset control	1
		0	Assert the CTIMER reset.	
		1	Clear the CTIMER reset.	
26	-	-	Reserved	1
27	DAC0_RST_N		DAC0 reset control	1
		0	Assert the DAC0 reset.	
		1	Clear the DAC0 reset.	
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.	
30	UART3_RST_N		UART3 reset control	1
		0	Assert the UART3 reset.	
		1	Clear the UART3 reset.	

Table 149. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
31	UART4_RST_N		UART4 reset control	1
		0	Assert the UART4 reset.	
		1	Clear the UART4 reset.	

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

Bit	Symbol	Value	Description	Reset value
0	CAPT_RST_N		Capacitive Touch reset control.	1
		0	Assert the Capacitive Touch reset.	
		1	Clear the Capacitive Touch reset.	
1	DAC1_RST_N		DAC1 reset control	1
		0	Assert the DAC1 reset.	
		1	Clear the DAC1 reset.	
2	-		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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **105 of 530**

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 151.

- 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).
- UART3 clock source select register (UART3CLKSEL, address 0x4004 809C).
- UART4 clock source select register (UART4CLKSEL, address 0x4004 80A0).
- I^{2}C0 clock source select register (I2C0CLKSEL, address 0x4004 80A4).
- I^{2}C1 clock source select register (I2C1CLKSEL, address 0x4004 80A8).
- I^{2}C2 clock source select register (I2C2CLKSEL, address 0x4004 80AC).
- I^{2}C3 clock source select register (I2C3CLKSEL, address 0x4004 80AC).
- SPI0 clock source select register (SPI0CLKSEL, address 0x4004 80B4).
- SPI1 clock source select register (SPI1CLKSEL, address 0x4004 80B8).

Table 151. 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 Fractional generator 0 divider value register

The UART, I^{2}C, 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **106 of 530**

- The MULT value of the fractional divider is programmed in the FRG0MULT register. See Table 153.

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 17.3.1 "Configure the USART clock and baud rate".

Section 17.7.1 "Clocking and baud rates".

Table 152. 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	-

8.6.28 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 152.
- 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 17.3.1 "Configure the USART clock and baud rate".

Section 17.7.1 "Clocking and baud rates".

Table 153. 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	-

8.6.29 FRG0 clock source select register

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

Table 154. 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	-

8.6.30 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 156.

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 17.3.1 "Configure the USART clock and baud rate".

Section 17.7.1 "Clocking and baud rates".

Table 155. 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	-

8.6.31 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 155.
- 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 17.3.1 "Configure the USART clock and baud rate".

Section 17.7.1 "Clocking and baud rates".

Table 156. 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	-

8.6.32 FRG1 clock source select register

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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **109 of 530**

Table 157. 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	-

8.6.33 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 158. 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	Watchdog oscillator	
		0x5	None	
		0x6	None	
		0x7	None	
31:3	-	-	Reserved	0

8.6.34 CLKOUT clock divider register

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

Table 159. 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	-

8.6.35 External trace buffer command register

This register works in conjunction with the MTB master register to start and stop tracing. Also see Section 32.5.4.

Table 160. External trace buffer command register (EXTTRACECMD, address 0x4004 80FC) bit description

Bit	Symbol	Description	Reset value
0	START	Trace start command. Writing a one to this bit sets the TSTART signal to the MTB to HIGH and starts tracing if the TSTARTEN bit in the MTB master register is set to one as well.	0
1	STOP	Trace stop command. Writing a one to this bit sets the TSTOP signal in the MTB to HIGH and stops tracing if the TSTOPEN bit in the MTB master register is set to one as well.	0
31:2	-	Reserved	0

8.6.36 POR captured PIO0 status register 0

The PIOPORCAP0 register captures the state of GPIO port 0 at power-on-reset. Each bit represents the reset state of one GPIO pin. This register is a read-only status register.

Table 161. POR captured PIO status register 0 (PIOPORCAP0, address 0x4004 8100) bit description

Bit	Symbol	Description	Reset value
31:0	PIOSTAT	State of PIO0_31 through PIO0_0 at power-on reset	Implementation dependent

8.6.37 POR captured PIO1 status register 1

The PIOPORCAP1 register captures the state of GPIO port 1 at power-on-reset. Each bit represents the reset state of one GPIO pin. This register is a read-only status register.

Table 162. POR captured PIO status register 1 (PIOPORCAP0, address 0x4004 8104 bit description

Bit	Symbol	Description	Reset value
31:0	PIOSTAT	State of PIO1_21 through PIO1_0 at power-on reset	Implementation dependent

8.6.38 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 163. 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

8.6.39 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 164 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 LPC84x data sheet for the BOD reset and interrupt levels.

Table 164. 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	0
		0	Disable reset function.	
		1	Enable reset function.	
31:5	-	-	Reserved	0x00

8.6.40 System tick counter calibration register

This register determines the value of the SYST_CALIB register.

Table 165. 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	-

8.6.41 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 166. 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	-

8.6.42 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 108. For a description of the NMI functionality, see Section 7.3.2.

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 167. 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 108 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.

8.6.43 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 108).

To use the selected pins for pin interrupts or the pattern match engine, see Section 13.5.2 "Pattern match engine".

Table 168. 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	-

8.6.44 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 108 "Connection of interrupt sources to the NVIC".

Table 169. 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	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **114 of 530**

Table 169. Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description   …continued

Bit	Symbol	Value	Description	Reset value
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	
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	-

8.6.45 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 108 "Connection of interrupt sources to the NVIC".

Table 170. 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	

Table 170. Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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	
6	-		Reserved	-
7	I2C1		I2C1 interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
8	I2C0		I2C0 interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
10:9	-		Reserved	0
11	Cap Touch		Cap Touch interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
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	
20:16	-		Reserved.	-
21	I2C2		I2C2 interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
22	I2C3		I2C3 interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
29:23	-		Reserved.	-
30	USART3		USART3 interrupt wake-up	0
		0	Disabled	
		1	Enabled	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **116 of 530**

Table 170. Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
31	USART4		USART4 interrupt wake-up	
		0	Disabled	
		1	Enabled	

8.6.46 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 watchdog 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 418) is set. See Section 22.5.3 for details.

Table 171. 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	WDTOSC_PD		Watchdog 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 watchdog oscillator is always running.	1
		0	Powered	
		1	Powered down	
15:7	-		Reserved	0b111111111
31:16	-	-	Reserved	0

8.6.47 Wake-up configuration register

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

Table 172. 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	WDTOSC_PD		Watchdog 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 watchdog 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	VREF2_PD		VREF2 wake-up configuration	1
		0	Powered	
		1	Powered down	
12:11	-		Reserved. Always write these bits as 0b01	0b01
13	DAC0		DAC0 wake-up configuration	1
		0	Powered	
		1	Powered down	
14	DAC1		DAC1 wake-up configuration	1
		0	Powered	
		1	Powered down	

Table 172. Wake-up configuration register (PDAWAKECFG, address 0x4004 8234) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
15	ACMP		Analog comparator wake-up configuration	1
		0	Powered	
		1	Powered down	
31:16	-	-	Reserved	0

8.6.48 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 173. 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	1
		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	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **119 of 530**

Table 173. Power configuration register (PDRUNCFG, address 0x4004 8238) bit description

Bit	Symbol	Value	Description	Reset value
6	WDTOSC_PD		Watchdog 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 watchdog 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
12:10	-		Reserved. Always write these bits as 0b011	0b011
13	DAC0		DAC0 power down	1
		0	Powered	
		1	Powered down	
14	DAC1		DAC1 power down	1
		0	Powered	
		1	Powered down	
15	ACMP		Analog comparator power down	1
		0	Powered	
		1	Powered down	
31:16	-	-	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 174).

Table 174. Device ID register values

Part number	Part ID
LPC844M201JBD48	0x00008442
LPC844M201JBD64	0x00008441
LPC844M201JHI33	0x00008444
LPC844M201JHI48	0x00008443
LPC845M301JBD48	0x00008452
LPC845M301JBD64	0x00008451
LPC845M301JHI33	0x00008454
LPC845M301JHI48	0x00008453

8.7 Functional description

8.7.1 Reset

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

The $\overline{RESET}$ 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 164).

The interrupt signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC (see Table 109) 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 170) 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 LPC84x uses the system PLL to create the clocks for the core and peripherals.
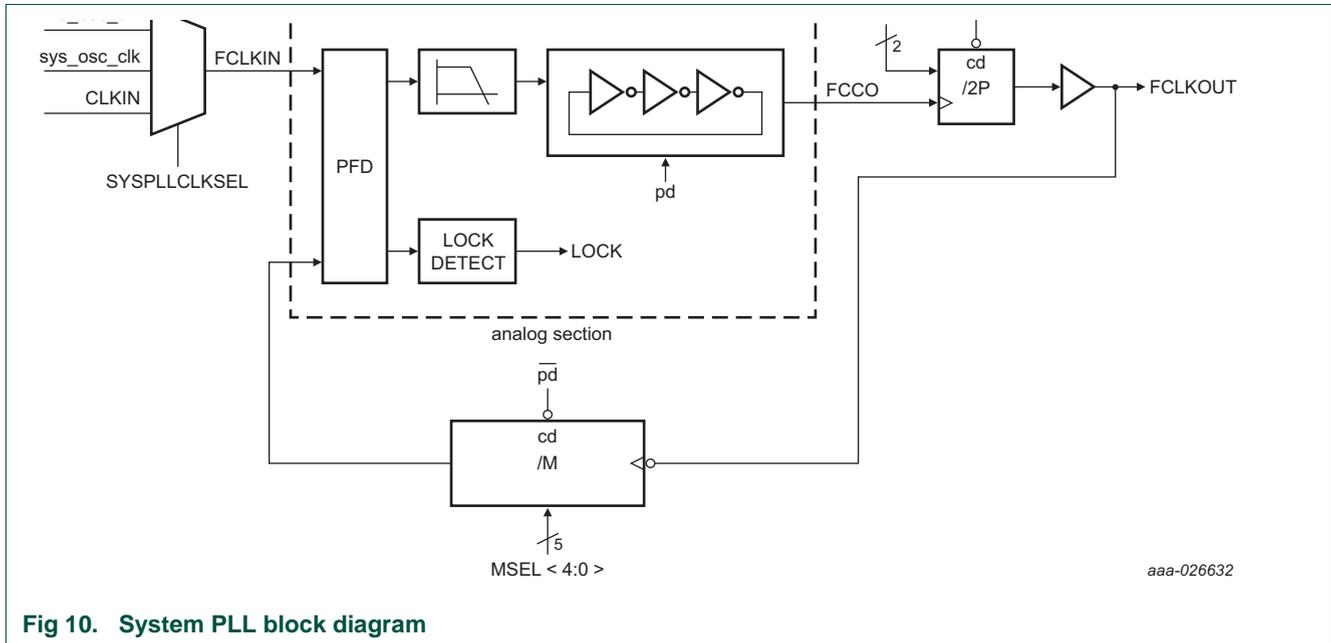
Fig 10. System PLL block diagram

The block diagram of this PLL is shown in Figure 10. 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 30 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **122 of 530**

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 173). 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 127. 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 127.

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 10):

Table 175. 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 ≤ 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).

UM11029

User manual **Rev. 1.6 — 8 December 2017** **123 of 530**

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:

(1)

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

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 127.

Remark: The divider values for P and M must be selected so that the PLL output clock frequency FCLKOUT is ≤ 30 MHz.

Table 176 shows how to configure the PLL for a 12 MHz crystal oscillator using the SYSPLLCTRL register (Table 127).

Table 176. PLL configuration examples

FCLKIN frequency	FCLKOUT frequency	MSEL bits Table 127	M divider value	PSEL bits Table 127	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 173), the PLL will resume its normal operation and will make the lock signal high once it has regained lock on the input clock.

9.1 How to read this chapter

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

9.2 Features

- Select desired on-chip fro_oscout (30 MHz/24 MHz/18 MHz)

9.3 General description

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 (30 MHz/ 24 MHz/18 MHz). This is performed by executing a function, which is pointed by a pointer within the ROM Driver Table. Figure 11 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.
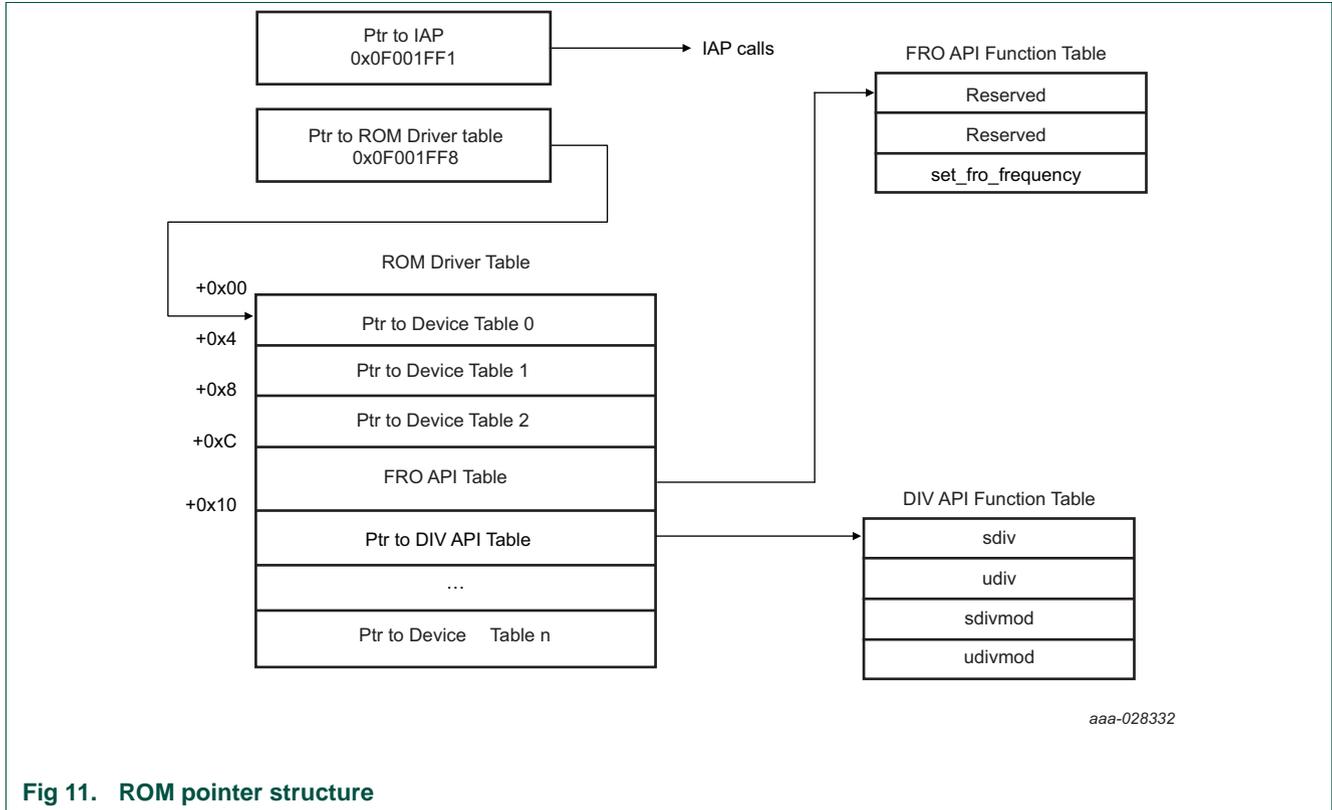
Fig 11. ROM pointer structure

9.4 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. An example is provided with the code bundle software package on nxp.com.

The following function prototypes are used:

Table 177. FRO API call

Function prototype	API description	Reference
`void set_fro_frequency(uint32_t iFreq);`	Setup the fro_oscout to either 30 MHz, 24 MHz or 18 MHz.	9.4.1

9.4.1 set_fro_frequency

This routine sets up the required fro_oscout (30 MHz/24 MHz/18 MHz). The requested frequency is set up and the appropriate factory trim value is used.

See Figure 9 "LPC84x clock generation (continued)" for more details to select fro_oscout (30/24/18 MHz) or select divided fro_oscout (1.125 MHz, 1.5 MHz, 1.875 MHz, 9 MHz, 12 MHz, or 15 MHz) based on FAIM low power boot value.

Table 178 shows the set_fro_frequency routine.

Table 178. set_fro_frequency routine

Routine	sidiv
Prototype	void set_fro_frequency(uint32_t iFreq);
Input parameter	**Param0 —** Required frequency (in kHz). Example: parameter 1: 18000 => 18MHz, 24000=>24Mhz, 30000=>30Mhz.
Return	None.
Description	Setup the fro_oscout to either 30 MHz, 24 MHz or 18 MHz.

9.4.1.1 Param0: frequency

The frequency is the required fro_oscout, 30 MHz, 24 MHz or 18 MHz.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **126 of 530**

10.1 How to read this chapter

The switch matrix is identical for all LPC84x parts.

10.2 Features

- Flexible assignment of digital peripheral functions to pins
- Enable/disable of analog functions

10.3 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.

10.3.1 Connect an internal signal to a package pin

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

Fig 12. Example: Connect function U0_RXD and U0_TXD to pins 4 and 14
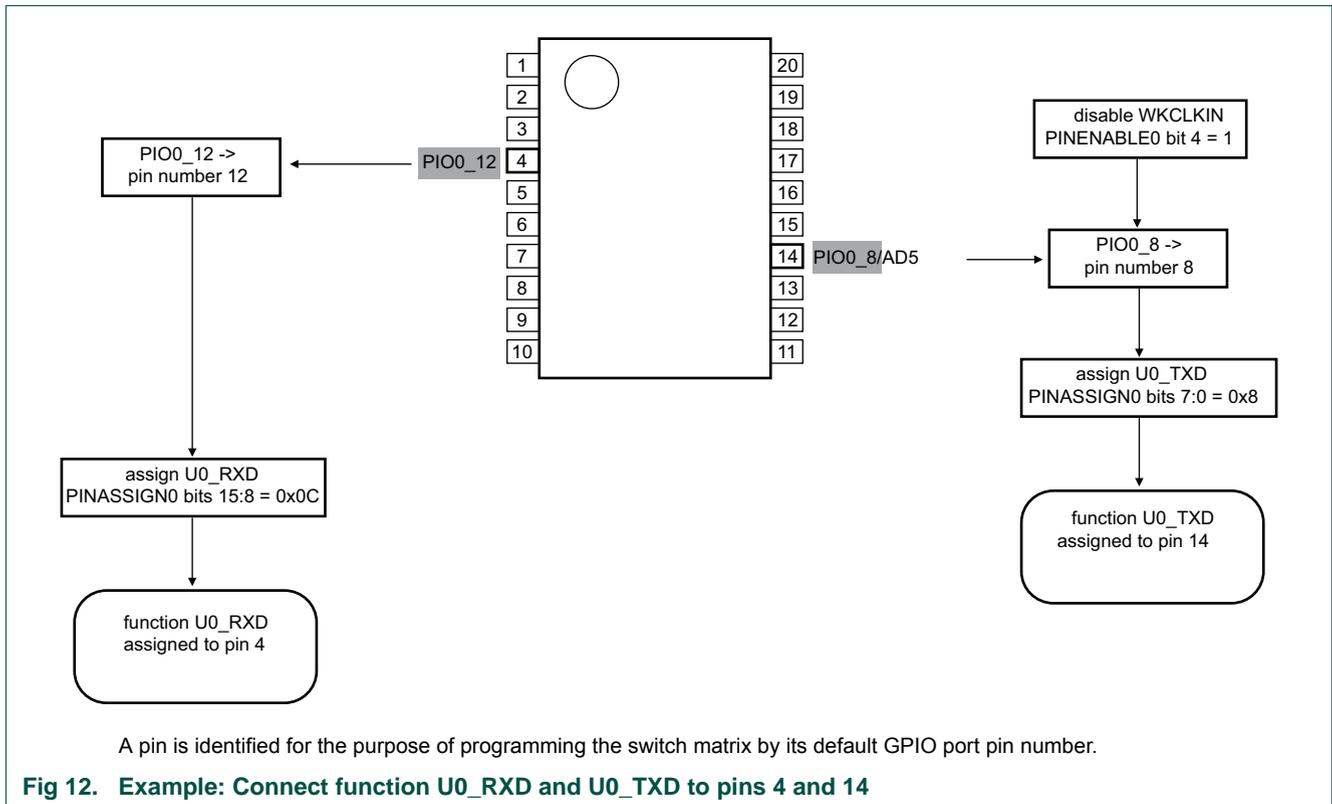
The switch matrix connects all internal signals listed in the table of movable functions through the pin assignment registers to external pins on the package. External pins are identified by their default GPIO pin number PIO0_n or PIO1_(n-32). Follow these steps to connect an internal signal FUNC to 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 179 or in the data sheet.

2. Use the LPC84x data sheet to decide which pin x on the LPC84x 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **128 of 530**

10.3.2 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 LPC84x 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.

  By default, all pins except pins PIO0_2, PIO0_3, PIO0_5, PIO1_8, and PIO1_9 are assigned to GPIO. By default, PIO0_2 and PIO0_3 are configured by FAIM.

- 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.

10.3.3 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.

10.4 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.

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.
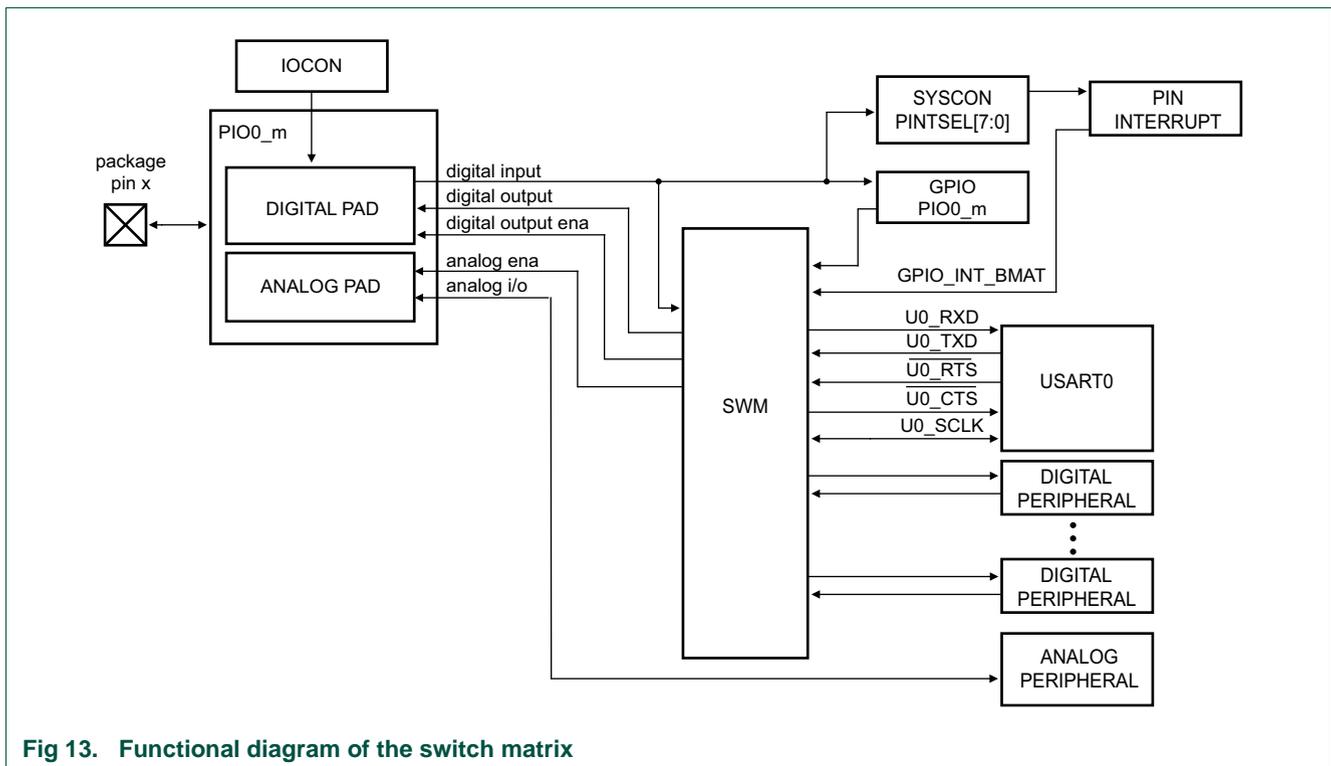
Fig 13. Functional diagram of the switch matrix

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 13). 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 and also assign one ore more SCT inputs to pin PIO0_1 through the PINASSIGN registers to feed the CLKIN into the SCT.

  You can send the input on one pin to all SCT inputs to use as an SCT abort signal.

- 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.

10.4.1 Movable functions

Table 179. 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 181
U0_RXD	I	Receiver input for USART0.	PINASSIGN0	Table 181
U0_RTS	O	Request To Send output for USART0.	PINASSIGN0	Table 181
U0_CTS	I	Clear To Send input for USART0.	PINASSIGN0	Table 181
U0_SCLK	I/O	Serial clock input/output for USART0 in synchronous mode.	PINASSIGN1	Table 182
U1_TXD	O	Transmitter output for USART1.	PINASSIGN1	Table 182
U1_RXD	I	Receiver input for USART1.	PINASSIGN1	Table 182
U1_RTS	O	Request To Send output for USART1.	PINASSIGN1	Table 182
U1_CTS	I	Clear To Send input for USART1.	PINASSIGN2	Table 183
U1_SCLK	I/O	Serial clock input/output for USART1 in synchronous mode.	PINASSIGN2	Table 183
U2_TXD	O	Transmitter output for USART2.	PINASSIGN2	Table 183
U2_RXD	I	Receiver input for USART2.	PINASSIGN2	Table 183
U2_RTS	O	Request To Send output for USART1.	PINASSIGN3	Table 184
U2_CTS	I	Clear To Send input for USART1.	PINASSIGN3	Table 184
U2_SCLK	I/O	Serial clock input/output for USART1 in synchronous mode.	PINASSIGN3	Table 184
SPI0_SCK	I/O	Serial clock for SPI0.	PINASSIGN3	Table 184

Table 179. 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
SPI0_MOSI	I/O	Master Out Slave In for SPI0.	PINASSIGN4	Table 185
SPI0_MISO	I/O	Master In Slave Out for SPI0.	PINASSIGN4	Table 185
SPI0_SSEL0	I/O	Slave select 0 for SPI0.	PINASSIGN4	Table 185
SPI0_SSEL1	I/O	Slave select 0 for SPI1.	PINASSIGN4	Table 185
SPI0_SSEL2	I/O	Slave select 0 for SPI2.	PINASSIGN5	Table 186
SPI0_SSEL3	I/O	Slave select 0 for SPI3.	PINASSIGN5	Table 186
SPI1_SCK	I/O	Serial clock for SPI1.	PINASSIGN5	Table 186
SPI1_MOSI	I/O	Master Out Slave In for SPI1.	PINASSIGN5	Table 186
SPI1_MISO	I/O	Master In Slave Out for SPI1.	PINASSIGN6	Table 187
SPI1_SSEL0	I/O	Slave select 0 for SPI1.	PINASSIGN6	Table 187
SPI1_SSEL1	I/O	Slave select 1 for SPI1.	PINASSIGN6	Table 187
SCT_PIN0	I	Pin input 0 to the SCT input multiplexer.	PINASSIGN6	Table 187
SCT_PIN1	I	Pin input 1 to the SCT input multiplexer.	PINASSIGN7	Table 188
SCT_PIN2	I	Pin input 2 to the SCT input multiplexer.	PINASSIGN7	Table 188
SCT_PIN3	I	Pin input 3 to the SCT input multiplexer.	PINASSIGN7	Table 188
SCT_OUT0	O	SCT output 0.	PINASSIGN7	Table 188
SCT_OUT1	O	SCT output 1.	PINASSIGN8	Table 189
SCT_OUT2	O	SCT output 2.	PINASSIGN8	Table 189
SCT_OUT3	O	SCT output 3.	PINASSIGN8	Table 189
SCT_OUT4	O	SCT output 4.	PINASSIGN8	Table 189
SCT_OUT5	O	SCT output 5.	PINASSIGN9	Table 190
SCT_OUT6	O	SCT output 6.	PINASSIGN9	
I2C1_SDA	I/O	I²C1-bus data input/output.	PINASSIGN9	Table 190
I2C1_SCL	I/O	I²C1-bus clock input/output.	PINASSIGN9	Table 190
I2C2_SDA	I/O	I²C2-bus data input/output.	PINASSIGN10	Table 191
I2C2_SCL	I/O	I²C2-bus clock input/output.	PINASSIGN10	Table 191
I2C3_SDA	I/O	I²C3-bus data input/output.	PINASSIGN10	Table 191
I2C3_SCL	I/O	I²C3-bus clock input/output.	PINASSIGN10	Table 191
ACMP_O	O	Analog comparator output	PINASSIGN11	Table 192
CLKOUT	O	Clock output	PINASSIGN11	Table 192
GPIO_INT_BMAT	O	Output of the pattern match engine	PINASSIGN11	Table 192
U3_TXD	O	Transmitter output for USART3	PINASSIGN11	Table 192
U3_RXD	I	Receiver input for USART3	PINASSIGN12	Table 193
U3_SCLK	I/O	Serial clock input/output for USART3 in synchronous mode	PINASSIGN12	Table 193
U4_TXD	O	Transmitter output for USART4	PINASSIGN12	Table 193
U4_RXD	I	Receiver input for USART4	PINASSIGN12	Table 193
U4_SCLK	I/O	Serial clock input/output for USART4 in synchronous mode.	PINASSIGN13	Table 194
T0_MAT0	O	Timer Match channel 0	PINASSIGN13	Table 194
T0_MAT1	O	Timer Match channel 1	PINASSIGN13	Table 194
T0_MAT2	O	Timer Match channel 2	PINASSIGN13	Table 194

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **132 of 530**

Table 179. 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
T0_MAT3	O	Timer Match channel 3	PINASSIGN14	Table 195
T0_CAP0	I	Timer Capture channel 0	PINASSIGN14	Table 195
T0_CAP1	I	Timer Capture channel 1	PINASSIGN14	Table 195
T0_CAP2	I	Timer Capture channel 2	PINASSIGN14	Table 195

10.4.2 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 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.

There are two types of functions which 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.

10.5 Register description

Table 180. 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 181
PINASSIGN1	R/W	0x004	Pin assign register 1. Assign movable functions U0_SCLK, U1_TXD, U1_RXD, U1_RTS.	0xFFFF FFFF	Table 182
PINASSIGN2	R/W	0x008	Pin assign register 2. Assign movable functions U1_CTS, U1_SCLK, U2_TXD, U2_RXD.	0xFFFF FFFF	Table 183
PINASSIGN3	R/W	0x00C	Pin assign register 3. Assign movable function U2_RTS, U2_CTS, U2_SCLK, SPI0_SCK.	0xFFFF FFFF	Table 184
PINASSIGN4	R/W	0x010	Pin assign register 4. Assign movable functions SPI0_MOSI, SPI0_MISO, SPI0_SSEL0, SPI0_SSEL1.	0xFFFF FFFF	Table 185
PINASSIGN5	R/W	0x014	Pin assign register 5. Assign movable functions SPI0_SSEL2, SPI0_SSEL3, SPI1_SCK, SPI1_MOSI	0xFFFF FFFF	Table 186
PINASSIGN6	R/W	0x018	Pin assign register 6. Assign movable functions SPI1_MISO, SPI1_SSEL0, SPI1_SSEL1, SCT0_IN0.	0xFFFF FFFF	Table 187
PINASSIGN7	R/W	0x01C	Pin assign register 7. Assign movable functions SCT_IN1, SCT_IN2, SCT_IN3, SCT_OUT0.	0xFFFF FFFF	Table 188
PINASSIGN8	R/W	0x020	Pin assign register 8. Assign movable functions SCT_OUT1, SCT_OUT2, SCT_OUT3, SCT_OUT4.	0xFFFF FFFF	Table 189
PINASSIGN9	R/W	0x024	Pin assign register 9. Assign movable functions SCT_OUT5, SCT_OUT6, I2C1_SDA, I2C1_SCL, I2C2_SDA.	0xFFFF FFFF	Table 190
PINASSIGN10	R/W	0x028	Pin assign register 10. Assign movable functions I2C2_SDA, I2C2_SCL, I2C3_SDA, I2C3_SCL	0xFFFF FFFF	Table 191
PINASSIGN11	R/W	0x02C	Pin assign register 11. Assign movable functions COMP0_OUT, CLKOUT, GPIO_INT_BMAT, UART3_TXD	0xFFFF FFFF	Table 192
PINASSIGN12	R/W	0x030	Pin assign register 12. Assign movable functions UART3_RXD, UART3_SCLK, UART4_TXD, UART4_RXD.	0xFFFF FFFF	Table 193
PINASSIGN13	R/W	0x034	Pin assign register 13. Assign movable functions UART4_SCLK, T0_MAT0, T0_MAT1, T0_MAT2.	0xFFFF FFFF	Table 194

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **134 of 530**

Table 180. Register overview: Switch matrix (base address 0x4000 C000) *…continued*

Name	Access	Offset	Description	Reset value	Reference
PINASSIGN14	R/W	0x038	Pin assign register 14. Assign movable functions T0_MAT3, T0_CAP0, T0_CAP1, T0_CAP2.	0xFFFF FFFF	Table 195
PINENABLE0	R/W	0x1C0	Pin enable register 0. Enables fixed-pin functions ACMP_In, SWCLK, SWDIO, XTALIN, XTALOUT, RESET, CLKIN, VDDCMP, I2C0_SDA, I2C0_SCL, ADC_n, DACOUTn, CAPT_X0, CAPT_X1, CAPT_X2, and CAPT_X3.	0xFFFF FDXF[1]	Table 196
PINENABLE1	R/W	0x1C4	Pin enable register 1. Enables fixed-pin functions CAPT_X4, CAPT_X5, CAPT_X6, CAPT_X7, CAPT_X8, CAPT_X4, CAPT_YL and CAPT_YH.	0xFFFF FECF	Table 197

[1]   x = reset values are controlled by FAIM for bit 5 and bit 6.
       For bit 4 and bit 7, the reset value is 1.

10.5.1 Pin assign register 0

Table 181. 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

10.5.2 Pin assign register 1

Table 182. 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

10.5.3 Pin assign register 2

Table 183. 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

10.5.4 Pin assign register 3

Table 184. 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

10.5.5 Pin assign register 4

Table 185. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **137 of 530**

10.5.6 Pin assign register 5

Table 186. 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

10.5.7 Pin assign register 6

Table 187. 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	SCT0_GPIO_IN_A_I	SCT0_GPIO_IN_A 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **138 of 530**

10.5.8 Pin assign register 7

Table 188. Pin assign register 7 (PINASSIGN7, address 0x4000 C01C) bit description

Bit	Symbol	Description	Reset value
7:0	SCT0_GPIO_IN_B_I	SCT0_GPIO_IN_B_Ifunction 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	SCT0_GPIO_IN_C_I	SCT0_GPIO_IN_C 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	SCT0_GPIO_IN_D_I	SCT0_GPIO_IN_D 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	SCT_OUT0_O	SCT_OUT0 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

10.5.9 Pin assign register 8

Table 189. Pin assign register 8 (PINASSIGN8, address 0x4000 C020) bit description

Bit	Symbol	Description	Reset value
7:0	SCT_OUT1_O	SCT_OUT1 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	SCT_OUT2_O	SCT_OUT2 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	SCT_OUT3_O	SCT_OUT3 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	SCT_OUT4_O	SCT_OUT4 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

10.5.10 Pin assign register 9

Table 190. Pin assign register 9 (PINASSIGN9, address 0x4000 C024) bit description

Bit	Symbol	Description	Reset value
7:0	SCT_OUT5_O	SCT_OUT5 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	SCT_OUT6_O	SCT_OUT6 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	I2C1_SDA_IO	I2C1_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
31:24	I2C1_SCL_IO	I2C1_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

10.5.11 Pin assign register 10

Table 191. Pin assign register 10 (PINASSIGN10, address 0x4000 C028) bit description

Bit	Symbol	Description	Reset value
7:0	I2C2_SDA_IO	I2C1_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
15:8	I2C2_SCL_IO	I2C1_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
23:16	I2C3_SDA_IO	I2C3_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
31:24	I2C3_SCL_IO	I2C3_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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **140 of 530**

10.5.12 Pin assign register 11

Table 192. Pin assign register 11 (PINASSIGN11, address 0x4000 C02C) bit description

Bit	Symbol	Description	Reset value
7:0	COMP0_OUT_O	COMP0_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	UART3_TXD	UART3_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

10.5.13 Pin assign register 12

Table 193. Pin assign register 12 (PINASSIGN12, address 0x4000 C030) bit description

Bit	Symbol	Description	Reset value
7:0	UART3_RXD	UART3_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
15:8	UART3_SCLK	UART3_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	UART4_TXD	UART4_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	UART4_RXD	UART4_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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **141 of 530**

10.5.14 Pin assign register 13

Table 194. Pin assign register 13 (PINASSIGN13, address 0x4000 C034) bit description

Bit	Symbol	Description	Reset value
7:0	UART4_SCLK	UART4_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	T0_MAT0	T0_MAT0 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	T0_MAT1	UT0_MAT1 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	T0_MAT2	T0_MAT2 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

10.5.15 Pin assign register 14

Table 195. Pin assign register 14 (PINASSIGN14, address 0x4000 C038) bit description

Bit	Symbol	Description	Reset value
7:0	T0_MAT3	T0_MAT3 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	T0_CAP0	T0_CAP0 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	T0_CAP1	T0_CAP1 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	T0_CAP2	T0_CAP2 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

10.5.16 PINENABLE 0

Table 196. 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.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **142 of 530**

Table 196. 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	XTALIN		XTALIN function select.	1
		0	XTALIN enabled on pin PIO0_8.	
		1	XTALIN disabled.	
8	XTALOUT		XTALOUT function select.	1
		0	XTALOUT enabled on pin PIO0_9.	
		1	XTALOUT disabled.	
9	RESETN		RESETN function select.	0
		0	RESETN enabled on pin PIO0_5.	
		1	RESETN disabled.	
10	CLKIN		CLKIN function select.	1
		0	CLKIN enabled on pin PIO0_1.	
		1	CLKIN disabled.	
11	VDDCMP		VDDCMP function select.	1
		0	VDDCMP enabled on pin PIO0_6.	
		1	VDDCMP disabled.	
12	I2C0_SDA		I2C0_SDA function select.	1
		0	I2C0_SDA enabled on pin PIO0_11.	
		1	I2C0_SDA disabled.	
13	I2C0_SCL		I2C0_SCL function select.	1
		0	I2C0_SCL enabled on pin PIO0_10.	
		1	I2C0_SCL disabled.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **143 of 530**

Table 196. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description

Bit	Symbol	Value	Description	Reset value
14	ADC_0		ADC_0 function select.	1
		0	ADC_0 enabled on pin PIO0_7.	
		1	ADC_0 disabled.	
15	ADC_1		ADC_1 function select.	1
		0	ADC_1 enabled on pin PIO0_6.	
		1	ADC_1 disabled.	
16	ADC_2		ADC_2 function select.	1
		0	ADC_2 enabled on pin PIO0_14.	
		1	ADC_2 disabled.	
17	ADC_3		ADC_3 function select.	1
		0	ADC_3 enabled on pin PIO0_23.	
		1	ADC_3 disabled.	
18	ADC_4		ADC_4 function select.	1
		0	ADC_4 enabled on pin PIO0_22.	
		1	ADC_4 disabled.	
19	ADC_5		ADC_5 function select.	1
		0	ADC_5 enabled on pin PIO0_21.	
		1	ADC_5 disabled.	
20	ADC_6		ADC_6 function select.	1
		0	ADC_6 enabled on pin PIO0_20.	
		1	ADC_6 disabled.	
21	ADC_7		ADC_7 function select.	1
		0	ADC_7 enabled on pin PIO0_19.	
		1	ADC_7 disabled.	
22	ADC_8		ADC_8 function select.	1
		0	ADC_8 enabled on pin PIO0_18.	
		1	ADC_8 disabled.	
23	ADC_9		ADC_9 function select.	1
		0	ADC_9 enabled on pin PIO0_17.	
		1	ADC_9 disabled.	
24	ADC_10		ADC_10 function select.	1
		0	ADC_10 enabled on pin PIO0_13.	
		1	ADC_10 disabled.	
25	ADC_11		ADC_11 function select.	1
		0	ADC_11 enabled on pin PIO0_4.	
		1	ADC_11 disabled.	
26	DACOUT0		DACOUT0 function select.	1
		0	DACOUT0 enabled on pin PIO0_17.	
		1	DACOUT0 disabled.	

Table 196. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description

Bit	Symbol	Value	Description	Reset value
27	DACOUT1		DACOUT1 function select.	1
		0	DACOUT1 enabled on pin PIO0_29.	
		1	DACOUT1 disabled.	
28	CAPT_X0		CAPT_X0 function select.	1
		0	CAPT_X0 enabled on pin PIO0_31.	
		1	CAPT_X0 disabled.	
29	CAPT_X1		CAPT_X1 function select.	1
		0	CAPT_X1 enabled on pin PIO1_0.	
		1	CAPT_X1 disabled.	
30	CAPT_X2		CAPT_X2 function select.	1
		0	CAPT_X2 enabled on pin PIO1_1.	
		1	CAPT_X2 disabled.	
31	CAPT_X3		CAPT_X3 function select.	1
		0	CAPT_X3 enabled on pin PIO1_2.	
		1	CAPT_X3 disabled.	

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

10.5.17 PINENABLE 1

Table 197. Pin enable register 1 (PINENABLE1, address 0x4000 C1C4) bit description

Bit	Symbol	Value	Description	Reset value
0	CAPT_X4		CAPT_X4 function select.	1
		0	CAPT_X4 enabled on pin PIO1_3.	
		1	CAPT_X4 disabled.	
1	CAPT_X5		CAPT_X5 function select.	1
		0	CAPT_X5 enabled on pin PIO1_4.	
		1	CAPT_X5 disabled.	
2	CAPT_X6		CAPT_X6 function select.	1
		0	CAPT_X6 enabled on pin PIO1_5.	
		1	CAPT_X6 disabled.	
3	CAPT_X7		CAPT_X7 function select.	1
		0	CAPT_X7 enabled on pin PIO1_6.	
		1	CAPT_X7 disabled.	
4	CAPT_X8		CAPT_X8 function select.	1
		0	CAPT_X8 enabled on pin PIO1_7.	
		1	CAPT_X8 disabled.	
5	CAPT_YL		CAPT_YL function select.	0
		0	CAPT_YL enabled on pin PIO1_8.	
		1	CAPT_YL disabled.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **145 of 530**

Table 197. Pin enable register 1 (PINENABLE1, address 0x4000 C1C4) bit description

Bit	Symbol	Value	Description	Reset value
6	CAPT_YH		CAPT_YH function select.	0
		0	CAPT_YH enabled on pin PIO1_9.	
		1	CAPT_YH disabled.	

11.1 How to read this chapter

The IOCON block is identical for all LPC84x parts. Registers for pins that are not available on a specific package are reserved.

Table 198. Pinout summary

Package	Pins/configuration registers available
HVQFN33	PIO0_0 to PIO0_28
HVQFN48	PIO0_0 to PIO0_31; PIO1_0 to PIO1_9
LQFP48	PIO0_0 to PIO0_31; PIO1_0 to PIO1_9
LQFP64	PIO0_0 to PIO0_31; PIO1_0 to PIO1_21

11.2 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 LPC84x 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.

11.3 Basic configuration

Enable the clock to the IOCON in the SYSAHBCLKCTRL register (Table 147, 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.

11.4 General description

11.4.1 Pin configuration
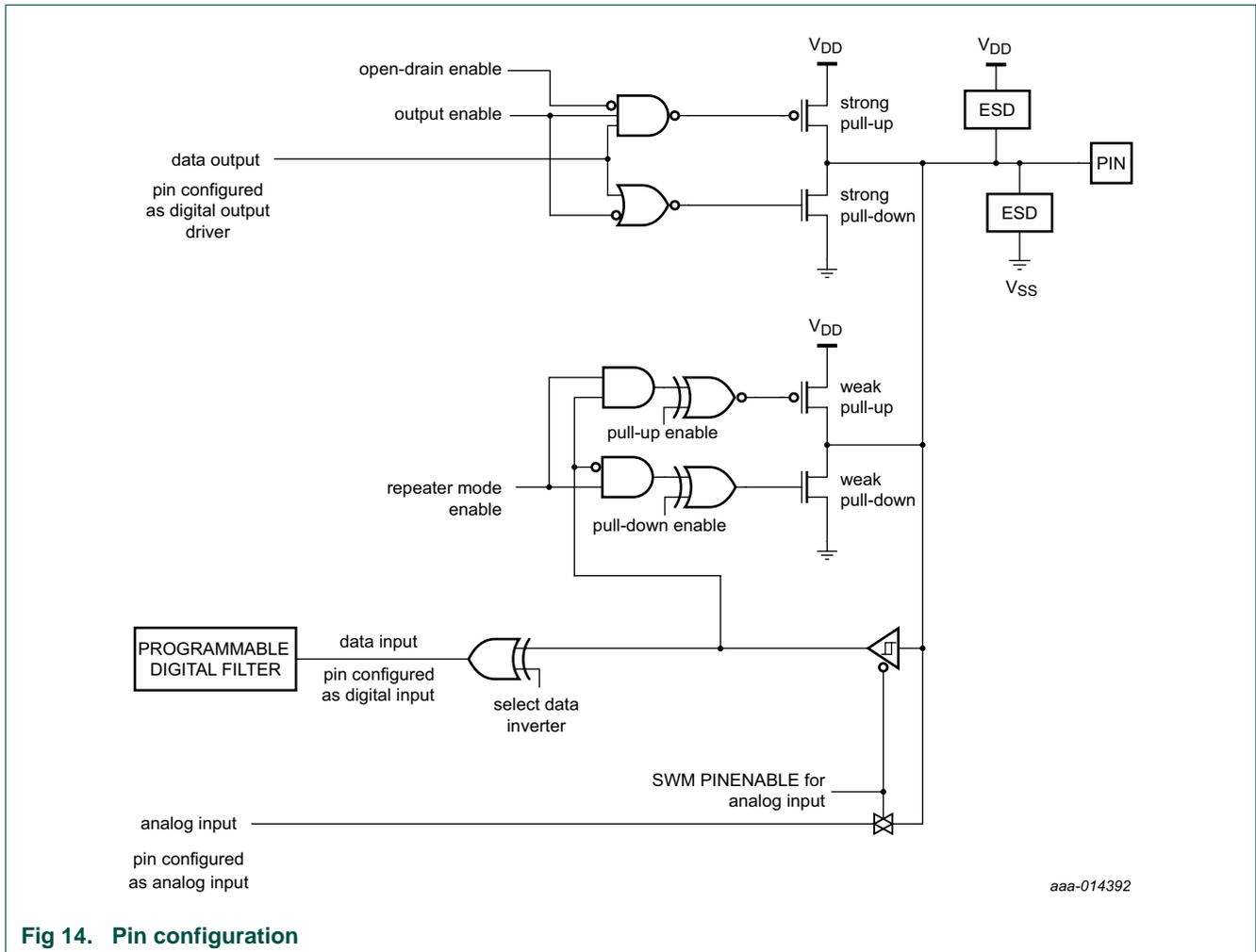
Fig 14. Pin configuration

11.4.2 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 179 "Movable functions (assign to pins PIO0_0 to PIO0_31 and PIO1_0 to PIO1_21 through switch matrix)"

11.4.3 Pin mode

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

MODE bit is controlled by FAIM value during reset, two FAIM bits data for each PAD MODE bit. After reset release, PAD pull-up/pull-down can be updated by software. See Section 4.2.1 "FAIM bit definitions".

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.

11.4.4 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.

11.4.5 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.

11.4.6 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.

11.4.7 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} \acute{} $ (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 163 "IOCON glitch filter clock divider registers 6 to 0 (IOCONCLKDIV[6:0], address 0x4004 8134 (IOCONCLKDIV6) to 0x004 814C (IOCONFILTCLKDIV0)) bit description"

11.5 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 200 for the IOCON register map ordered by pin name. If FAIM is not configured, the default IOCON should have pull-up enabled in the MODE field.

Table 199. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
PIO0_17	R/W	0x000	I/O configuration for pin PIO0_17/ADC_9/DACOUT0	FAIM value dependent	Table 201
PIO0_13	R/W	0x004	I/O configuration for pin PIO0_13/ADC_10	FAIM value dependent	Table 202
PIO0_12	R/W	0x008	I/O configuration for pin PIO0_12	FAIM value dependent	Table 203
PIO0_5	R/W	0x00C	I/O configuration for pin PIO0_5/RESET	FAIM value dependent	Table 204
PIO0_4	R/W	0x010	I/O configuration for pin PIO0_4/ADC_11/TRSTN/WAKEUP	FAIM value dependent	Table 205
PIO0_3	R/W	0x014	I/O configuration for pin PIO0_3/SWCLK	FAIM value dependent	Table 206
PIO0_2	R/W	0x018	I/O configuration for pin PIO0_2/SWDIO	FAIM value dependent	Table 207
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.	FAIM value dependent	Table 208
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.	FAIM value dependent	Table 209
PIO0_16	R/W	0x024	I/O configuration for pin PIO0_16	FAIM value dependent	Table 210
PIO0_15	R/W	0x028	I/O configuration for pin PIO0_15	FAIM value dependent	Table 211
PIO0_1	R/W	0x02C	I/O configuration for pin PIO0_1/ACMP_I2/CLKIN	FAIM value dependent	Table 212
-	-	0x030	Reserved	FAIM value dependent	-
PIO0_9	R/W	0x034	I/O configuration for pin PIO0_9/XTALOUT	FAIM value dependent	Table 213
PIO0_8	R/W	0x038	I/O configuration for pin PIO0_8/XTALIN	FAIM value dependent	Table 214

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **150 of 530**

Table 199. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
PIO0_7	R/W	0x03C	I/O configuration for pin PIO0_7/ADC_0	FAIM value dependent	Table 215
PIO0_6	R/W	0x040	I/O configuration for pin PIO0_6/ADC_1/ CMPVREF	FAIM value dependent	Table 216
PIO0_0	R/W	0x044	I/O configuration for pin PIO0_0/ACMP_I1	FAIM value dependent	Table 217
PIO0_14	R/W	0x048	I/O configuration for pin PIO0_14/ ACMP_I3/ADC_2	FAIM value dependent	Table 218
-	-	0x04C	Reserved.	FAIM value dependent	-
PIO0_28	R/W	0x050	I/O configuration for pin PIO0_28/WKTCLKIN	FAIM value dependent	Table 219
PIO0_27	R/W	0x054	I/O configuration for pin PIO0_27	FAIM value dependent	Table 220
PIO0_26	R/W	0x058	I/O configuration for pin PIO0_26	FAIM value dependent	Table 221
PIO0_25	R/W	0x05C	I/O configuration for pin PIO0_25	FAIM value dependent	Table 222
PIO0_24	R/W	0x060	I/O configuration for pin PIO0_24	FAIM value dependent	Table 223
PIO0_23	R/W	0x064	I/O configuration for pin PIO0_23/ADC_3/ACMP_I4	FAIM value dependent	Table 224
PIO0_22	R/W	0x068	I/O configuration for pin PIO0_22/ADC_4	FAIM value dependent	Table 225
PIO0_21	R/W	0x06C	I/O configuration for pin PIO0_21/ADC_5	FAIM value dependent	Table 226
PIO0_20	R/W	0x070	I/O configuration for pin PIO0_20/ADC_6	FAIM value dependent	Table 227
PIO0_19	R/W	0x074	I/O configuration for pin PIO0_19/ADC_7	FAIM value dependent	Table 228
PIO0_18	R/W	0x078	I/O configuration for pin PIO0_18/ADC_8	FAIM value dependent	Table 229
PIO1_8	R/W	0x07C	I/O configuration for pin PIO1_8/CAPT_YL	FAIM value dependent	Table 230
PIO1_9	R/W	0x080	I/O configuration for pin PIO1_9/CAPT_YH	FAIM value dependent	Table 231
PIO1_12	R/W	0x084	I/O configuration for pin PIO1_12	FAIM value dependent	Table 232
PIO1_13	R/W	0x088	I/O configuration for pin PIO1_13	FAIM value dependent	Table 233
PIO0_31	R/W	0x08C	I/O configuration for pin PIO0_31/CAPT_X0	FAIM value dependent	Table 234
PIO1_0	R/W	0x090	I/O configuration for pin PIO1_0/CAPT_X1	FAIM value dependent	Table 235
PIO1_1	R/W	0x094	I/O configuration for pin PIO1_1/CAPT_X2	FAIM value dependent	Table 236
PIO1_2	R/W	0x098	I/O configuration for pin PIO1_2/CAPT_X3	FAIM value dependent	Table 237
PIO1_14	R/W	0x09C	I/O configuration for pin PIO1_14	FAIM value dependent	Table 238
PIO1_15	R/W	0x0A0	I/O configuration for pin PIO1_15	FAIM value dependent	Table 239
PIO1_3	R/W	0x0A4	I/O configuration for pin PIO1_3/CAPT_X4	FAIM value dependent	Table 240
PIO1_4	R/W	0x0A8	I/O configuration for pin PIO1_4/CAPT_X5	FAIM value dependent	Table 241
PIO1_5	R/W	0x0AC	I/O configuration for pin PIO1_5/CAPT_X6	FAIM value dependent	Table 242
PIO1_16	R/W	0x0B0	I/O configuration for pin PIO1_16	FAIM value dependent	Table 243
PIO1_17	R/W	0x0B4	I/O configuration for pin PIO1_17	FAIM value dependent	Table 244
PIO1_6	R/W	0x0B8	I/O configuration for pin PIO1_6/CAPT_X7	FAIM value dependent	Table 245
PIO1_18	R/W	0x0BC	I/O configuration for pin PIO1_18	FAIM value dependent	Table 246
PIO1_19	R/W	0x0C0	I/O configuration for pin PIO1_19	FAIM value dependent	Table 247
PIO1_7	R/W	0x0C4	I/O configuration for pin PIO1_7/CAPT_X8	FAIM value dependent	Table 248
PIO0_29	R/W	0x0C8	I/O configuration for pin PIO0_29/DACOUT1	FAIM value dependent	Table 249
PIO0_30	R/W	0x0CC	I/O configuration for pin PIO0_30/ACMP_I5	FAIM value dependent	Table 250

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **151 of 530**

Table 199. Register overview: I/O configuration (base address 0x4004 4000)

Name	Access	Address offset	Description	Reset value	Reference
PIO1_20	R/W	0x0D0	I/O configuration for pin PIO1_20	FAIM value dependent	Table 251
PIO1_21	R/W	0x0D4	I/O configuration for pin PIO1_21	FAIM value dependent	Table 252
PIO1_11	R/W	0x0D8	I/O configuration for pin PIO1_11	FAIM value dependent	Table 253
PIO1_10	R/W	0x0DC	I/O configuration for pin PIO1_10	FAIM value dependent	Table 254

Table 200. 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 217
PIO0_1	0x02C	no	yes	yes	no	Table 212
PIO0_2	0x018	no	no	yes	yes	Table 207
PIO0_3	0x014	no	no	yes	yes	Table 206
PIO0_4	0x010	no	yes	yes	no	Table 205
PIO0_5	0x00C	no	no	yes	no	Table 204
PIO0_6	0x040	no	yes	yes	no	Table 216
PIO0_7	0x03C	no	yes	yes	no	Table 215
PIO0_8	0x038	no	yes	yes	no	Table 214
PIO0_9	0x034	no	yes	yes	no	Table 213
PIO0_10	0x020	yes	no	yes	no	Table 209
PIO0_11	0x01C	yes	no	yes	no	Table 208
PIO0_12	0x008	no	no	yes	yes	Table 203
PIO0_13	0x004	no	yes	yes	no	Table 202
PIO0_14	0x048	no	yes	yes	no	Table 218
PIO0_15	0x028	no	no	yes	no	Table 211
PIO0_16	0x024	no	no	yes	yes	Table 210
PIO0_17	0x000	no	yes	yes	no	Table 201
PIO0_18	0x078	no	yes	yes	no	Table 229
PIO0_19	0x074	no	yes	yes	no	Table 228
PIO0_20	0x070	no	yes	yes	no	Table 227
PIO0_21	0x06C	no	yes	yes	no	Table 226
PIO0_22	0x068	no	yes	yes	no	Table 225
PIO0_23	0x064	no	yes	yes	no	Table 224
PIO0_24	0x060	no	no	yes	no	Table 223
PIO0_25	0x05C	no	no	yes	no	Table 222
PIO0_26	0x058	no	no	yes	no	Table 221
PIO0_27	0x054	no	no	yes	no	Table 220
PIO0_28	0x050	no	no	yes	no	Table 219
PIO0_29	0x0C8	no	yes	yes	no	Table 249
PIO0_30	0x0CC	no	yes	yes	no	Table 250
PIO0_31	0x08C	no	no	yes	no	Table 234
PIO1_0	0x090	no	no	yes	no	Table 235

Table 200. I/O configuration registers ordered by pin name

Name	Address offset	True open-drain	Analog[1]	Digital filter	High-drive output	Reference
PIO1_1	0x094	no	no	yes	no	Table 236
PIO1_2	0x098	no	no	yes	no	Table 237
PIO1_3	0x0A4	no	no	yes	no	Table 240
PIO1_4	0x0A8	no	no	yes	no	Table 241
PIO1_5	0x0AC	no	no	yes	no	Table 242
PIO1_6	0x0B8	no	no	yes	no	Table 245
PIO1_7	0x0C4	no	no	yes	no	Table 248
PIO1_8	0x07C	no	no	yes	no	Table 230
PIO1_9	0x080	no	no	yes	no	Table 231
PIO1_10	0x0DC	no	no	yes	no	Table 254
PIO1_11	0x0D8	no	no	yes	no	Table 253
PIO1_12	0x084	no	no	yes	no	Table 232
PIO1_13	0x088	no	no	yes	no	Table 233
PIO1_14	0x09C	no	no	yes	no	Table 238
PIO1_15	0x0A0	no	no	yes	no	Table 239
PIO1_16	0x0B0	no	no	yes	no	Table 243
PIO1_17	0x0B4	no	no	yes	no	Table 244
PIO1_18	0x0BC	no	no	yes	no	Table 246
PIO1_19	0x0C0	no	no	yes	no	Table 247
PIO1_20	0x0D0	no	no	yes	no	Table 251
PIO1_21	0x0D4	no	no	yes	no	Table 252

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

11.5.1 PIO0_17 register

Table 201. 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).	FAIM value dependent
		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.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **153 of 530**

Table 201. PIO0_17 register (PIO0_17, address 0x4004 4000) bit description

Bit	Symbol	Value	Description	Reset value
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.	
16	DACMODE		DAC mode enable.	0
		0x0	Disable	
		0x1	Enable	
31:17	-	-	Reserved.	0

11.5.2 PIO0_13 register

Table 202. 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).	FAIM value dependent
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

Table 202. PIO0_13 register (PIO0_13, address 0x4004 4004) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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

11.5.3 PIO0_12 register

Table 203. 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).	FAIM value dependent
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **155 of 530**

Table 203. PIO0_12 register (PIO0_12, address 0x4004 4008) bit description …*continued*

Bit	Symbol	Value	Description	Reset value
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

11.5.4 PIO0_5 register

Table 204. 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).	FAIM value dependent
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **156 of 530**

Table 204. PIO0_5 register (PIO0_5, address 0x4004 400C) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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

11.5.5 PIO0_4 register

Table 205. 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).	FAIM value dependent
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

UM11029
© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **157 of 530**

Table 205. PIO0_4 register (PIO0_4, address 0x4004 4010) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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

11.5.6 PIO0_3 register

Table 206. 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).	FAIM value dependent
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

Table 206. PIO0_3 register (PIO0_3, address 0x4004 4014) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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

11.5.7 PIO0_2 register

Table 207. 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).	FAIM value dependent
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **159 of 530**

Table 207. PIO0_2 register (PIO0_2, address 0x4004 4018) bit description

Bit	Symbol	Value	Description	Reset value
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

11.5.8 PIO0_11 register

Table 208. 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

Table 208. PIO0_11 register (PIO0_11, address 0x4004 401C) bit description

Bit	Symbol	Value	Description	Reset value
9:8	I2CMODE		Selects I2C mode.	00
			Select Standard mode (I2CMODE = 00, default) or Standard I/O functionality (I2CMODE = 01) if the pin function is GPIO.	
		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.	
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.	-

11.5.9 PIO0_10 register

Table 209. 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.	00
			Select Standard mode (I2CMODE = 00, default) or Standard I/O functionality (I2CMODE = 01) if the pin function is GPIO.	
		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.	

Table 209. PIO0_10 register (PIO0_10, address 0x4004 4020) bit description …*continued*

Bit	Symbol	Value	Description	Reset value
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.	
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.	-

11.5.10 PIO0_16 register

Table 210. 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).	FAIM value dependent
		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 210. PIO0_16 register (PIO0_16, address 0x4004 4024) 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

11.5.11 PIO0_15 register

Table 211. 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).	FAIM value dependent
		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.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **163 of 530**

Table 211. PIO0_15 register (PIO0_15, address 0x4004 4028) 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

11.5.12 PIO0_1 register

Table 212. 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).	FAIM value dependent
		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 212. PIO0_1 register (PIO0_1, address 0x4004 402C) 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

11.5.13 PIO0_9 register

Table 213. 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).	FAIM value dependent
		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.	

Table 213. PIO0_9 register (PIO0_9, address 0x4004 4034) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
		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

11.5.14 PIO0_8 register

Table 214. 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).	FAIM value dependent
		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 214. PIO0_8 register (PIO0_8, address 0x4004 4038) 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

11.5.15 PIO0_7 register

Table 215. 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).	FAIM value dependent
		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 215. PIO0_7 register (PIO0_7, address 0x4004 403C) 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

11.5.16 PIO0_6 register

Table 216. 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).	FAIM value dependent
		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.	

Table 216. PIO0_6 register (PIO0_6, address 0x4004 4040) 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

11.5.17 PIO0_0 register

Table 217. 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).	FAIM value dependent
		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 217. PIO0_0 register (PIO0_0, address 0x4004 4044) 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

11.5.18 PIO0_14 register

Table 218. 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).	FAIM value dependent
		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 218. PIO0_14 register (PIO0_14, address 0x4004 4048) 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

11.5.19 PIO0_28 register

Table 219. 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).	FAIM value dependent
		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.	

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **171 of 530**

Table 219. PIO0_28 register (PIO0_28, address 0x4004 4050) 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

11.5.20 PIO0_27 register

Table 220. 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).	FAIM value dependent
		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.	

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **172 of 530**

Table 220. PIO0_27 register (PIO0_27, address 0x4004 4054) 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

11.5.21 PIO0_26 register

Table 221. 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).	FAIM value dependent
		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 221. PIO0_26 register (PIO0_26, address 0x4004 4058) 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

11.5.22 PIO0_25 register

Table 222. 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).	FAIM value dependent
		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 222. PIO0_25 register (PIO0_25, address 0x4004 405C) 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

11.5.23 PIO0_24 register

Table 223. 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).	FAIM value dependent
		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.	

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **175 of 530**

Table 223. PIO0_24 register (PIO0_24, address 0x4004 4060) 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

11.5.24 PIO0_23 register

Table 224. 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).	FAIM value dependent
		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.	

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **176 of 530**

Table 224. PIO0_23 register (PIO0_23, address 0x4004 4064) 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

11.5.25 PIO0_22 register

Table 225. 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).	FAIM value dependent
		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 225. PIO0_22 register (PIO0_22, address 0x4004 4068) 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

11.5.26 PIO0_21 register

Table 226. 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).	FAIM value dependent
		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 226. PIO0_21 register (PIO0_21, address 0x4004 406C) 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

11.5.27 PIO0_20 register

Table 227. 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).	FAIM value dependent
		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 227. PIO0_20 register (PIO0_20, address 0x4004 4070) 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

11.5.28 PIO0_19 register

Table 228. 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).	FAIM value dependent
		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 228. PIO0_19 register (PIO0_19, address 0x4004 4074) 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

11.5.29 PIO0_18 register

Table 229. 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).	FAIM value dependent
		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 229. PIO0_18 register (PIO0_18, address 0x4004 4078) 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

11.5.30 PIO1_8 register

Table 230. 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).	FAIM value dependent
		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 230. PIO1_8 register (PIO1_8, address 0x4004 407C) 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

11.5.31 PIO1_9 register

Table 231. 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).	FAIM value dependent
		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 231. PIO1_9 register (PIO1_9, address 0x4004 4080) 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

11.5.32 PIO1_12 register

Table 232. 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).	FAIM value dependent
		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.	

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual

Rev. 1.6 — 8 December 2017

184 of 530

Table 232. PIO1_12 register (PIO1_12, address 0x4004 4084) 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

11.5.33 PIO1_13 register

Table 233. 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).	FAIM value dependent
		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 233. PIO1_13 register (PIO1_13, address 0x4004 4088) 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

11.5.34 PIO0_31 register

Table 234. 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).	FAIM value dependent
		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 234. PIO0_31 register (PIO0_31, address 0x4004 408C) 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

11.5.35 PIO1_0 register

Table 235. 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).	FAIM value dependent
		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 235. PIO1_0 register (PIO1_0, address 0x4004 4090) 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

11.5.36 PIO1_1 register

Table 236. 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).	FAIM value dependent
		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 236. PIO1_1 register (PIO1_1, address 0x4004 4094) 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

11.5.37 PIO1_2 register

Table 237. 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).	FAIM value dependent
		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 237. PIO1_2 register (PIO1_2, address 0x4004 4098) 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

11.5.38 PIO1_14 register

Table 238. 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).	FAIM value dependent
		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 238. PIO1_14 register (PIO1_14, address 0x4004 409C) 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

11.5.39 PIO1_15 register

Table 239. 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).	FAIM value dependent
		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 239. PIO1_15 register (PIO1_15, address 0x4004 40A0) 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

11.5.40 PIO1_3 register

Table 240. 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).	FAIM value dependent
		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 240. PIO1_3 register (PIO1_3, address 0x4004 40A4) 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

11.5.41 PIO1_4 register

Table 241. 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).	FAIM value dependent
		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 241. PIO1_4 register (PIO1_4, address 0x4004 40A8) 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

11.5.42 PIO1_5 register

Table 242. 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).	FAIM value dependent
		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 242. PIO1_5 register (PIO1_5, address 0x4004 40AC) 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

11.5.43 PIO1_16 register

Table 243. 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).	FAIM value dependent
		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 243. PIO1_16 register (PIO1_16, address 0x4004 40B0) 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

11.5.44 PIO1_17 register

Table 244. 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).	FAIM value dependent
		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 244. PIO1_17 register (PIO1_17, address 0x4004 40B4) 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

11.5.45 PIO1_6 register

Table 245. 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).	FAIM value dependent
		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 245. PIO1_6 register (PIO1_6, address 0x4004 40B8) 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

11.5.46 PIO1_18 register

Table 246. 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).	FAIM value dependent
		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 246. PIO1_18 register (PIO1_18, address 0x4004 40BC) 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

11.5.47 PIO1_19 register

Table 247. 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).	FAIM value dependent
		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 247. PIO1_19 register (PIO1_19, address 0x4004 40C0) 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

11.5.48 PIO1_7 register

Table 248. 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).	FAIM value dependent
		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 248. PIO1_7 register (PIO1_7, address 0x4004 40C4) 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

11.5.49 PIO0_29 register

Table 249. 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).	FAIM value dependent
		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 249. PIO0_29 register (PIO0_29, address 0x4004 40C8) 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

11.5.50 PIO1_30 register

Table 250. 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).	FAIM value dependent
		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 250. PIO0_30 register (PIO0_30, address 0x4004 40CC) 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

11.5.51 PIO1_20 register

Table 251. 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).	FAIM value dependent
		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.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **203 of 530**

Table 251. PIO1_20 register (PIO1_20, address 0x4004 40D0) 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

11.5.52 PIO1_21 register

Table 252. 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).	FAIM value dependent
		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 252. PIO1_21 register (PIO1_21, address 0x4004 40D4) 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

11.5.53 PIO1_11 register

Table 253. 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).	FAIM value dependent
		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 253. PIO1_11 register (PIO1_11, address 0x4004 40D8) 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

11.5.54 PIO1_10 register

Table 254. 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).	FAIM value dependent
		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 254. PIO1_10 register (PIO1_10, address 0x4004 40DC) 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

UM11029

User manual   **Rev. 1.6 — 8 December 2017**   **207 of 530**

12.1 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 11 "LPC84x I/O Configuration (IOCON)" for more details. Table 255 shows the GPIO pins that are available for each package type.

Table 255. GPIO pins available

Package	GPIO Port 0	GPIO Port 1
HVQFN33	PIO0_0 to PIO0_28	N/A
HVQFN48	PIO0_0 to PIO0_31	PIO1_0 to PIO1_9
LQFP48	PIO0_0 to PIO0_31	PIO1_0 to PIO1_9
LQFP64	PIO0_0 to PIO0_31	PIO1_0 to PIO1_21

12.2 Basic configuration

For the GPIO port registers, enable the clock to the GPIO port in the SYSAHBCLKCTRL register (Table 147).

12.3 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.

12.4 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.

12.5 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 256. 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 257
B32 to B53	R/W	0x0020 to 0x0035	Byte pin registers port1; PIO1_0 to PIO1_21	ext	byte (8 bit)	Table 257
-	-	0x0036 to 0x003F	Reserved	-	-	-
W0 to W31	R/W	0x1000 to 0x107C	Word pin registers port 0	ext	word (32 bit)	Table 258
W32 to W53	R/W	0x1080 to 0x10D4	Word pin registers port 1; PIO1_0 to PIO1_21	ext	word (32 bit)	Table 258
DIR0	R/W	0x2000	Direction registers port 0	0	word (32 bit)	Table 259
DIR1	R/W	0x2004	Direction registers port 1	0	word (32 bit)	Table 259
MASK0	R/W	0x2080	Mask register port 0	0	word (32 bit)	Table 260
MASK1	R/W	0x2084	Mask register port 1	0	word (32 bit)	Table 260
PIN0	R/W	0x2100	Port pin register port 0	ext	word (32 bit)	Table 261
PIN1	R/W	0x2104	Port pin register port 1	ext	word (32 bit)	Table 261
MPIN0	R/W	0x2180	Masked port register port 0	ext	word (32 bit)	Table 262
SET0	R/W	0x2200	Write: Set register for port 0 Read: output bits for port 0	0	word (32 bit)	Table 263
SET1	R/W	0x2204	Write: Set register for port 1 Read: output bits for port 1	0	word (32 bit)	Table 263
CLR0	WO	0x2280	Clear port 0	-	word (32 bit)	Table 264
CLR1	WO	0x2284	Clear port 1	-	word (32 bit)	Table 264
NOT0	WO	0x2300	Toggle port 0	-	word (32 bit)	Table 265
NOT1	WO	0x2304	Toggle port 1	-	word (32 bit)	Table 265
DIRSET0	WO	0x2380	Set pin direction bits for port 0.	0	word (32 bit)	Table 266
DIRSET1	WO	0x2384	Set pin direction bits for port 1.	0	word (32 bit)	Table 266
DIRCLR0	WO	0x2400	Clear pin direction bits for port 0.	-	word (32 bit)	Table 267
DIRCLR1	WO	0x2404	Clear pin direction bits for port 1.	-	word (32 bit)	Table 267
DIRNOT0	WO	0x2480	Toggle pin direction bits for port 0.	-	word (32 bit)	Table 268
DIRNOT1	WO	0x2484	Toggle pin direction bits for port 1.	-	word (32 bit)	Table 268

12.5.1 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 257. 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	-

12.5.2 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 258. 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

12.5.3 GPIO port direction registers

Each GPIO port has one direction register for configuring the port pins as inputs or outputs.

Table 259. 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

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **210 of 530**

12.5.4 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 260. 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

12.5.5 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 261. 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

12.5.6 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 262. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **211 of 530**

12.5.7 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 263. 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

12.5.8 GPIO port clear registers

Output bits can be cleared by writing ones to these write-only registers, regardless of MASK registers.

Table 264. 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

12.5.9 GPIO port toggle registers

Output bits can be set by writing ones to these write-only registers, regardless of MASK registers.

Table 265. 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

12.5.10 GPIO port direction set registers

Direction bits can be set by writing ones to these registers.

Table 266. 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

12.5.11 GPIO port direction clear registers

Direction bits can be cleared by writing ones to these write-only registers.

Table 267. 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

12.5.12 GPIO port direction toggle registers

Direction bits can be set by writing ones to these write-only registers.

Table 268. 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

12.6 Functional description

12.6.1 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.

12.6.2 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 12.6.1 returns the state of pins, regardless of their direction or alternate functions.

12.6.3 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.

12.6.4 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.

12.6.5 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.

13.1 How to read this chapter

The pin interrupt generator and the pattern match engine are available on all LPC84x parts.

13.2 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.

13.3 Basic configuration

- Pin interrupts:
    - Select up to eight external interrupt pins from all GPIO port pins in the SYSCON block (Table 168). 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 147, 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 169).
    - 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 168). 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 147, 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 192).

13.3.1 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 LPC84x 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 8.6.43 "Pin interrupt select registers" in the SYSCON block for the PINTSEL registers.

13.4 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 8.6.43 "Pin interrupt select registers".

The pattern match engine output is assigned to an external pin through the switch matrix.

See Section 10.3.1 "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 269. 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 192

13.5 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **217 of 530**

13.5.1 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 168). 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.
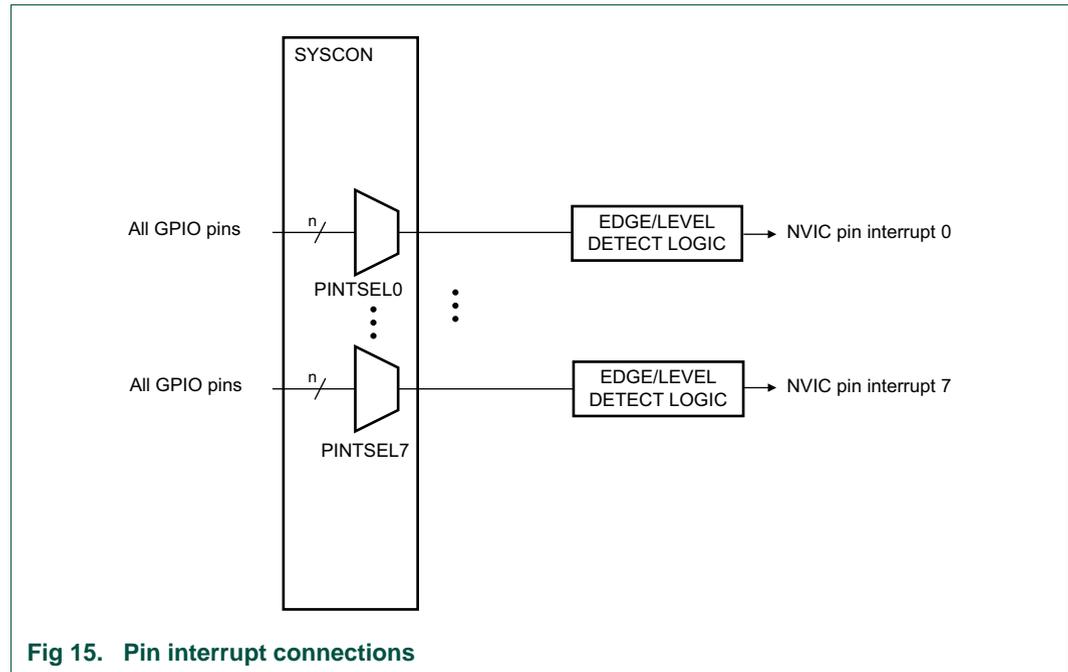
Fig 15.   Pin interrupt connections

13.5.2 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 17 for the detect logic block.

Fig 16.   Pattern match engine connections
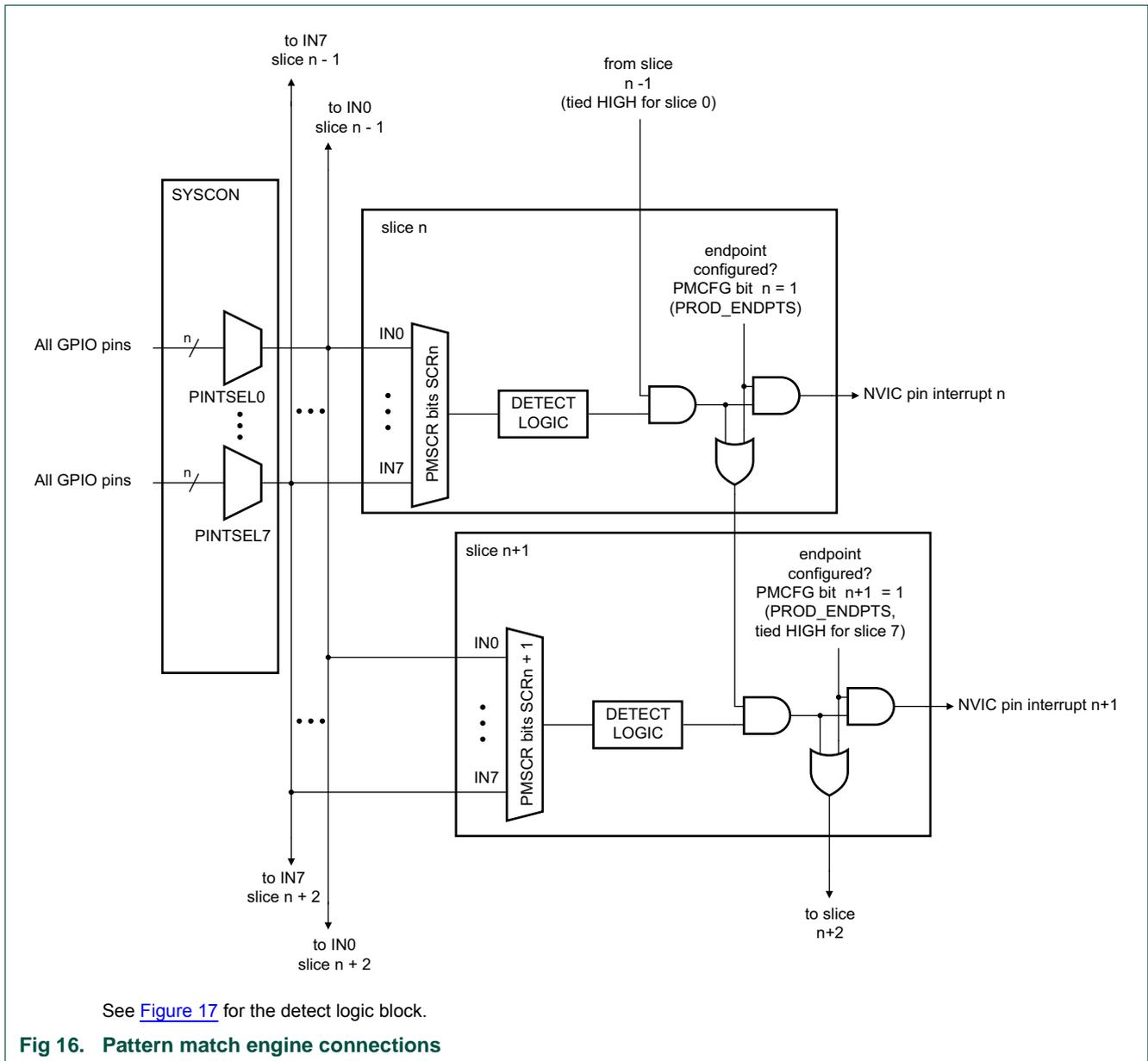
The detect logic of each slice can detect the following events on the selected input:

- 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 17 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 13.7.3 for details.
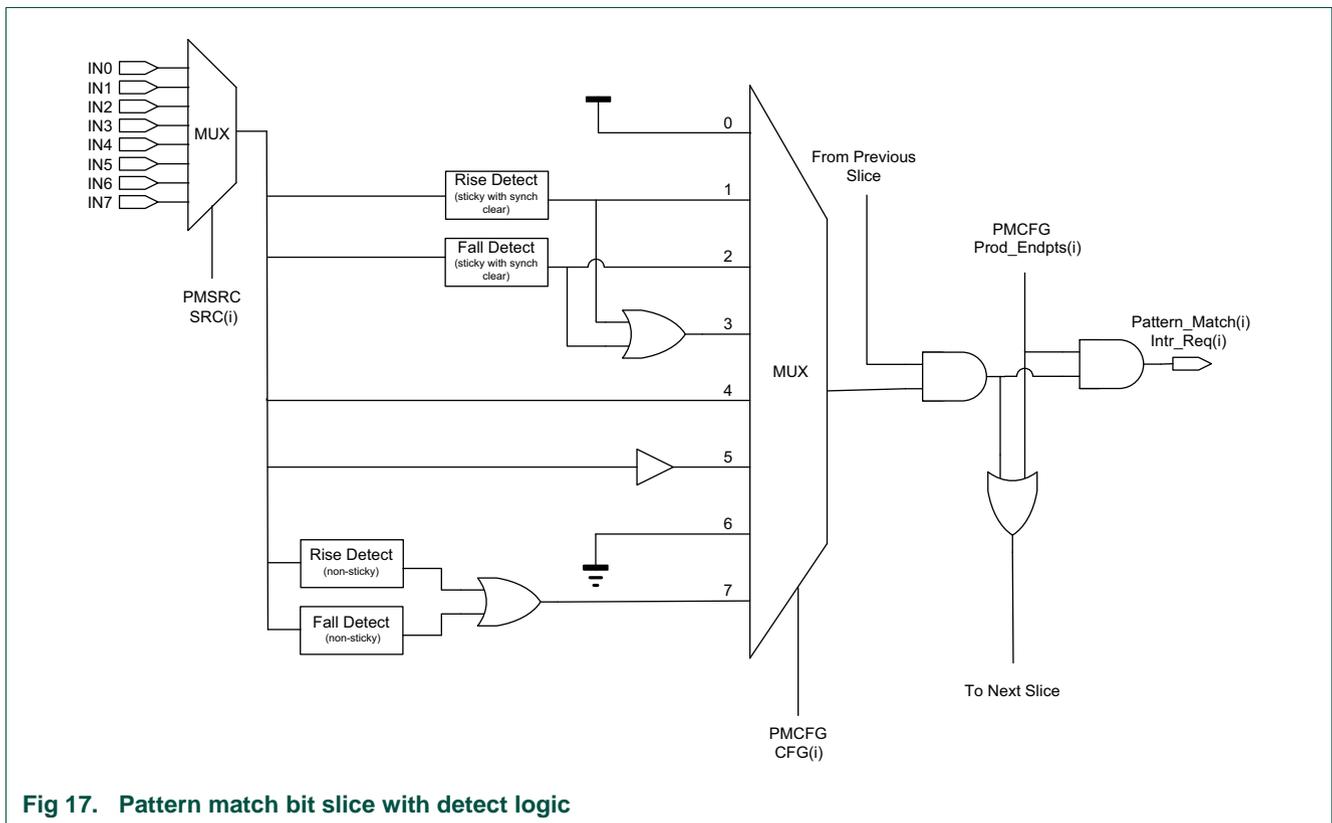
Fig 17. Pattern match bit slice with detect logic

13.5.2.1 Inputs and outputs of the pattern match engine

The connections between the pins and the pattern match engine are shown in Figure 16. 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **220 of 530**

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.

13.5.2.2 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 17).

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 282) and PMCFG (Table 283). 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.

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **221 of 530**

- 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 13.7.2

13.6 Register description

Table 270. 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 271
IENR	R/W	0x004	Pin interrupt level or rising edge interrupt enable register	0	Table 272
SIENR	WO	0x008	Pin interrupt level or rising edge interrupt set register	NA	Table 273
CIENR	WO	0x00C	Pin interrupt level (rising edge interrupt) clear register	NA	Table 274
IENF	R/W	0x010	Pin interrupt active level or falling edge interrupt enable register	0	Table 275
SIENF	WO	0x014	Pin interrupt active level or falling edge interrupt set register	NA	Table 276
CIENF	WO	0x018	Pin interrupt active level or falling edge interrupt clear register	NA	Table 277
RISE	R/W	0x01C	Pin interrupt rising edge register	0	Table 278
FALL	R/W	0x020	Pin interrupt falling edge register	0	Table 279
IST	R/W	0x024	Pin interrupt status register	0	Table 280
PMCTRL	R/W	0x028	Pattern match interrupt control register	0	Table 281
PMSRC	R/W	0x02C	Pattern match interrupt bit-slice source register	0	Table 282
PMCFG	R/W	0x030	Pattern match interrupt bit slice configuration register	0	Table 283

13.6.1 Pin interrupt mode register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), one bit in the ISEL register determines whether the interrupt is edge or level sensitive.

Table 271. 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.	-	-

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **222 of 530**

13.6.2 Pin interrupt level or rising edge interrupt enable register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), 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 272. 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.	-	-

13.6.3 Pin interrupt level or rising edge interrupt set register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), 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 273. 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.	-	-

13.6.4 Pin interrupt level or rising edge interrupt clear register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), 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.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **223 of 530**

Table 274. 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.	-	-

13.6.5 Pin interrupt active level or falling edge interrupt enable register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), 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 275.   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.	-	-

13.6.6 Pin interrupt active level or falling edge interrupt set register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), 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 276. 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.	-	-

13.6.7 Pin interrupt active level or falling edge interrupt clear register

For each of the 8 pin interrupts selected in the PINTSELn registers (see Section 8.6.43), 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 277. 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.	-	-

13.6.8 Pin interrupt rising edge register

This register contains ones for pin interrupts selected in the PINTSELn registers (see Section 8.6.43) 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 278. 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.	-	-

13.6.9 Pin interrupt falling edge register

This register contains ones for pin interrupts selected in the PINTSELn registers (see Section 8.6.43) 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 279. 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.	-	-

13.6.10 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 280. 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.	-	-

13.6.11 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.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **226 of 530**

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 281. 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

13.6.12 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 8.6.43. 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 282. 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 282. 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 282. 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 282. 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 282. 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.	

13.6.13 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 283. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description

Bit	Symbol	Value	Description	Reset value
0	PROD_EN DPTS0		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_EN DPTS1		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_EN DPTS2		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_EN DPTS3		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_EN DPTS4		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 283. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
5	PROD_EN DPTS5		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_EN DPTS6		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 283. 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 283. 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 283. 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 283. 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.	

13.7 Functional description

13.7.1 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 284.

Table 284. 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.

13.7.2 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 282):
  - 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 283):
  - 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 281):

- 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.

13.7.3 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 18. Pattern match engine examples: sticky edge detect

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 true

Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

Fig 20. Pattern match engine examples: Windowed non-sticky edge detect evaluates as false

UM11029

User manual **Rev. 1.6 — 8 December 2017** **240 of 530**

14.1 How to read this chapter

SCT input multiplexing and DMA input multiplexing is available for all parts.

14.2 Features

- Configures the inputs to the SCTimer/PWM.
- Configures the inputs to the DMA triggers.

14.3 Basic configuration

- In the SYSAHBCLKCTRL register, enable the SCT clock to write to the INPUT MUX registers. See Table 147.
- In the SYSAHBCLKCTRL register, enable the DMA clock to write to the DMA TRIGMUX registers. See Table 147.

14.4 Pin description

The input multiplexer has no dedicated pins. External pins can be selected as inputs to the SCT and DMA triggers. Multiplexer inputs from external pins are assigned through the switch matrix to pins.

Table 285. INPUT MUX pin description

Pin functions	Peripheral	Input mux reference
SCT_PIN0, SCT_PIN1, SCT_PIN2, SCT_PIN3	SCT0	Table 288

14.5 General description

The inputs to the SCTimer/PWM and 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 SCTimer/PWM and the ADC or the SCTimer/PWM and the analog comparator.

The DMA can use trigger input multiplexing to sequence DMA transactions without the use of interrupt service routines.

14.5.1 SCT input multiplexing
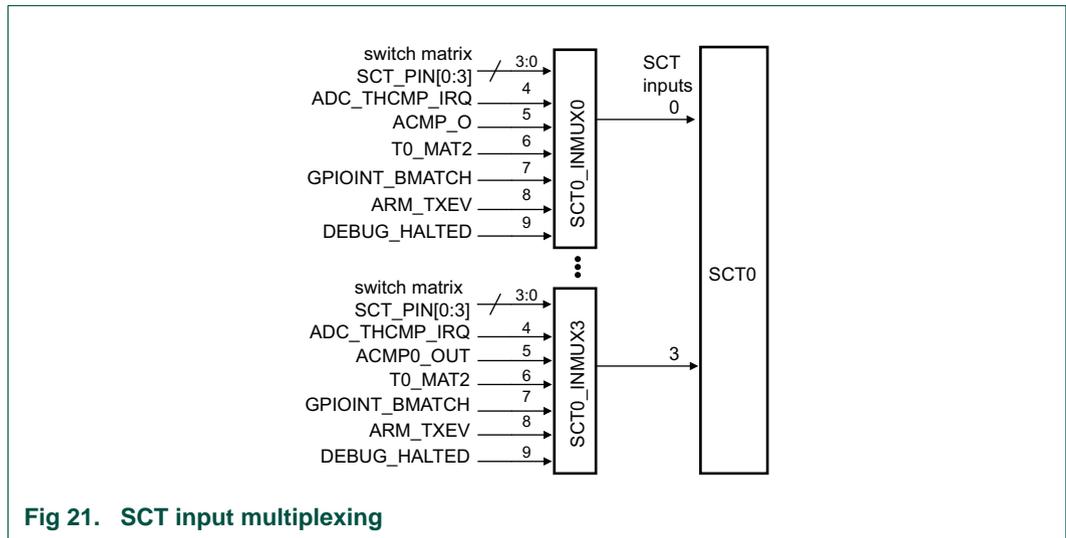
Fig 21.   SCT input multiplexing

14.5.2 DMA trigger input multiplexing
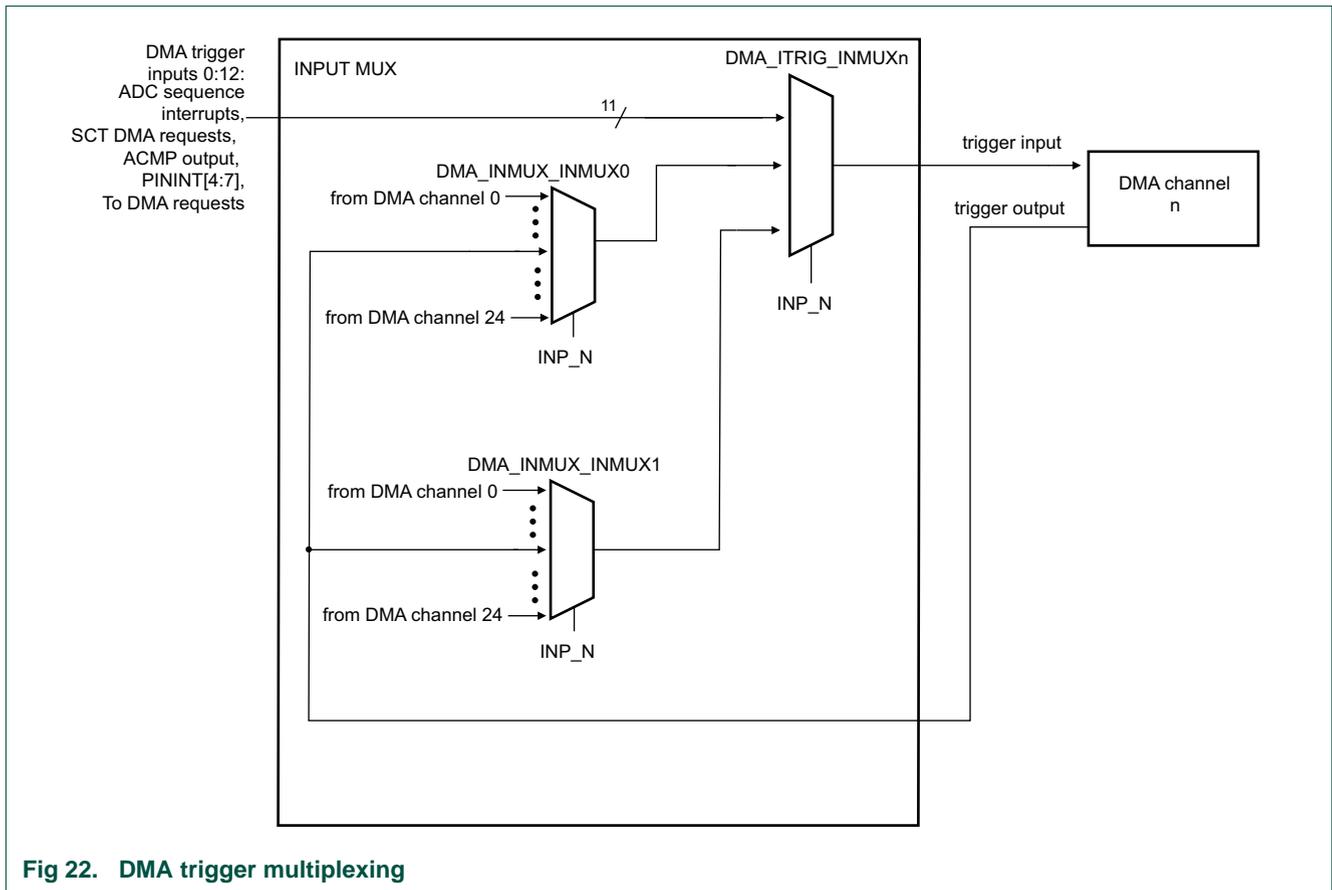
Fig 22.   DMA trigger multiplexing

14.6 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 286 are reserved and should not be written to.

Table 286. 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 25 DMA trigger outputs.	0x1F	Table 287
DMA_INMUX_INMUX1	R/W	0x004	Input mux register for DMA trigger input 12. Selects from 25 DMA trigger outputs.	0x1F	Table 287
SCT0_INMUX0	R/W	0x020	Input mux register for SCT input 0.	0x0F	Table 288
SCT0_INMUX1	R/W	0x024	Input mux register for SCT input 1.	0x0F	Table 288
SCT0_INMUX2	R/W	0x028	Input mux register for SCT input 2.	0x0F	Table 288
SCT0_INMUX3	R/W	0x02C	Input mux register for SCT input 3.	0x0F	Table 288
DMA_ITRIG_INMUX0	R/W	0x040	Input mux register for trigger inputs 0 to 12 connected to DMA channel 0. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX1	R/W	0x044	Input mux register for trigger inputs 0 to 12 connected to DMA channel 1. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX2	R/W	0x048	Input mux register for trigger inputs 0 to 12 connected to DMA channel 2. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX3	R/W	0x04C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 3. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX4	R/W	0x050	Input mux register for trigger inputs 0 to 12 connected to DMA channel 4. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX5	R/W	0x054	Input mux register for trigger inputs 0 to 12 connected to DMA channel 5. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX6	R/W	0x058	Input mux register for trigger inputs 0 to 12 connected to DMA channel 6. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX7	R/W	0x05C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 7. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX8	R/W	0x060	Input mux register for trigger inputs 0 to 12 connected to DMA channel 8. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX9	R/W	0x064	Input mux register for trigger inputs 0 to 12 connected to DMA channel 9. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX10	R/W	0x068	Input mux register for trigger inputs 0 to 12 connected to DMA channel 10. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289

UM11029

User manual **Rev. 1.6 — 8 December 2017** **243 of 530**

Table 286. Register overview: Input multiplexing (base address 0x4002 C000) *…continued*

Name	Access	Offset	Description	Reset value	Reference
DMA_ITRIG_INMUX11	R/W	0x06C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 11. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX12	R/W	0x070	Input mux register for trigger inputs 0 to 12 connected to DMA channel 12. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX13	R/W	0x074	Input mux register for trigger inputs 0 to 12 connected to DMA channel 13. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX14	R/W	0x078	Input mux register for trigger inputs 0 to 12 connected to DMA channel 14. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX15	R/W	0x07C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 15. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX16	R/W	0x080	Input mux register for trigger inputs 0 to 12 connected to DMA channel 16. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX17	R/W	0x084	Input mux register for trigger inputs 0 to 12 connected to DMA channel 17. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX18	R/W	0x088	Input mux register for trigger inputs 0 to 12 connected to DMA channel 18. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32,and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX19	R/W	0x08C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 19. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX20	R/W	0x090	Input mux register for trigger inputs 0 to 12 connected to DMA channel 20. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX21	R/W	0x094	Input mux register for trigger inputs 0 to 12 connected to DMA channel 21. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX22	R/W	0x098	Input mux register for trigger inputs 0 to 12 connected to DMA channel 22. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX23	R/W	0x09C	Input mux register for trigger inputs 0 to 12 connected to DMA channel 23. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests.	0x0F	Table 289
DMA_ITRIG_INMUX24	R/W	0x0A0	Input mux register for trigger inputs 0 to 12 connected to DMA channel 24. Selects from ADC, SCT, ACMP, pin interrupts, CTIMER32, and DMA requests	0x0F	Table 289

14.6.1 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 25 trigger outputs generated by the DMA (one trigger output per channel).

By default, none of the triggers are selected.

Table 287. 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 24).	0x1F
31:5	-	Reserved.	-

14.6.2 SCT input mux registers 0 to 3

With the SCT0 Input mux registers you can select one input source for each SCT input from 10 external and internal sources.

The output of SCT Input mux register 0 selects the source for SCT0 input 0, the output of SCT0 Input mux register 1 selects the source for SCT0 input 1, and so forth up to SCT0 Input mux register 3, which selects the input for SCT0 input 3.

The value to be programmed in this register is the input number ranging from 0 for pin function SCT_IN0 to 9 for the DEBUG_HALTED signal from the ARM CoreSight debug signal.

Inputs 0 to 3 are connected to external pins through the switch matrix.

Table 288. SCT input mux registers 0 to 3 (SCT0_INMUX[0:3], address 0x4002 C020 (SCT0_INMUX0) to 0x4002 C02C (SCT0_INMUX3)) bit description

Bit	Symbol	Value	Description	Reset value
3:0	INP_N		Input number (decimal value) to SCT0 inputs 0 to 3.	0x0F
		0x0	SCT_PIN0. Assign to pin using the switch matrix.	
		0x1	SCT_PIN1. Assign to pin using the switch matrix.	
		0x2	SCT_PIN2. Assign to pin using the switch matrix.	
		0x3	SCT_PIN3. Assign to pin using the switch matrix.	
		0x4	ADC_THCMP_IRQ	
		0x5	ACMP_O	
		0x6	T0_MAT2	
		0x7	GPIOINT_BMATCH	
		0x8	ARM_TXEV	
		0x9	DEBUG_HALTED	
31:4	-		Reserved.	-

14.6.3 DMA input trigger input mux registers 0 to 24

With the DMA input trigger input mux registers you can select one trigger input for each of the 25 DMA channels from multiple internal sources.

By default, none of the triggers are selected.

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **245 of 530**

Table 289. DMA input trigger Input mux registers 0 to 24 (DMA_ITRIG_INMUX[0:24], address 0x4002 C040 (DMA_ITRIG_INMUX0) to 0x4002 C0A0 (DMA_ITRIG_INMUX24)) 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	ADC_SEQA_IRQ	
		0x1	ADC_SEQB_IRQ	
		0x2	SCT_DMA0	
		0x3	SCT_DMA1	
		0x4	ACMP_O	
		0x5	PININT4	
		0x6	PININT5	
		0x7	PININT6	
		0x8	PININT7	
		0x9	T0_DMAREQ_M0	
		0xA	T0_DMAREQ_M1	
		0xB	DMA trigger mux 0. (DMA_INMUX_INMUX0)	
		0xC	DMA trigger mux 1. (DMA_INMUX_INMUX1)	
31:4	-		Reserved.	-

UM11029

User manual **Rev. 1.6 — 8 December 2017** **246 of 530**

15.1 How to read this chapter

The LPC84x provides an on-chip API in the boot ROM to optimize power consumption in active and sleep modes. See Section 16.1.

Read this chapter to configure the reduced power modes deep-sleep mode, power-down mode, and deep power-down mode.

15.2 Features

- Reduced power modes control
- Low-power oscillator control
- Five general purpose backup registers to retain data in deep power-down mode

15.3 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 295.

If using the WKTCLKIN function, disable the hysteresis for that pin in the DPDCTRL register. See Table 295.

15.3.1 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.

15.3.1.1 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 293).

Table 290. 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

15.4 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 15.6.3 "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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **248 of 530**

15.5 General description

Power on the LPC84x 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.

15.5.1 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 172 "Wake-up configuration register (PDAWAKECFG, address 0x4004 8234) bit description".
- To use external interrupts for wake-up: Table 169 "Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description" and Table 168 "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 170 "Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description"
- To configure the USART to wake up the part: Section 17.3.2 "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 291 "Wake-up sources for reduced power modes".

Table 291. Wake-up sources for reduced power modes

power mode	Wake-up source	Conditions
Sleep	Any interrupt	Enable interrupt in NVIC.
	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 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 Capacitive Touch peripheral	• Enable interrupt in NVIC and STARTERP1 registers. • Enable the Capacitive Touch interrupt. • Switch FCLK clock source to the WDOsc. • Set Capacitive Touch registers. • Provide a touch event to the peripheral.
	RESET pin PIO0_5	Enable the reset function in the PINENABLE0 register via switch matrix.
Deep power-down	WAKEUP pin PIO0_4	Enable the WAKEUP function in the DPDCTRL register in the PMU.
	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 low-power oscillator in the DPDCTRL register in the PMU. • Enable the 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.

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **250 of 530**

15.6 Register description

Table 292. Register overview: PMU (base address 0x4002 0000)

Name	Access	Address offset	Description	Reset value	Reference
PCON	R/W	0x000	Power control register	0x0	Table 293
GPREG0	R/W	0x004	General purpose register 0	0x0	Table 294
GPREG1	R/W	0x008	General purpose register 1	0x0	Table 294
GPREG2	R/W	0x00C	General purpose register 2	0x0	Table 294
GPREG3	R/W	0x010	General purpose register 3	0x0	Table 294
DPDCTRL	R/W	0x014	Deep power-down control register. Also includes bits for general purpose storage.	0x0	Table 295

15.6.1 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 293. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **251 of 530**

Table 293. 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

15.6.2 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 294. 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

15.6.3 Deep power-down control register

The deep power-down control register controls the 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 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 295. 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.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **252 of 530**

Table 295. 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	LPOSCEN		Enable the 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 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	LPOSCDPDEN		Enable the low-power oscillator in deep power-down mode. Setting this bit causes the 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 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	-		Data retained during deep power-down mode.	0x0

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **253 of 530**

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.

15.7 Functional description

15.7.1 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 296. 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/low-power oscillator	software configurable	software configurable	software configurable	software configurable
ADC	software configurable	off	off	off
DAC0/1	software configurable	off	off	off
Capacitive Touch	software configurable	software configurable	software configurable	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.

15.7.2 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".

15.7.3 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **255 of 530**

15.7.3.1 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 147).
- The power to various analog blocks (PLL, oscillators, the BOD circuit, the ADC block, the comparator block, the FAIM block, and the flash block) can be controlled at any time individually through the PDRUNCFG register (Table 173 "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 127) and the SYSAHBCLKDIV register (Table 140).
- The USART, ADC, SCTimer/PWM, and CLKOUT use individual peripheral clocks with their own clock dividers. The peripheral clocks can be shut down through the corresponding clock divider registers.

15.7.4 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 remain static.

15.7.4.1 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.

15.7.4.2 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 290).
3. Use the ARM Cortex-M0+ Wait-For-Interrupt (WFI) instruction.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **256 of 530**

15.7.4.3 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.

15.7.5 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 watchdog 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 watchdog 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.

15.7.5.1 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 171) 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.

15.7.5.2 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 293).
2. Select the power configuration in deep-sleep mode in the PDSLEEPCFG (Table 171) register.
3. Select the power configuration after wake-up in the PDAWAKECFG (Table 172) 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 169, Table 170) and in the NVIC.
5. Select the FRO as the main clock. See Table 137.
6. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register (Table 290).
7. Use the ARM WFI instruction.

15.7.5.3 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 168. Each pin interrupt must also be enabled in the STARTERP0 register (Table 169) 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 170). 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 164).

- WWDT signal, if the watchdog oscillator is enabled in the PDSLEEPCFG register:

– WWDT interrupt using the interrupt wake-up register 1 (Table 170). 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 watchdog 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 17.3.2 "Configure the USART for wake-up".
- Via the I2C. See Section 19.3.3.
- Via any of the SPI blocks. See Section 18.3.1.
- Via the Capacitive Touch block.

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).

15.7.6 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 watchdog 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 watchdog 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.

15.7.6.1 Power configuration in power-down mode

Power consumption in power-down mode can be configured by the power configuration setting in the PDSLEEPCFG (Table 171) register in the same way as for deep-sleep mode (see Section 15.7.5.1):

- The watchdog 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.

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **258 of 530**

15.7.6.2 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 293).
2. Select the power configuration in power-down mode in the PDSLEEPCFG (Table 171) register.
3. Select the power configuration after wake-up in the PDAWAKECFG (Table 172) 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 169, Table 170) and in the NVIC.
5. Select the FRO as the main clock. See Table 137.
6. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register (Table 290).
7. Use the ARM WFI instruction.

15.7.6.3 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 168. Each pin interrupt must also be enabled in the STARTERP0 register (Table 169) 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 170). 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 164).
- WWDT signal, if the watchdog oscillator is enabled in the PDSLEEPCFG register:
  - WWDT interrupt using the interrupt wake-up register 1 (Table 170). 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 17.3.2 "Configure the USART for wake-up".
  - Via the I2C. See Section 19.3.3.
  - Via any of the SPI blocks. See Section 18.3.1.
  - Via the Capacitive Touch block.

  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).

15.7.7 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 293) prevents the part from entering deep-power down mode.

15.7.7.1 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.

15.7.7.2 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 293) is cleared.
3. Write 0x3 to the PM bits in the PCON register (see Table 293).
4. Store data to be retained in the general purpose registers (Section 15.6.2).
5. Write one to the SLEEPDEEP bit in the ARM Cortex-M0+ SCR register (Table 290).
6. Use the ARM WFI instruction.

15.7.7.3 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 LPC84x 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 293) 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 293).
4. (Optional) Read the stored data in the general purpose registers (Section 15.6.2).
5. Set up the PMU for the next deep power-down cycle.

15.7.7.4 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 low-power oscillator to run in deep power-down mode by setting bits 2 and 3 in the DPDCTRL register to 1 (see Table 295)

2. Ensure that bit 3 in the PCON register (Table 293) is cleared.

3. Write 0x3 to the PM bits in the PCON register (see Table 293).

4. Store data to be retained in the general purpose registers (Section 15.6.2).

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 427).

7. Use the ARM WFI instruction.

15.7.7.5 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 293) 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 293).

4. (Optional) Read the stored data in the general purpose registers (Section 15.6.2).

5. Set up the PMU for the next deep power-down cycle.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **261 of 530**

16.1 How to read this chapter

The DMA controller is available on all parts.

16.2 Features

- 25 channels supported with 25 channels connected to peripheral request inputs and outputs of the USART, SPI, I^{2}C, DAC, and Capacitive Touch.
- 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.

16.3 Basic configuration

Configure the DMA as follows:

- Use the SYSAHBCLKCTRL register (Table 147) 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[0:3] as inputs to DMA triggers.

For details on the trigger input and output multiplexing, see Section 14.5.2 "DMA trigger input multiplexing".

16.3.1 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 14.5.2 "DMA trigger input multiplexing".

See Table 297 for the connection of input multiplexers to DMA channels.

See Table 289 for a list of possible trigger input sources.

16.3.2 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".

16.3.3 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.
- SCT DMA request 0 SCT_DMA0.
- SCT DMA request 1 SCT_DMA1.
- ACMP_O comparator output.
- GPIO pin interrupt 4 (PININT4).
- GPIO pin interrupt 5 (PININT5).
- GPIO pin interrupt 6 (PININT6).
- GPIO pin interrupt 7 (PININT7).
- CTIMER32 MR0 match DMA request (T0_DMAREQ_M0).
- CTIMER32 MR1 match DMA request (T0_DMAREQ_M1).
- Two choices of one of the DMA output triggers.

Table 297. DMA requests

DMA channel #	Request input	DMA trigger multiplexer
0	USART0_RX_DMA	DMA_ITRIG_INMUX0
1	USART0_TX_DMA	DMA_ITRIG_INMUX1
2	USART1_RX_DMA	DMA_ITRIG_INMUX2
3	USART1_TX_DMA	DMA_ITRIG_INMUX3
4	USART2_RX_DMA	DMA_ITRIG_INMUX4
5	USART2_TX_DMA	DMA_ITRIG_INMUX5
6	USART3_RX_DMA	DMA_ITRIG_INMUX6
7	USART3_TX_DMA	DMA_ITRIG_INMUX7
8	USART4_RX_DMA	DMA_ITRIG_INMUX8
9	USART4_TX_DMA	DMA_ITRIG_INMUX9
10	SPI0_RX_DMA	DMA_ITRIG_INMUX10
11	SPI0_TX_DMA	DMA_ITRIG_INMUX11
12	SPI1_RX_DMA	DMA_ITRIG_INMUX12

UM11029

User manual **Rev. 1.6 — 8 December 2017** **263 of 530**

Table 297. DMA requests

DMA channel #	Request input	DMA trigger multiplexer
13	SPI1_TX_DMA	DMA_ITRIG_INMUX13
14	I2C0_SLV_DMA	DMA_ITRIG_INMUX14
15	I2C0_MST_DMA	DMA_ITRIG_INMUX15
16	I2C1_SLV_DMA	DMA_ITRIG_INMUX16
17	I2C1_MST_DMA	DMA_ITRIG_INMUX17
18	I2C2_SLV_DMA	DMA_ITRIG_INMUX18
19	I2C2_MST_DMA	DMA_ITRIG_INMUX19
20	I2C3_SLV_DMA	DMA_ITRIG_INMUX20
21	I2C3_MST_DMA	DMA_ITRIG_INMUX21
22	DAC0_DMAREQ	DMA_ITRIG_INMUX22
23	DAC1_DMAREQ	DMA_ITRIG_INMUX23
24	CAPT_DMA	DMA_ITRIG_INMUX24

16.3.4 DMA in sleep mode

The DMA can operate and access all SRAM blocks in sleep mode.

16.4 Pin description

The DMA controller has no configurable pins.

16.5 General description
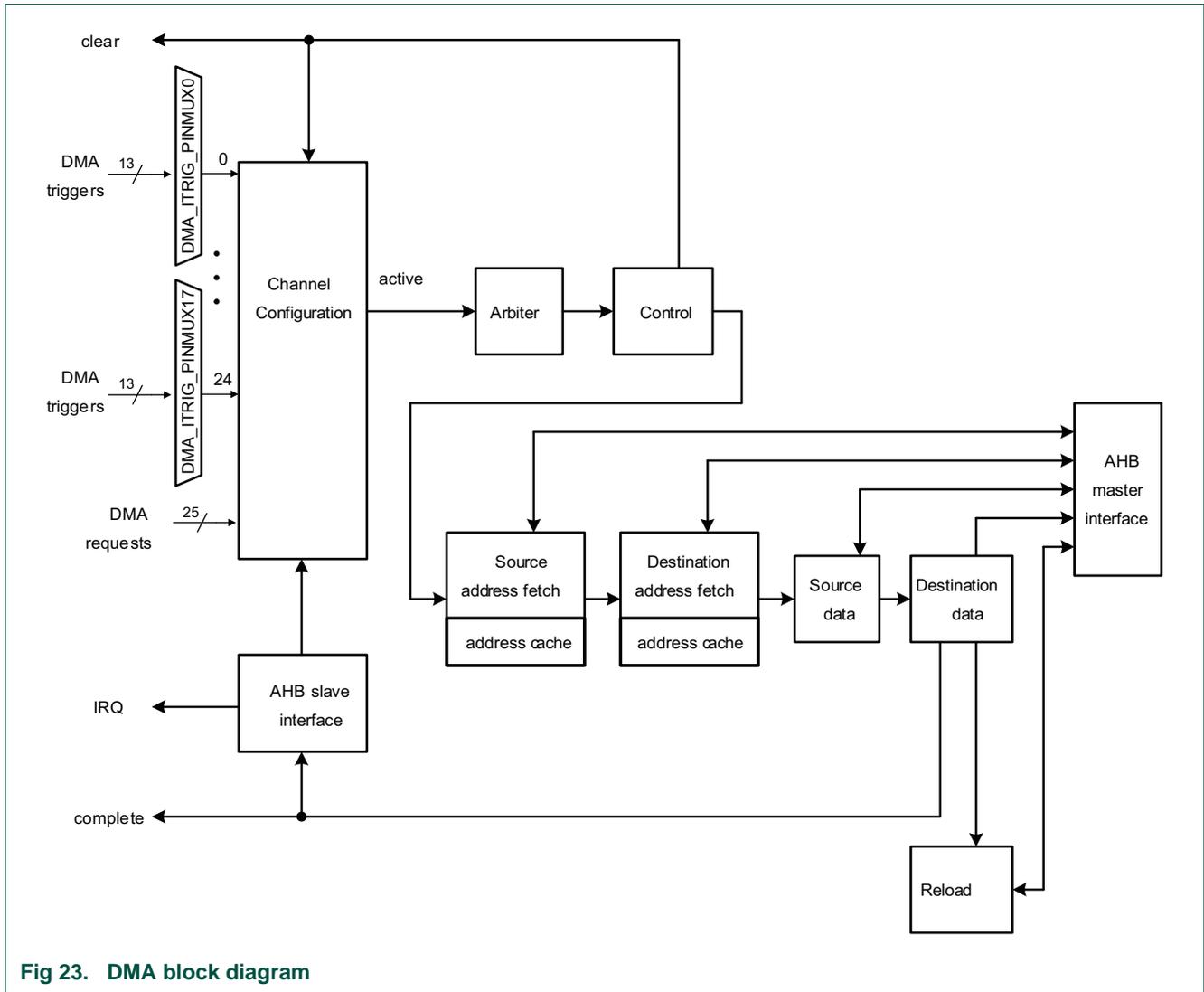
Fig 23. DMA block diagram

16.5.1 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, DAC, and Capacitive Touch.

A trigger initiates a DMA operation and can be a signal from an unrelated peripheral. Peripherals that generate triggers are the SCT, 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.

16.5.2 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 297), and supplying at least the channel descriptor, which is located somewhere in memory, typically in on-chip SRAM (see Section 16.6.3). The channel descriptor is shown in Table 298.

Table 298. 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 299. 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 299. 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).

16.5.3 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 300 is needed.

Table 300. 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.

16.5.4 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 301 shows an example of descriptors for ping-pong from a peripheral to two buffers in memory.

Table 301. 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.

16.5.5 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 301). 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.

16.5.6 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.

16.5.7 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.

16.6 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 302. Register overview: DMA controller (base address 0x5000 8000)

Name	Access	Address offset	Description	Reset Value	Reference
Global control and status registers					
CTRL	R/W	0x000	DMA control.	0	Table 303
INTSTAT	RO	0x004	Interrupt status.	0	Table 304
SRAMBASE	R/W	0x008	SRAM address of the channel configuration table.	0	Table 305

Table 302. Register overview: DMA controller (base address 0x5000 8000)

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 307
ENABLECLR0	W1	0x028	Channel Enable Clear for all DMA channels.	NA	Table 308
ACTIVE0	RO	0x030	Channel Active status for all DMA channels.	0	Table 309
BUSY0	RO	0x038	Channel Busy status for all DMA channels.	0	Table 310
ERRINT0	RO/W1	0x040	Error Interrupt status for all DMA channels.	0	Table 311
INTENSET0	RO/W1	0x048	Interrupt Enable read and Set for all DMA channels.	0	Table 312
INTENCLR0	W1	0x050	Interrupt Enable Clear for all DMA channels.	NA	Table 313
INTA0	RO/W1	0x058	Interrupt A status for all DMA channels.	0	Table 314
INTB0	RO/W1	0x060	Interrupt B status for all DMA channels.	0	Table 315
SETVALID0	W1	0x068	Set ValidPending control bits for all DMA channels.	NA	Table 316
SETTRIG0	W1	0x070	Set Trigger control bits for all DMA channels.	NA	Table 317
ABORT0	W1	0x078	Channel Abort control for all DMA channels.	NA	Table 318
Channel0 registers					
CFG0	R/W	0x400	Configuration register for DMA channel 0.		Table 319
CTLSTAT0	RO	0x404	Control and status register for DMA channel 0.		Table 321
XFERCFG0	R/W	0x408	Transfer configuration register for DMA channel 0.		Table 322
Channel1 registers					
CFG1	R/W	0x410	Configuration register for DMA channel 1.		Table 319
CTLSTAT1	RO	0x414	Control and status register for DMA channel 1.		Table 321
XFERCFG1	R/W	0x418	Transfer configuration register for DMA channel 1.		Table 322
Channel2 registers					
CFG2	R/W	0x420	Configuration register for DMA channel 2.		Table 319
CTLSTAT2	RO	0x424	Control and status register for DMA channel 2.		Table 321
XFERCFG2	R/W	0x428	Transfer configuration register for DMA channel 2.		Table 322
Channel3 registers					
CFG3	R/W	0x430	Configuration register for DMA channel 3.		Table 319
CTLSTAT3	RO	0x434	Control and status register for DMA channel 3.		Table 321
XFERCFG3	R/W	0x438	Transfer configuration register for DMA channel 3.		Table 322
Channel4 registers					
CFG4	R/W	0x440	Configuration register for DMA channel 4.		Table 319
CTLSTAT4	RO	0x444	Control and status register for DMA channel 4.		Table 321
XFERCFG4	R/W	0x448	Transfer configuration register for DMA channel 4.		Table 322
Channel5 registers					
CFG5	R/W	0x450	Configuration register for DMA channel 5.		Table 319
CTLSTAT5	RO	0x454	Control and status register for DMA channel 5.		Table 321
XFERCFG5	R/W	0x458	Transfer configuration register for DMA channel 5.		Table 322
Channel6 registers					
CFG6	R/W	0x460	Configuration register for DMA channel 6.		Table 319
CTLSTAT6	RO	0x464	Control and status register for DMA channel 6.		Table 321

Table 302. Register overview: DMA controller (base address 0x5000 8000)

Name	Access	Address offset	Description	Reset Value	Reference
XFERCFG6	R/W	0x468	Transfer configuration register for DMA channel 6.		Table 322
Channel7 registers					
CFG7	R/W	0x470	Configuration register for DMA channel 7.		Table 319
CTLSTAT7	RO	0x474	Control and status register for DMA channel 7.		Table 321
XFERCFG7	R/W	0x478	Transfer configuration register for DMA channel 7.		Table 322
Channel8 registers					
CFG8	R/W	0x480	Configuration register for DMA channel 8.		Table 319
CTLSTAT8	RO	0x484	Control and status register for DMA channel 8.		Table 321
XFERCFG8	R/W	0x488	Transfer configuration register for DMA channel 8.		Table 322
Channel9 registers					
CFG9	R/W	0x490	Configuration register for DMA channel 9.		Table 319
CTLSTAT9	RO	0x494	Control and status register for DMA channel 9.		Table 321
XFERCFG9	R/W	0x498	Transfer configuration register for DMA channel 9.		Table 322
Channel10 registers					
CFG10	R/W	0x4A0	Configuration register for DMA channel 10.		Table 319
CTLSTAT10	RO	0x4A4	Control and status register for DMA channel 10.		Table 321
XFERCFG10	R/W	0x4A8	Transfer configuration register for DMA channel 10.		Table 322
Channel11 registers					
CFG11	R/W	0x4B0	Configuration register for DMA channel 11.		Table 319
CTLSTAT11	RO	0x4B4	Control and status register for DMA channel 11.		Table 321
XFERCFG11	R/W	0x4B8	Transfer configuration register for DMA channel 11.		Table 322
Channel12 registers					
CFG12	R/W	0x4C0	Configuration register for DMA channel 12.		Table 319
CTLSTAT12	RO	0x4C4	Control and status register for DMA channel 12.		Table 321
XFERCFG12	R/W	0x4C8	Transfer configuration register for DMA channel 12.		Table 322
Channel13 registers					
CFG13	R/W	0x4D0	Configuration register for DMA channel 13.		Table 319
CTLSTAT13	RO	0x4D4	Control and status register for DMA channel 13.		Table 321
XFERCFG13	R/W	0x4D8	Transfer configuration register for DMA channel 13.		Table 322
Channel14 registers					
CFG14	R/W	0x4E0	Configuration register for DMA channel 14.		Table 319
CTLSTAT14	RO	0x4E4	Control and status register for DMA channel 14.		Table 321
XFERCFG14	R/W	0x4E8	Transfer configuration register for DMA channel 14.		Table 322
Channel15 registers					
CFG15	R/W	0x4F0	Configuration register for DMA channel 15.		Table 319
CTLSTAT15	RO	0x4F4	Control and status register for DMA channel 15.		Table 321
XFERCFG15	R/W	0x4F8	Transfer configuration register for DMA channel 15.		Table 322
Channel16 registers					
CFG16	R/W	0x500	Configuration register for DMA channel 16.		Table 319
CTLSTAT16	RO	0x504	Control and status register for DMA channel 16.		Table 321

Table 302. Register overview: DMA controller (base address 0x5000 8000)

Name	Access	Address offset	Description	Reset Value	Reference
XFERCFG16	R/W	0x508	Transfer configuration register for DMA channel 16.		Table 322
Channel17 registers					
CFG17	R/W	0x510	Configuration register for DMA channel 17.		Table 319
CTLSTAT17	RO	0x514	Control and status register for DMA channel 17.		Table 321
XFERCFG17	R/W	0x518	Transfer configuration register for DMA channel 17.		Table 322
Channel18 registers					
CFG18	R/W	0x520	Configuration register for DMA channel 18.		Table 319
CTLSTAT18	RO	0x524	Control and status register for DMA channel 18.		Table 321
XFERCFG18	R/W	0x528	Transfer configuration register for DMA channel 18.		Table 322
Channel19 registers					
CFG19	R/W	0x530	Configuration register for DMA channel 19.		Table 319
CTLSTAT19	RO	0x534	Control and status register for DMA channel 19.		Table 321
XFERCFG19	R/W	0x538	Transfer configuration register for DMA channel 19.		Table 322
Channel20 registers					
CFG20	R/W	0x540	Configuration register for DMA channel 20.		Table 319
CTLSTAT20	RO	0x544	Control and status register for DMA channel 20.		Table 321
XFERCFG20	R/W	0x548	Transfer configuration register for DMA channel 20.		Table 322
Channel21 registers					
CFG21	R/W	0x550	Configuration register for DMA channel 21.		Table 319
CTLSTAT21	RO	0x554	Control and status register for DMA channel 21.		Table 321
XFERCFG21	R/W	0x558	Transfer configuration register for DMA channel 21.		Table 322
Channel22 registers					
CFG22	R/W	0x560	Configuration register for DMA channel 22.		Table 319
CTLSTAT22	RO	0x564	Control and status register for DMA channel 22.		Table 321
XFERCFG22	R/W	0x568	Transfer configuration register for DMA channel 22.		Table 322
Channel23 registers					
CFG23	R/W	0x570	Configuration register for DMA channel 23.		Table 319
CTLSTAT23	RO	0x574	Control and status register for DMA channel 23.		Table 321
XFERCFG23	R/W	0x578	Transfer configuration register for DMA channel 23.		Table 322
Channel24 registers					
CFG24	R/W	0x580	Configuration register for DMA channel 24.		Table 319
CTLSTAT24	RO	0x584	Control and status register for DMA channel 24.		Table 321
XFERCFG24	R/W	0x588	Transfer configuration register for DMA channel 24.		Table 322

16.6.1 Control register

The CTRL register contains global the control bit for a enabling the DMA controller.

Table 303. 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

16.6.2 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 304. 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

16.6.3 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 305. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **273 of 530**

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 306. Only the descriptors for channels defined at extraction are used. The contents of each channel descriptor are described in Table 298.

Table 306. 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
Channel descriptor for DMA channel 16	0x100
Channel descriptor for DMA channel 17	0x110
Channel descriptor for DMA channel 18	0x120
Channel descriptor for DMA channel 19	0x130
Channel descriptor for DMA channel 20	0x140
Channel descriptor for DMA channel 21	0x150
Channel descriptor for DMA channel 22	0x160
Channel descriptor for DMA channel 23	0x170
Channel descriptor for DMA channel 24	0x180

16.6.4 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 307. Enable read and Set register 0 (ENABLESET0, address 0x5000 8020) bit description

Bit	Symbol	Description	Reset value
24:0	ENA	Enable for DMA channels 24:0. Bit n enables or disables DMA channel n. 0 = disabled. 1 = enabled.	0
31:25	-	Reserved.	-

16.6.5 Enable Clear register

The ENABLECLR0 register is used to clear the channel enable bits in ENABLESET0. This register is write-only.

Table 308. 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.	-

16.6.6 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 16.6.15).

Table 309. 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.	-

16.6.7 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 310. 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.	-

16.6.8 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 311. 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.	-

16.6.9 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 312. 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.	-

16.6.10 Interrupt Enable Clear register

The INTENCLR0 register is used to clear interrupt enable bits in INTENSET0. The register is write-only.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **276 of 530**

Table 313. 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.	-

16.6.11 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 314. 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.	-

16.6.12 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 315. 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.	-

16.6.13 Set Valid register

The SETVALID0 register allows setting the Valid bit in the CTRLSTAT register for one or more DMA channels. See Section 16.6.17 for a description of the VALID bit.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **277 of 530**

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 316. 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.	-

16.6.14 Set Trigger register

The SETTRIG0 register allows setting the TRIG bit in the CTRLSTAT register for one or more DMA channel. See Section 16.6.17 for a description of the TRIG bit, and Section 16.5.1 for a general description of triggering.

Table 317. 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.	-

16.6.15 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 318. 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.	-

UM11029
© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **278 of 530**

16.6.16 Channel configuration registers

The CFGn register contains various configuration options for DMA channel n.

See Table 320 for a summary of trigger options.

Table 319. Configuration registers for channel 0 to 24 (CFG[0:24], addresses 0x5000 8400 (CFG0) to address 0x5000 8564 (CFG24)) 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 319. Configuration registers for channel 0 to 24 (CFG[0:24], addresses 0x5000 8400 (CFG0) to address 0x5000 8564 (CFG24)) 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).	0
			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.	
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.	
18:16	CHPRIORITY		Priority of this channel when multiple DMA requests are pending.	0
			Eight priority levels are supported.	
			0x0 = highest priority.	
			0x7 = lowest priority.	
31:19	-		Reserved. Read value is undefined, only zero should be written.	NA

UM11029

User manual

© NXP Semiconductors N.V. 2017. All rights reserved.

Rev. 1.6 — 8 December 2017 **280 of 530**

Table 320. 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.

16.6.17 Channel control and status registers

The CTLSTATn register provides status flags specific to DMA channel n.

Table 321. Control and Status registers for channel 0 to 24 (CTLSTAT[0:24], 0x5000 8404 (CTLSTAT0) to address 0x5000 8568 (CTLSTAT24)) 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

16.6.18 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 322. Transfer Configuration registers for channel 0 to 24 (XFERCFG[0:24], addresses 0x5000 8408 (XFERCFG0) to 0x5000 856C (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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **282 of 530**

Table 322. Transfer Configuration registers for channel 0 to 24 (XFERCFG[0:24], addresses 0x5000 8408 (XFERCFG0) to 0x5000 856C (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

User manual   **Rev. 1.6 — 8 December 2017**   **283 of 530**

16.7 Functional description

16.7.1 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.

17.1 How to read this chapter

Five USARTs are available on all parts depending on the switch matrix configuration. UART3 and UART4 do not support RTS/CTS hardware signaling for automatic flow control.

17.2 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.

17.3 Basic configuration

Configure the USARTs for receiving and transmitting data:

- In the SYSAHBCLKCTRL register, set bit 14 to 16, bit 30, and 31 (Table 147) to enable the clock to the register interface.
- Clear the USART peripheral resets using the PRESETCTRL register (Table 149).
- Enable or disable the USART interrupts in slots #3, #4, #5, #30, and #31 in the NVIC. See Table 108.
- Configure the USART pin functions through the switch matrix.

- Configure the USART clock and baud rate. See Section 17.3.1.
- Send and receive lines are connected to DMA request lines. See Table 297.

For wake-up from deep-sleep and power-down modes the USART must be configured in synchronous mode. See Section 17.3.2 for details.

17.3.1 Configure the USART clock and baud rate

All five 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 24):

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 17.6.9 "USART Baud Rate Generator register"

3. In synchronous mode: The serial clock is Un_SCLK = FCLK/(BRGVAL+1).
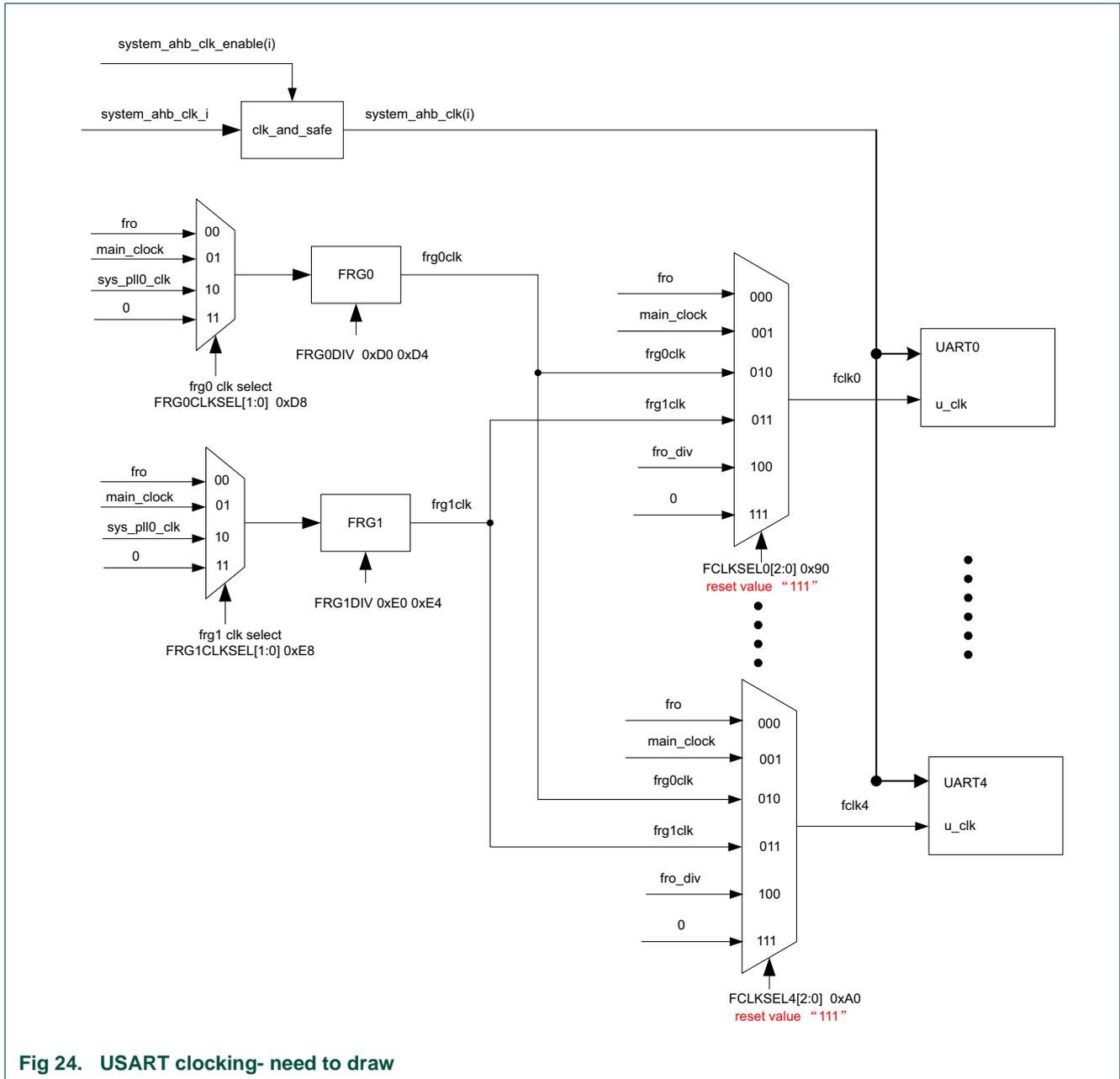
Fig 24. USART clocking- need to draw

For details on the clock configuration see:

Section 17.7.1 "Clocking and baud rates"

17.3.2 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.

17.3.2.1 Wake-up from sleep mode

- Configure the USART in either asynchronous mode or synchronous mode. See Table 325.
- 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 328).

17.3.2.2 Wake-up from deep-sleep or power-down mode

- Configure the USART in synchronous slave mode. See Table 325. 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 170 "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 328), you can customize when the wake-up occurs in the USART receive/transmit protocol.

17.4 Pin description

Table 323. USART pin description

Function	I/O	Type	Connect to	Use register	Reference	Description
U0_TXD	O	external to pin	any pin	PINASSIGN0	Table 181	Transmitter output for USART0. Serial transmit data.
U0_RXD	I	external to pin	any pin	PINASSIGN0	Table 181	Receiver input for USART0.
U0_RTS	O	external to pin	any pin	PINASSIGN0	Table 181	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 181	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 182	Serial clock input/output for USART0 n synchronous mode.
U1_TXD	O	external to pin	any pin	PINASSIGN1	Table 182	Transmitter output for USART1. Serial transmit data.
U1_RXD	I	external to pin	any pin	PINASSIGN1	Table 182	Receiver input for USART1.
$\overline{\text{U1\_RTS}}$	O	external to pin	any pin	PINASSIGN1	Table 182	Request To Send output for USART1.
$\overline{\text{U1\_CTS}}$	I	external to pin	any pin	PINASSIGN2	Table 183	Clear To Send input for USART1.
U1_SCLK	I/O	external to pin	any pin	PINASSIGN2	Table 183	Serial clock input/output for USART1 in synchronous mode.
U2_TXD	O	external to pin	any pin	PINASSIGN2	Table 183	Transmitter output for USART2. Serial transmit data.
U2_RXD	I	external to pin	any pin	PINASSIGN2	Table 183	Receiver input for USART2.
$\overline{\text{U2\_RTS}}$	O	external to pin	any pin	PINASSIGN3	Table 184	Request To Send output for USART2.
$\overline{\text{U2\_CTS}}$	I	external to pin	any pin	PINASSIGN3	Table 184	Clear To Send input for USART2.
U2_SCLK	I/O	external to pin	any pin	PINASSIGN3	Table 184	Serial clock input/output for USART2 in synchronous mode.
U3_TXD	O	external to pin	any pin	PINASSIGN11	Table 182	Transmitter output for USART3. Serial transmit data.
U3_RXD	I	external to pin	any pin	PINASSIGN12	Table 182	Receiver input for USART3.

Table 323. USART pin description

Function	I/O	Type	Connect to	Use register	Reference	Description
U3_SCLK	I/O	external to pin	any pin	PINASSIGN12	Table 183	Serial clock input/output for USART3 in synchronous mode.
U4_TXD	O	external to pin	any pin	PINASSIGN12	Table 183	Transmitter output for USART4. Serial transmit data.
U4_RXD	I	external to pin	any pin	PINASSIGN12	Table 183	Receiver input for USART4.
U4_SCLK	I/O	external to pin	any pin	PINASSIGN13	Table 184	Serial clock input/output for USART4 in synchronous mode.

17.5 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.

USART block diagram

Fig 25.   USART block diagram-need to modify in eps
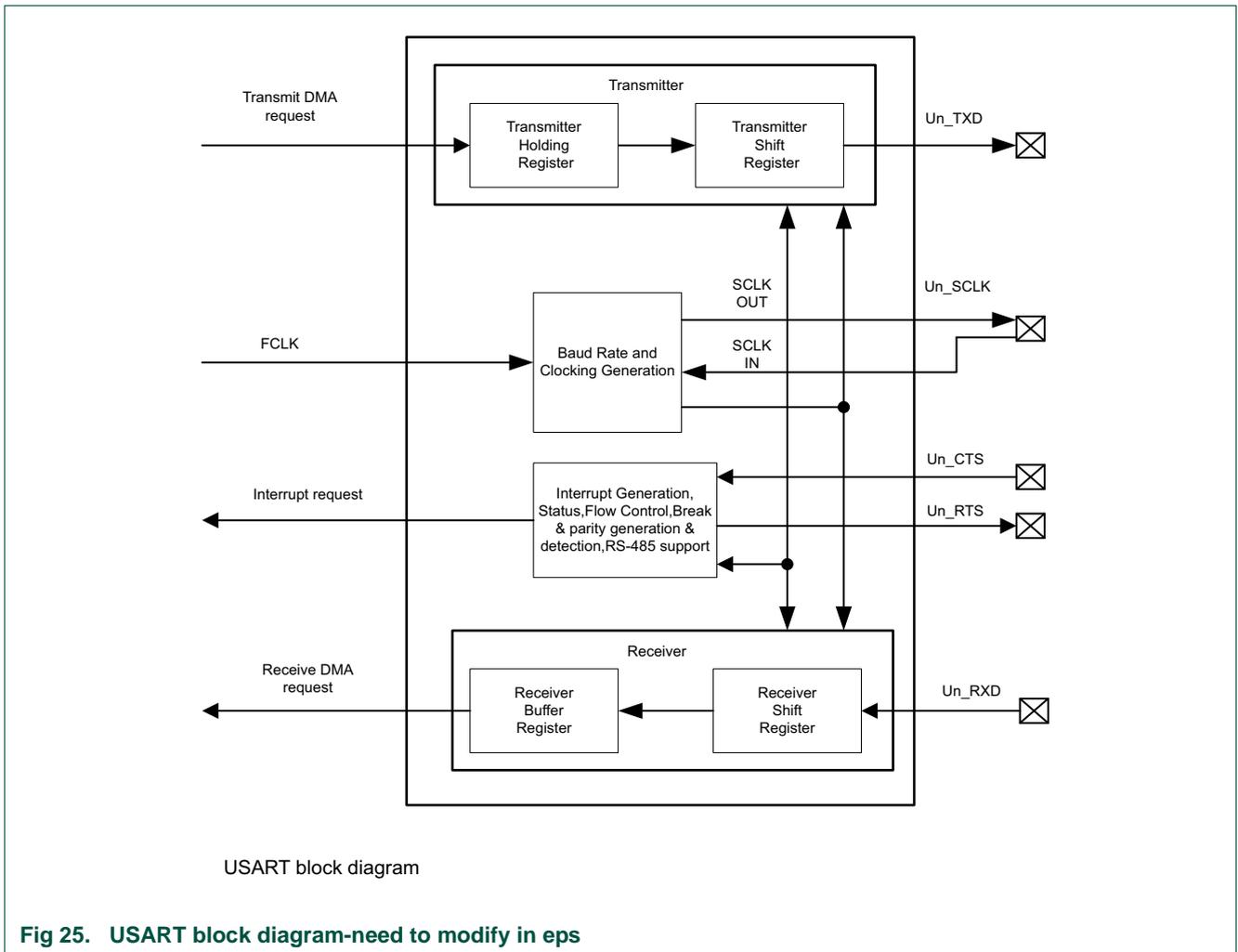
17.6 Register description

The reset value reflects the data stored in used bits only. It does not include the content of reserved bits.

Table 324. Register overview: USART (base address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2), 0x4007 0000 (USART3), 0x4007 4000 (USART4))

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 325
CTL	R/W	0x004	USART Control register. USART control settings that are more likely to change during operation.	0	Table 326
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 327
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 328
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 329
RXDAT	R	0x014	Receiver Data register. Contains the last character received.	-	Table 330
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 331
TXDAT	R/W	0x01C	Transmit Data register. Data to be transmitted is written here.	0	Table 332
BRG	R/W	0x020	Baud Rate Generator register. 16-bit integer baud rate divisor value.	0	Table 333
INTSTAT	R	0x024	Interrupt status register. Reflects interrupts that are currently enabled.	0x0005	Table 334
OSR	R/W	0x028	Oversample selection register for asynchronous communication.	0xF	Table 335
ADDR	R/W	0x02C	Address register for automatic address matching.	0	Table 336

17.6.1 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 325. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2), 0x4007 0000 (USART3), 0x4007 4000 (USART4)) 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

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **293 of 530**

Table 325. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2), 0x4007 0000 (USART3), 0x4007 4000 (USART4)) 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

Table 325. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1), 0x4006 C000 (USART2), 0x4007 0000 (USART3), 0x4007 4000 (USART4)) bit description *…continued*

Bit	Symbol	Value	Description	Reset Value
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.	
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

17.6.2 USART Control register

The CTL register controls aspects of USART operation that are more likely to change during operation.

Table 326. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1), 0x4006 C004 (USART2), 0x4007 0004 (USART3), 0x4007 4004 (USART4)) bit description

Bit	Symbol	Value	Description	Reset value
0	-		Reserved. Read value is undefined, only zero should be written.	NA
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).	

Table 326. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1), 0x4006 C004 (USART2), 0x4007 0004 (USART3), 0x4007 4004 (USART4)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
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.	
31:17	-		Reserved. Read value is undefined, only zero should be written.	NA

17.6.3 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 329).

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 327. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1), 0x4006 C008 (USART2), 0x4007 0008 (USART3), 0x4007 4008 (USART4)) 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

Table 327. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1), 0x4006 C008 (USART2), 0x4007 0008 (USART3), 0x4007 4008 (USART4)) bit description

Bit	Symbol	Description	Reset value	Access [1]
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
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
31:17	-	Reserved. Read value is undefined, only zero should be written.	NA	NA

[1] RO = Read-only, W1 = write 1 to clear.

17.6.4 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 328. USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1), 0x4006C00C (USART2), 0x4007 000C (USART3), 0x4007 400C (USART4)) 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
31:17	-	Reserved. Read value is undefined, only zero should be written.	NA

17.6.5 USART Interrupt Enable Clear register

The INTENCLR register is used to clear bits in the INTENSET register.

Table 329. USART Interrupt Enable clear register (INTENCLR, address 0x4006 4010 (USART0), 0x4006 8010 (USART1), 0x4006 C010 (USART2), 0x4007 0010 (USART3), 0x4007 4010 (USART4)) 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
31:17	-	Reserved. Read value is undefined, only zero should be written.	NA

17.6.6 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 330. USART Receiver Data register (RXDAT, address 0x4006 4014 (USART0),
0x4006 8014 (USART1), 0x4006 C014 (USART2), 0x4007 0014 (USART3), 0x4007
4014 (USART4)) 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

17.6.7 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 331. USART Receiver Data with Status register (RXDATSTAT, address 0x4006 4018
(USART0), 0x4006 8018 (USART1), 0x4006 C018 (USART2), 0x4007 0018
(USART3), 0x4007 4018 (USART4)) 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 327.	0
31:16	-	Reserved, the value read from a reserved bit is not defined.	NA

17.6.8 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 332. USART Transmitter Data Register (TXDAT, address 0x4006 401C (USART0), 0x4006 801C (USART1), 0x4006 C01C (USART2), 0x4007 001C (USART3), 0x4007 401C (USART4)) 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

17.6.9 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 <u>Section 17.7.1</u>.

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 333. USART Baud Rate Generator register (BRG, address 0x4006 4020 (USART0), 0x4006 8020 (USART1), 0x4006 8020 (USART2), 0x4007 0020 (USART3), 0x4007 4020 (USART4)) 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

17.6.10 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 327 for detailed descriptions of the interrupt flags.

Table 334. USART Interrupt Status register (INTSTAT, address 0x4006 4024 (USART0), 0x4006 8024 (USART1), 0x4006 8024 (USART2), 0x4007 0024 (USART3), 0x4007 4024 (USART4)) 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
31:17	-	Reserved. Read value is undefined, only zero should be written.	NA

17.6.11 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 335. USART Oversample selection register (OSR, address 0x4006 4028 (USART0), 0x4006 4028 (USART1), 0x4006 8028 (USART2), 0x4007 0028 (USART3), 0x4007 4028 (USART4)) 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

17.6.12 USART Address register

The ADDR register holds the address for hardware address matching in address detect mode with automatic address matching enabled.

Table 336. USART Address register (ADDR, address 0x4006 402C (USART0), 0x4006 802C (USART1), 0x4006 C02C (USART2), 0x4007 002C (USART3), 0x4007 402C (USART4)) 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

17.7 Functional description

17.7.1 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 24.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **304 of 530**

17.7.1.1 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 LPC84x device can use either FRG0 or FRG1. See Table 152 "Fractional generator 0 divider value register (FRG0DIV, address 0x4004 80D0) bit description" and Table 153 "Fractional generator 0 multiplier value register (FRG0MULT, address 0x4004 80D4) bit description".

For details see Section 17.3.1 "Configure the USART clock and baud rate".

17.7.1.2 Baud Rate Generator (BRG)

The Baud Rate Generator (see Section 17.6.9) 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.

17.7.1.3 Baud rate calculations

Base clock rates are 16x for asynchronous mode and 1x for synchronous mode.

17.7.2 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.

17.7.3 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.

17.7.4 Flow control

The USART supports both hardware and software flow control.

17.7.4.1 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 26 shows an overview of RTS and CTS within the USART.
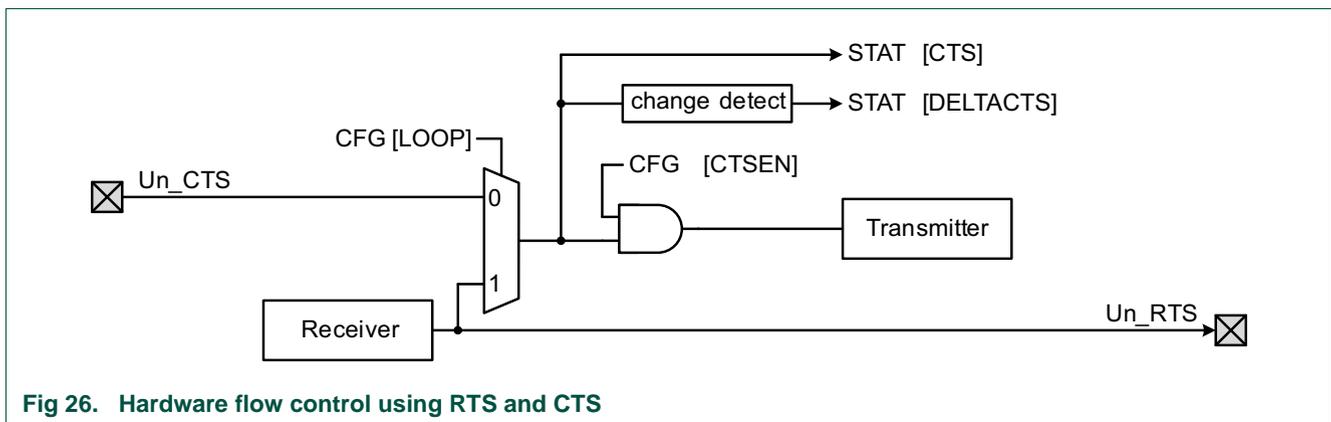
Fig 26.   Hardware flow control using RTS and CTS

17.7.4.2 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).

17.7.5 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.

17.7.6 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 17.6.1 and the ADDR register in Section 17.6.12), as well as software address recognition (see the ADDRDET bit in the CTL register in Section 17.6.2).

Automatic data direction control with the RTS pin can be set up using the OESEL, OEPOL, and OETA bits in the CFG register (Section 17.6.1). Data direction control can also be implemented in software using a GPIO pin.

17.7.7 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.

18.1 How to read this chapter

The SPI interfaces are available on all parts depending on the switch matrix configuration.

18.2 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.

18.3 Basic configuration

Configure SPI0/1 using the following registers:

- In the SYSAHBCLKCTRL register, set bit 11 and 12 (Table 147) to enable the clock to the register interface.
- Clear the SPI0/1 peripheral resets using the PRESETCTRL register (Table 149).
- 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 18.4.
- Using the SPI0CLKSEL register, the peripheral clock sources for the SPI can be FRO, main_clk, frg0clk, frg1clk, and fro_div. (see Figure 7 "LPC84x clock generation").
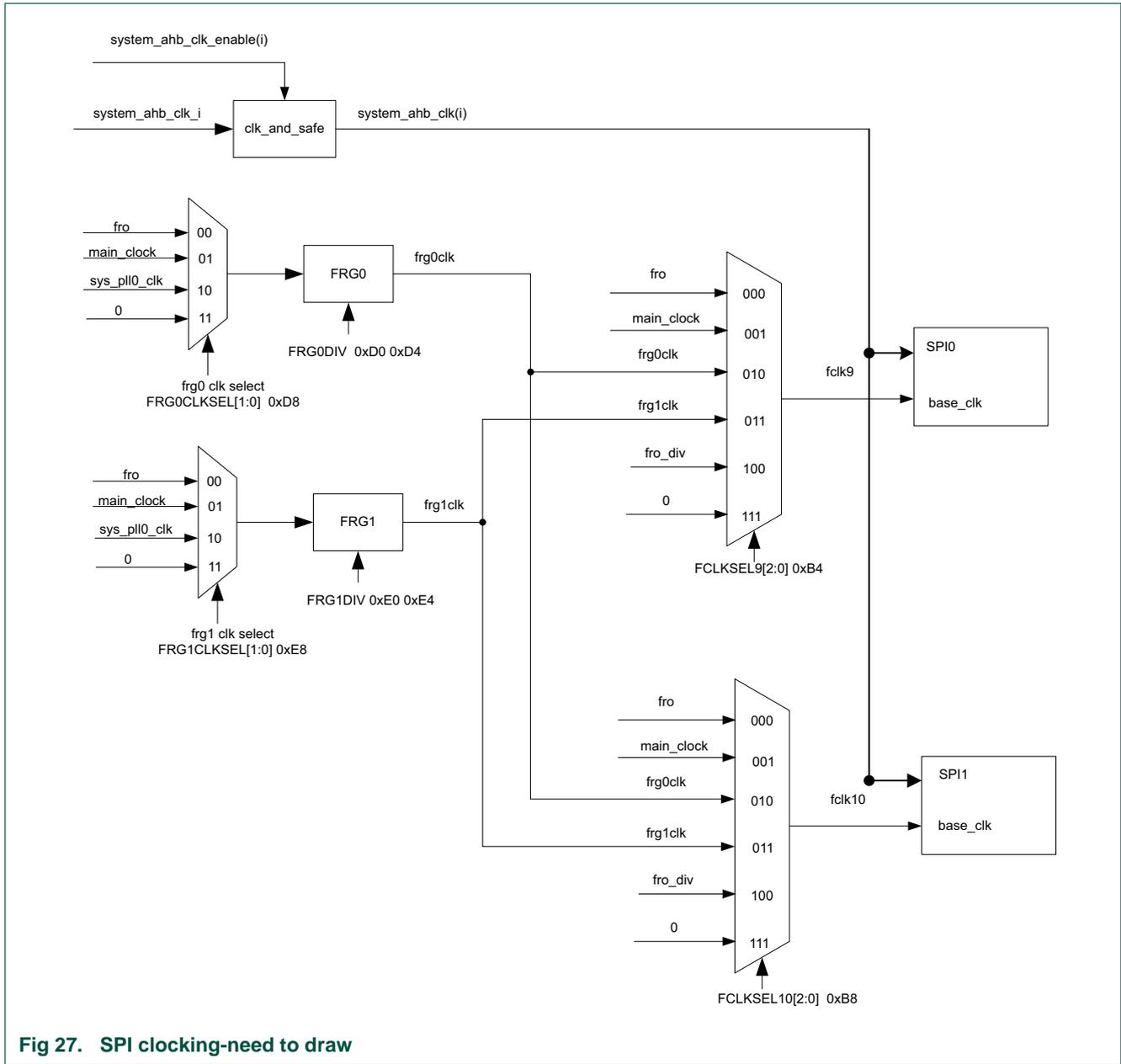
Fig 27. SPI clocking-need to draw

18.3.1 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.

18.3.1.1 Wake-up from Sleep mode

- Configure the SPI in either master or slave mode. See Table 339.

- 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 342).

18.3.1.2 Wake-up from Deep-sleep or Power-down mode

- Configure the SPI in slave mode. See Table 339. 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 170 "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 342). Examples are the following wake-up events:

  - A change in the state of the SSEL pins.

  - Data available to be received.

  - Receiver overrun.

18.4 Pin description

The SPI signals are movable functions and are assigned to external pins through the switch matrix.

See Section 10.3.1 "Connect an internal signal to a package pin" to assign the SPI functions to pins on the part.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **310 of 530**

Table 337. SPI Pin Description

Function	I/O	Type	Connect to	Use register	Reference	Description
SPI0_SCK	I/O	external to pin	any pin	PINASSIGN3	Table 184	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 185	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 185	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.
SPI0_SSEL0	I/O	external to pin	any pin	PINASSIGN4	Table 185	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 185	Slave Select 1.
SPI0_SSEL2	I/O	external to pin	any pin	PINASSIGN5	Table 186	Slave Select 2.
SPI0_SSEL3	I/O	external to pin	any pin	PINASSIGN5	Table 186	Slave Select 3.
SPI1_SCK	I/O	external to pin	any pin	PINASSIGN5	Table 186	Serial Clock.
SPI1_MOSI	I/O	external to pin	any pin	PINASSIGN5	Table 186	Master Out Slave In.
SPI1_MISO	I/O	external to pin	any pin	PINASSIGN6	Table 187	Master In Slave Out.
SPI1_SSEL0	I/O	external to pin	any pin	PINASSIGN6	Table 187	Slave Select 0.
SPI1_SSEL1	I/O	external to pin	any pin	PINASSIGN6	Table 187	Slave Select 1.

18.5 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 28.   SPI block diagram
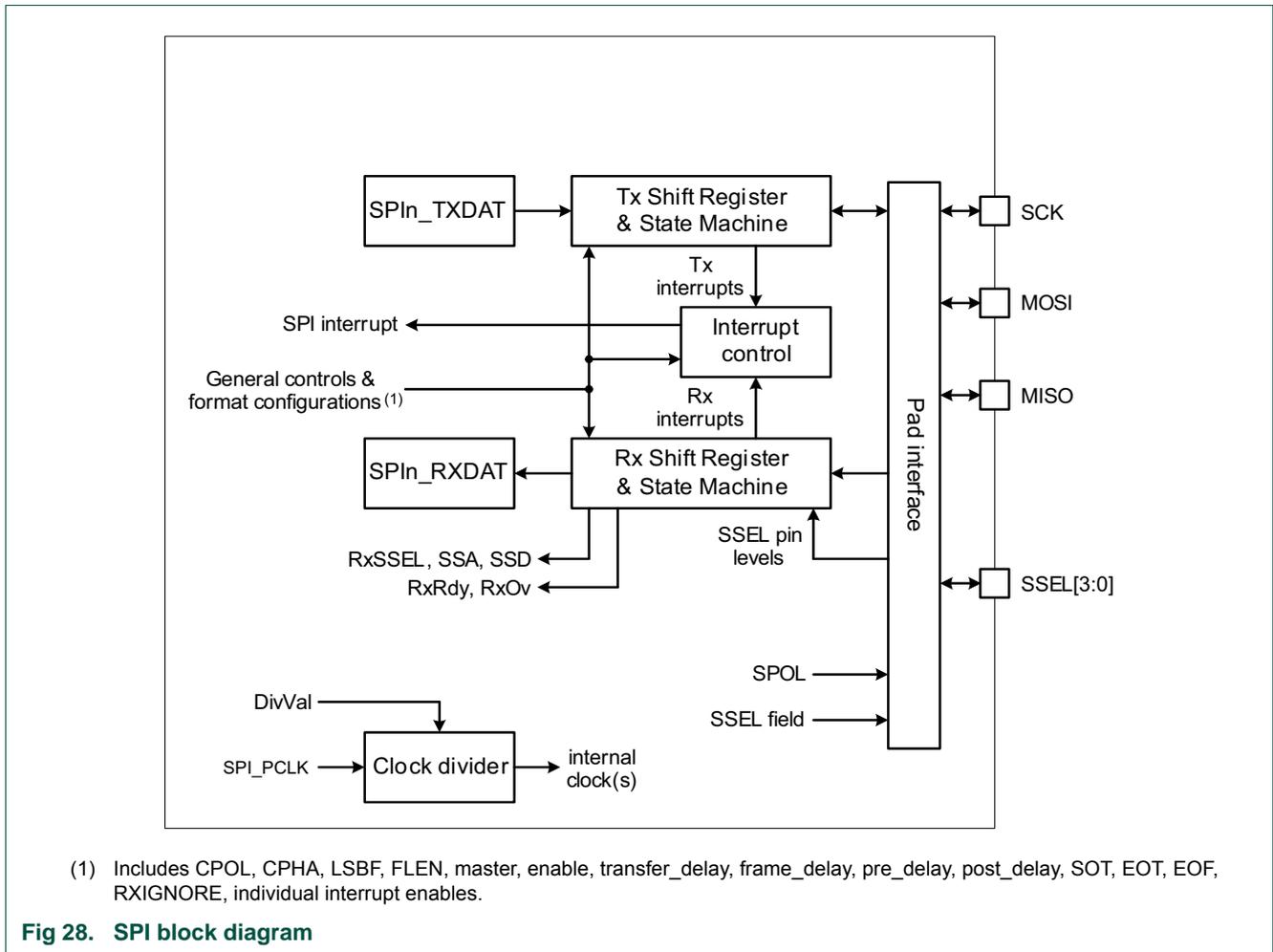
18.6 Register description

The Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

Table 338.   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 339
DLY	R/W	0x004	SPI Delay register	0	Table 340
STAT	R/W	0x008	SPI Status. Some status flags can be cleared by writing a 1 to that bit position	0x0102	Table 341

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **312 of 530**

Table 338. 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 342
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 343
RXDAT	R	0x014	SPI Receive Data	NA	Table 344
TXDATCTL	R/W	0x018	SPI Transmit Data with Control	0	Table 345
TXDAT	R/W	0x01C	SPI Transmit Data	0	Table 346
TXCTL	R/W	0x020	SPI Transmit Control	0	Table 347
DIV	R/W	0x024	SPI clock Divider	0	Table 348
INTSTAT	R	0x028	SPI Interrupt Status	0x02	Table 349

18.6.1 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 341) 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 339. 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 339. 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

18.6.2 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 18.7.2.1 "Pre_delay and Post_delay"

Section 18.7.2.2 "Frame_delay"

Section 18.7.2.3 "Transfer_delay"

Table 340. 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

18.6.3 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 341. 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

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **316 of 530**

Table 341. 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.

18.6.4 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 341 for details of the interrupts.

Table 342. 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.	0
			The interface prevents receiver overrun in Master mode by not allowing a new transmission to begin when a receiver overrun would otherwise occur.	
		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.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **317 of 530**

Table 342. 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

18.6.5 SPI Interrupt Enable Clear register

The INTENCLR register is used to clear interrupt enable bits in the INTENSET register.

Table 343. 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

18.6.6 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 18.7.4.

Table 344. 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	RXSSEL3_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

18.6.7 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 18.6.8) instead of the TXDATCTL register. Control information can then be written separately via the TXCTL register (see Section 18.6.9). 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 18.7.4.

For details on using multiple consecutive data transmits for transfer lengths larger than 16 bit, see Section 18.7.6 "Data lengths greater than 16 bits".

Table 345. 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 345. 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

18.6.8 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 18.6.7). That data will be sent to the transmit shift register when it is available, and another character may then be written to TXDAT.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **321 of 530**

Table 346. 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

18.6.9 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 18.6.7). 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 18.6.7) instead of TXDAT. Control information can then be written along with data.

Table 347. 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

18.6.10 SPI Divider register

The DIV register determines the clock used by the SPI in master mode.

For details on clocking, see Section 18.7.3 "Clocking and data rates".

Table 348. 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

18.6.11 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 341 for detailed descriptions of the interrupt flags.

Table 349. 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

18.7 Functional description

18.7.1 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 350 and shown in Figure 29. CPOL and CPHA are configured by bits in the CFG register (Section 18.6.1).

Table 350. 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
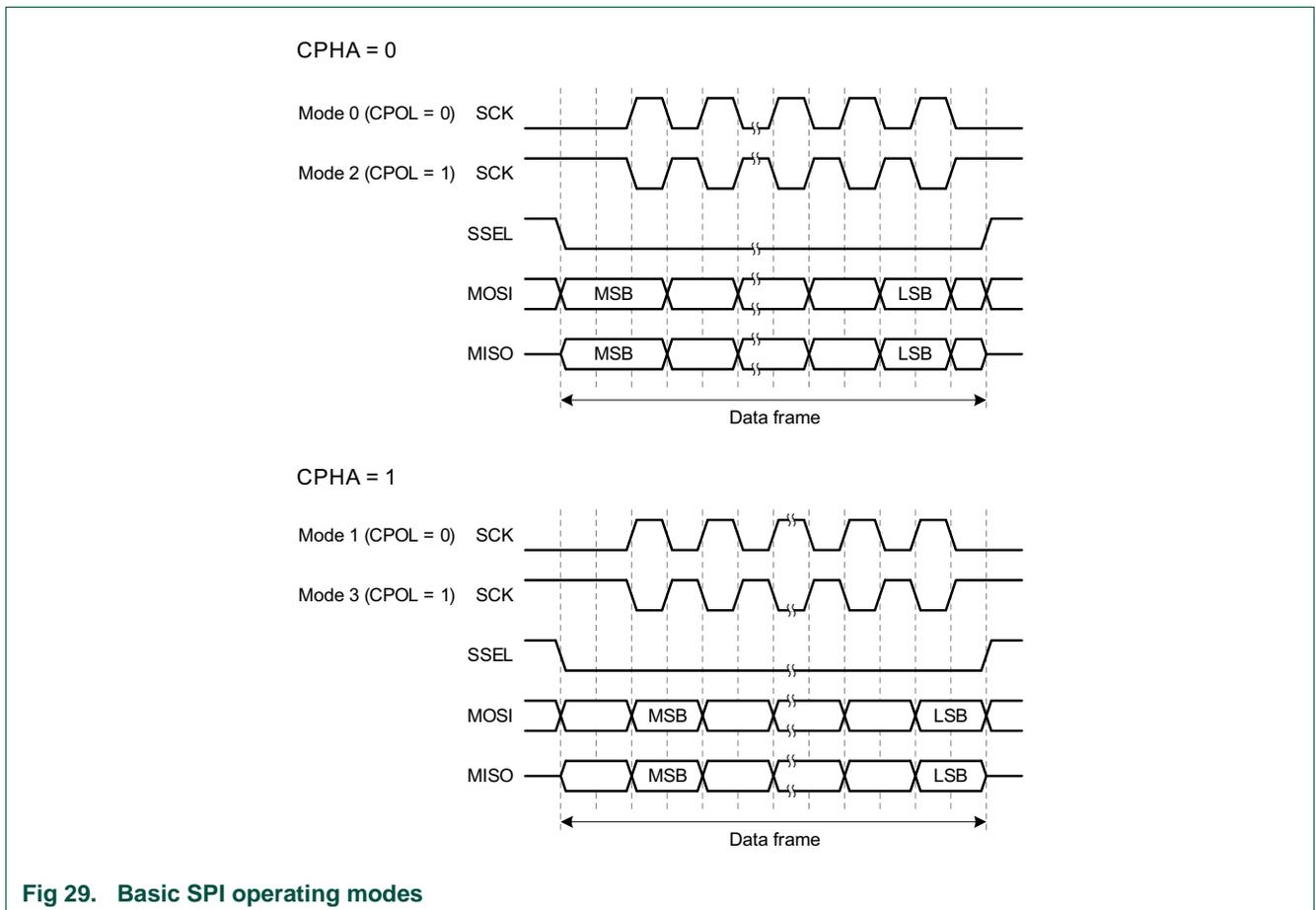
Fig 29. Basic SPI operating modes

18.7.2 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

18.7.2.1 Pre_delay and Post_delay

Pre_delay and Post_delay are illustrated by the examples in Figure 30. 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 30.   Pre_delay and Post_delay

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **325 of 530**
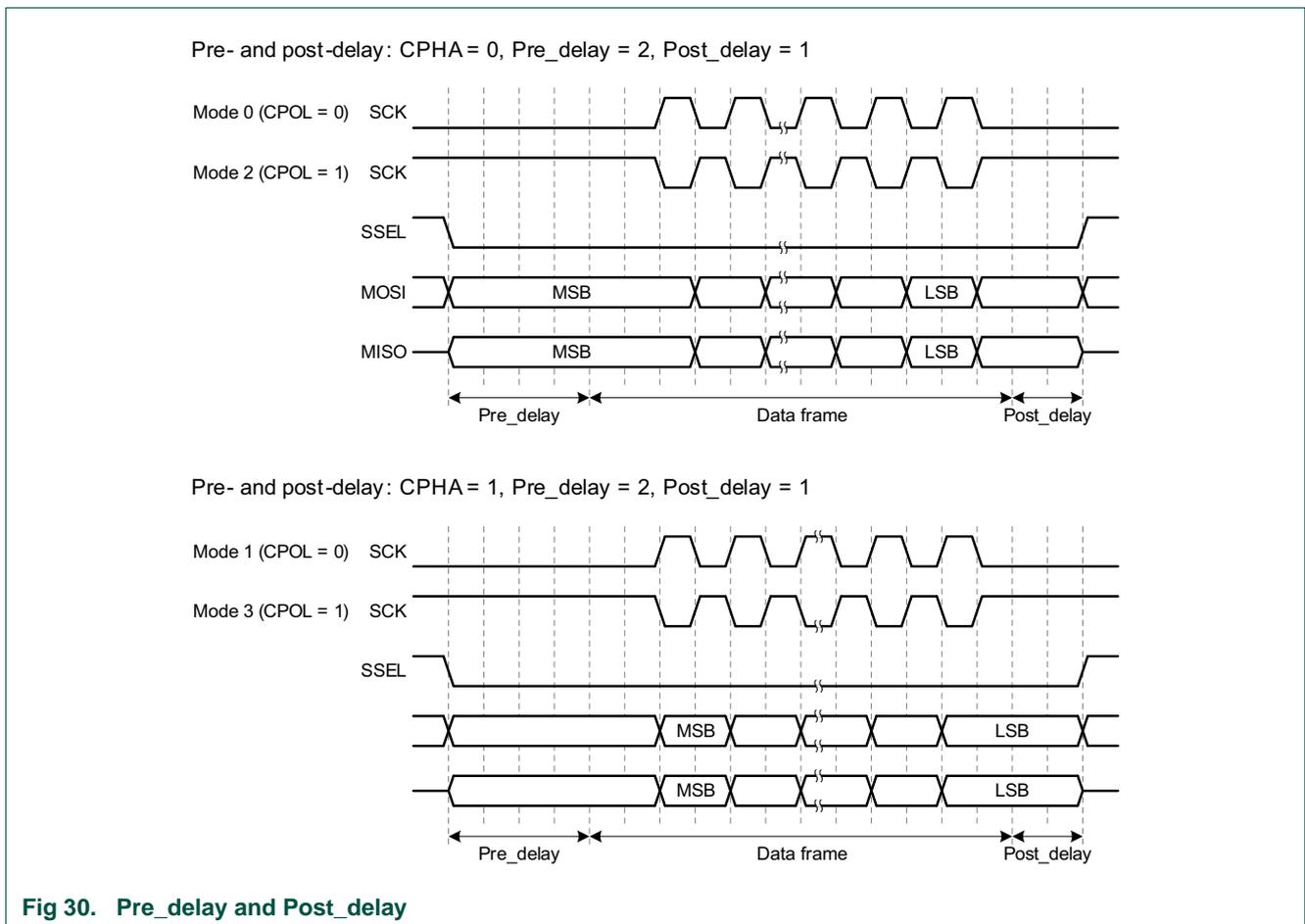
18.7.2.2 Frame_delay

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 31. Note that frame boundaries occur only where specified. This is because frame lengths can be any size, involving multiple data writes. See Section 18.7.6 for more information.

Fig 31. Frame_delay
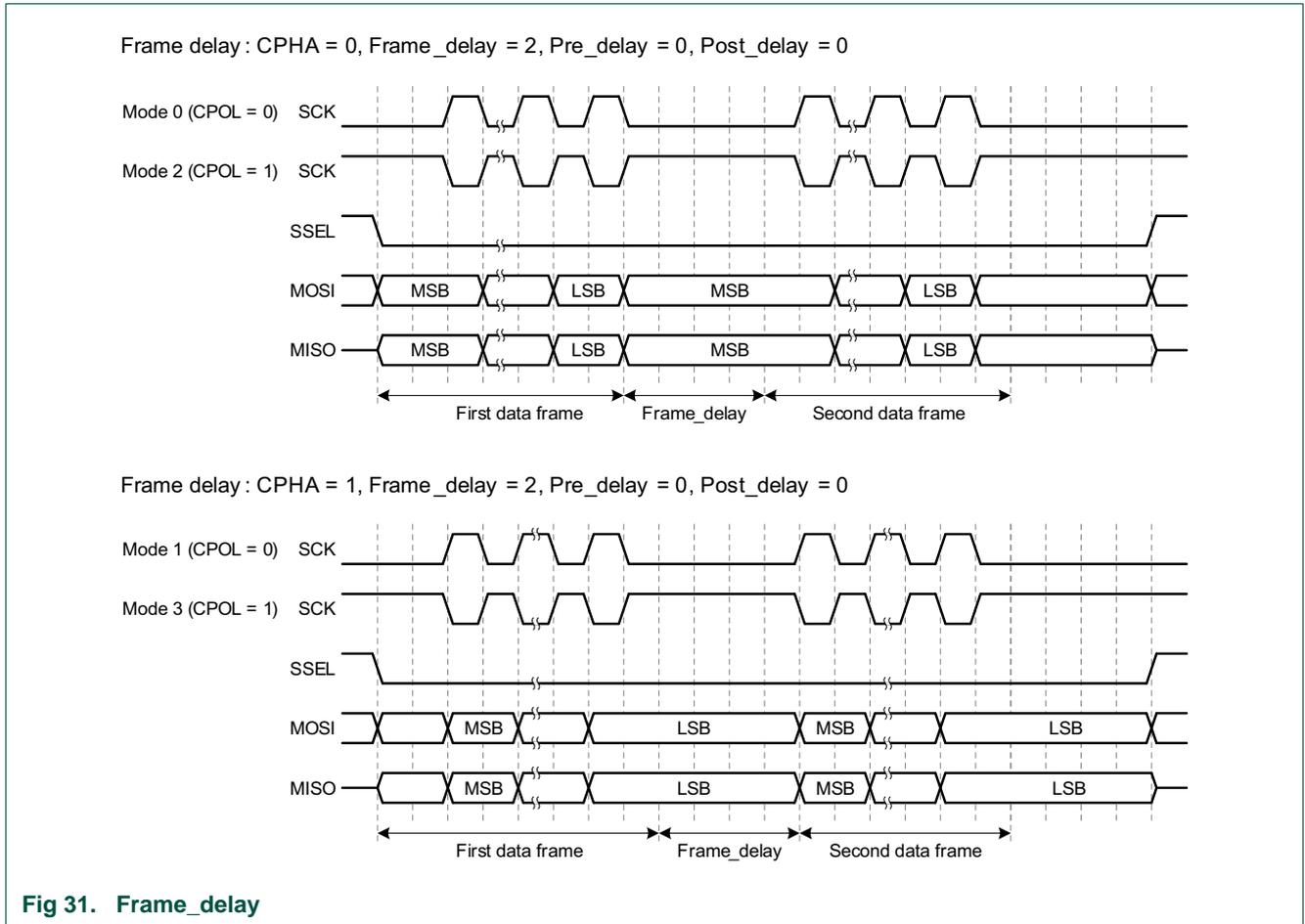
18.7.2.3 Transfer_delay

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 32.

Fig 32. Transfer_delay
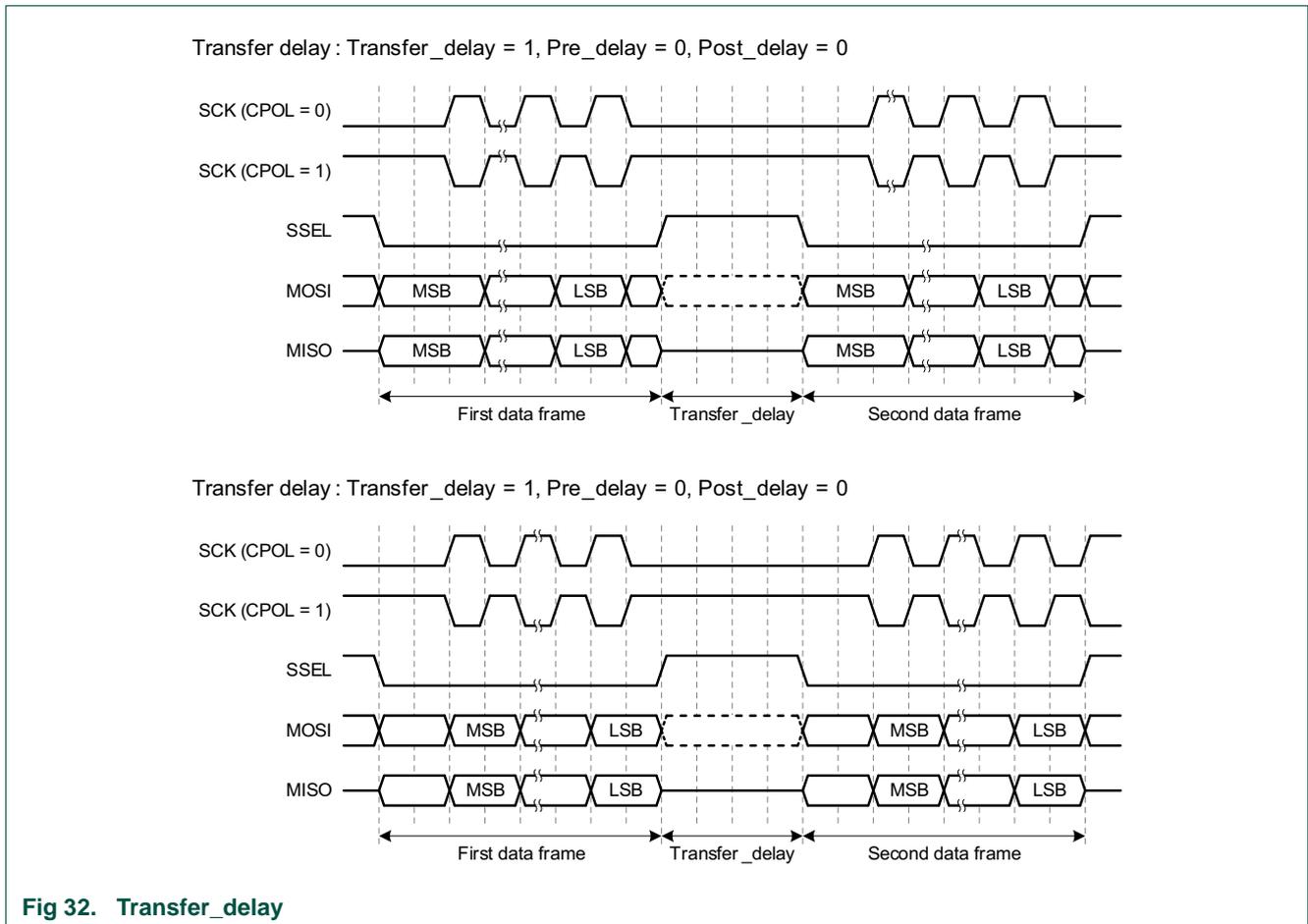
18.7.3 Clocking and data rates

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 27.

18.7.3.1 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.

18.7.4 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.

18.7.5 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.

18.7.6 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.

18.7.7 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.

Fig 33. Examples of data stalls

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **330 of 530**
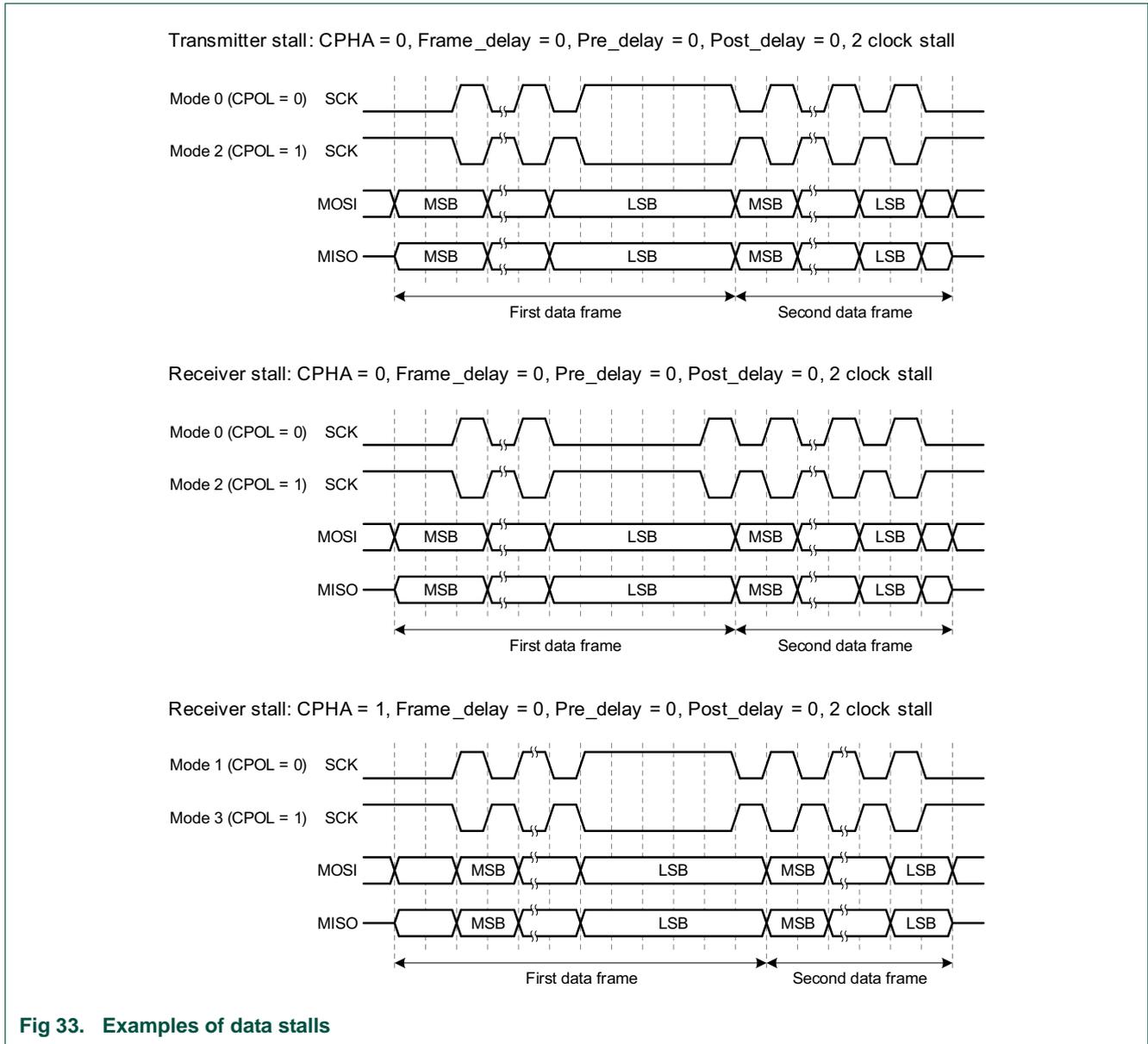
19.1 How to read this chapter

Four I2C interfaces are available on all parts depending on the switch matrix configuration.

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.

19.2 Features

- Independent Master, Slave, and Monitor functions.
- Supports bothDMA_ITRIG_PINMUX Multi-master and Multi-master with Slave functions.
- Multiple I^{2}C 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^{2}C bus addresses.
- 10-bit addressing supported with software assist.
- Supports System Management (SMBus).
- Separate DMA requests for Master and Slave.
- Supports the I^{2}C-bus specification up to Fast-mode Plus (up to 1 MHz).

19.3 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 147.
- Clear the I2C peripheral resets using the PRESETCTRL register (Table 149).
- Enable/disable the I2C interrupt in interrupt slots #7, 8, 21, 22 in the NVIC. See Table 108.
- Configure the I2C pin functions through the switch matrix. See Table 351.
- The peripheral clock for the I2C is the system clock (see Figure 34).

Gray-shaded blocks show peripherals that can provide hardware triggers or fixed DMA requests for DMA transfers.

Fig 34.   I2C clocking-need to draw
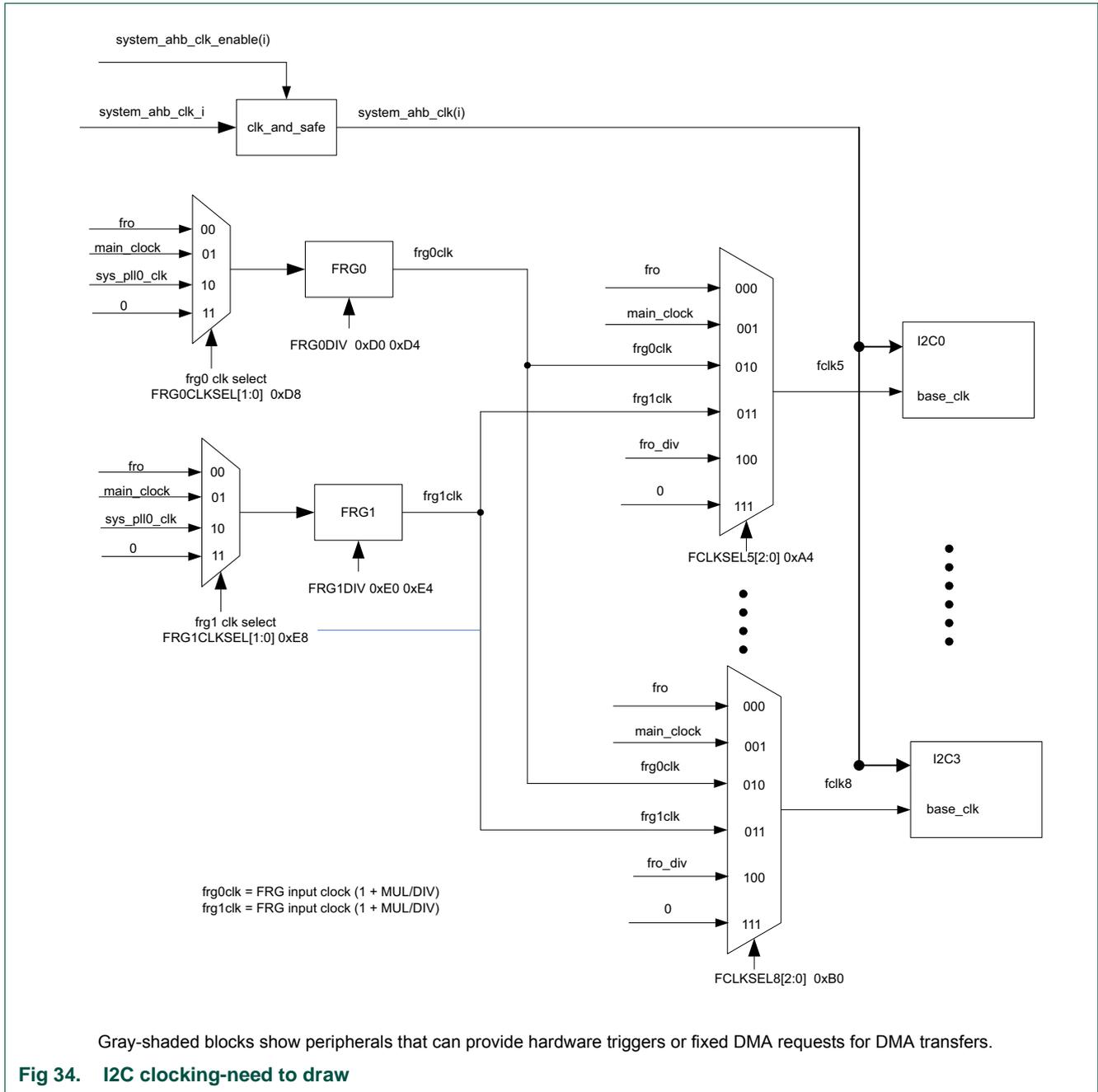
19.3.1 I2C transmit/receive in master mode

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 30 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 351.

For a 400 kHz bit rate, the I2C0 pins can be configured in standard mode in the IOCON block. See Table 208 "PIO0_11 register (PIO0_11, address 0x4004 401C) bit description" and Table 209 "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 360 "I^{2}C Clock Divider register (CLKDIV, address 0x4005 0014 (I2C0), 0x4005 4014 (I2C1), 0x4003 0014 (I2C2), 0x4003 4014 (I2C3)) bit description".
- Set the SCL high and low times to 2 clock cycles each. This is the default. See Table 363 "Master Time register (MSTTIME, address 0x4005 0024 (I2C0), 0x4005 4024 (I2C1), 0x4003 0024 (I2C2), 0x4003 4024 (I2C3)) bit description". The result is an SCL clock of 375 kHz.

19.3.1.1 Master write to slave

Configure the I2C as master: Set the MSTEN bit to 1 in the CFG register. See Table 353.

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 364.

2. Start the transmission by setting the MSTSTART bit to 1 in the Master control register. See Table 362. 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 362. 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 362.

19.3.1.2 Master read from slave

Configure the I2C as master: Set the MSTEN bit to 1 in the CFG register. See Table 353.

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 364.

2. Start the transmission by setting the MSTSTART bit to 1 in the Master control register. See Table 362. 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 362.

19.3.2 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 351.

For a 400 kHz bit rate, the pins can be configured in standard mode in the IOCON block. See Table 208 "PIO0_11 register (PIO0_11, address 0x4004 401C) bit description" and Table 209 "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.

19.3.2.1 Slave read from master

Configure the I2C as slave with address x:

- Set the SLVEN bit to 1 in the CFG register. See Table 353.
- Write the slave address x to the address 0 match register. See Table 367.

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 365.
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 366.
5. Acknowledge ("ack") the data by setting SLVCONTINUE = 1 in the slave control register. See Table 365.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **335 of 530**

19.3.2.2 Slave write to master

- Set the SLVEN bit to 1 in the CFG register. See Table 353.
- Write the slave address x to the address 0 match register. See Table 367.

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 365.
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 366.
5. Continue the transaction by setting SLVCONTINUE = 1 in the slave control register. See Table 365.

19.3.3 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.

19.3.3.1 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

19.3.3.2 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 170 "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

19.4 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 208 "PIO0_11 register (PIO0_11, address 0x4004 401C) bit description" and Table 209 "PIO0_10 register (PIO0_10, address 0x4004 4020) bit description".

Pins for the I2C1/2/3 interfaces are movable functions and can be assigned to any pin. However, except for PIO0_10 and PIO0_11, the pins are not open-drain and do not support Fast-mode Plus mode. Bit rates of 400 kHz are supported on all pins.

Table 351. I2C-bus pin description

Function	Direction	Type	Connect to	Use register	Reference	Description
I2C0_SDA	I/O	external to pin	PIO0_11	PINENABLE0	Table 196	I2C0 serial data.
I2C0_SCL	I/O	external to pin	PIO0_10	PINENABLE0	Table 196	I2C0 serial clock.
I2C1_SDA	I/O	external to pin	any pin	PINASSIGN9	Table 190	I2C1 serial data.
I2C1_SCL	I/O	external to pin	any pin	PINASSIGN9	Table 190	I2C1 serial clock.
I2C2_SDA	I/O	external to pin	any pin	PINASSIGN10	Table 190	I2C2 serial data.
I2C2_SCL	I/O	external to pin	any pin	PINASSIGN10	Table 191	I2C2 serial clock.
I2C3_SDA	I/O	external to pin	any pin	PINASSIGN10	Table 191	I2C3 serial data.
I2C3_SCL	I/O	external to pin	any pin	PINASSIGN10	Table 191	I2C3 serial clock.

19.5 General description

The architecture of the I2C-bus interface is shown in Figure 35.

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **337 of 530**

Fig 35. I2C block diagram

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **338 of 530**
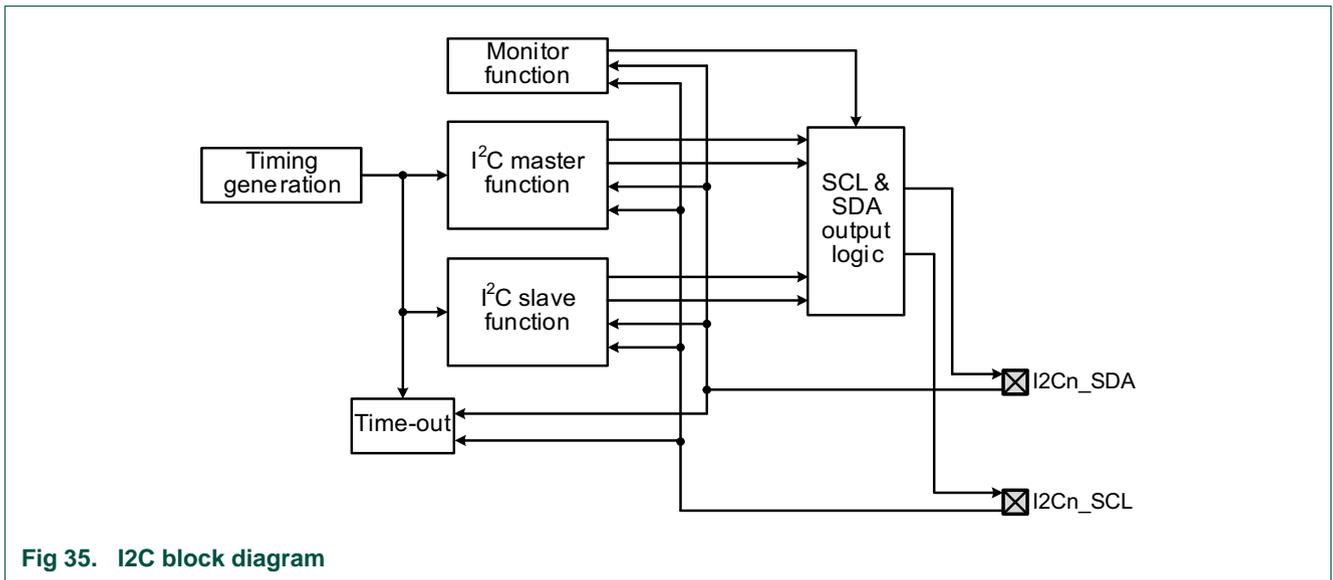
19.6 Register description

Table 352. Register overview: I2C (base address 0x4005 0000 (I2C0), 0x4005 4000 (I2C1), 0x4003 0000 (I2C2), 0x4003 4000 (I2C3))

Name	Access	Offset	Description	Reset value	Reference
CFG	R/W	0x00	Configuration for shared functions.	0	Table 353
STAT	R/W	0x04	Status register for Master, Slave, and Monitor functions.	0x000801	Table 354
INTENSET	R/W	0x08	Interrupt Enable Set and read register.	0	Table 357
INTENCLR	W	0x0C	Interrupt Enable Clear register.	NA	Table 358
TIMEOUT	R/W	0x10	Time-out value register.	0xFFFF	Table 359
CLKDIV	R/W	0x14	Clock pre-divider for the entire I2C block. This determines what time increments are used for the MSTTIME and SLVTIME registers.	0	Table 360
INTSTAT	R	0x18	Interrupt Status register for Master, Slave, and Monitor functions.	0	Table 361
MSTCTL	R/W	0x20	Master control register.	0	Table 362
MSTTIME	R/W	0x24	Master timing configuration.	0x77	Table 363
MSTDAT	R/W	0x28	Combined Master receiver and transmitter data register.	NA	Table 364
SLVCTL	R/W	0x40	Slave control register.	0	Table 365
SLVDAT	R/W	0x44	Combined Slave receiver and transmitter data register.	NA	Table 366
SLVADR0	R/W	0x48	Slave address 0.	0x01	Table 367
SLVADR1	R/W	0x4C	Slave address 1.	0x01	Table 367
SLVADR2	R/W	0x50	Slave address 2.	0x01	Table 367
SLVADR3	R/W	0x54	Slave address 3.	0x01	Table 367
SLVQUAL0	R/W	0x58	Slave Qualification for address 0.	0	Table 368
MONRXDAT	RO	0x80	Monitor receiver data register.	0	Table 369

19.6.1 I2C Configuration register

The CFG register contains mode settings that apply to Master, Slave, and Monitor functions.

Table 353. I2C Configuration register (CFG, address 0x4005 0000 (I2C0), 0x4005 4000 (I2C1), 0x4003 0000 (I2C2), 0x4003 4000 (I2C3)) 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 I2C Master function is disabled.	
		1	Enabled. The I2C 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 I2C slave function is disabled.	
		1	Enabled. The I2C slave function is enabled.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **339 of 530**

Table 353. I2C Configuration register (CFG, address 0x4005 0000 (I2C0), 0x4005 4000 (I2C1), 0x4003 0000 (I2C2), 0x4003 4000 (I2C3)) 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **340 of 530**

19.6.2 I2C Status register

The STAT register provides status flags and state information about all of the functions of the I2C 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 355 and Table 356.

Table 354. I2C Status register (STAT, address 0x4005 0004 (I2C0), 0x4005 4004 (I2C1), 0x4003 0004 (I2C2), 0x4003 4004 (I2C3)) 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 354. I²C Status register (STAT, address 0x4005 0004 (I2C0), 0x4005 4004 (I2C1), 0x4003 0004 (I2C2), 0x4003 4004 (I2C3)) 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 I²C 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 I²C 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 I²C 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 I²C bus clock. Deep-sleep or Power-down mode could be entered at this time.		

Table 354. I²C Status register (STAT, address 0x4005 0004 (I2C0), 0x4005 4004 (I2C1), 0x4003 0004 (I2C2), 0x4003 4004 (I2C3)) bit description *…continued*

Bit	Symbol	Value	Description	Reset value	Access
13:12	SLVIDX		Slave address match Index. This field is valid when the I²C 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 I²C 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 I²C bus to be inactive.		
		1	Active. The Monitor function considers the I²C bus to be active.		

UM11029

User manual **Rev. 1.6 — 8 December 2017** **343 of 530**

Table 354. I²C Status register (STAT, address 0x4005 0004 (I2C0), 0x4005 4004 (I2C1), 0x4003 0004 (I2C2), 0x4003 4004 (I2C3)) 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 I²C 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 I²C bus is not idle, or this flag has been cleared by software.		
		1	Idle. The I²C 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. I²C bus events have not caused a time-out.		
		1	Event time-out. The time between I²C 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 355. 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 356. 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 19.7.3 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.	-	-

19.6.3 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 357. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0), 0x4005 4008 (I2C1), 0x4003 0008 (I2C2), 0x4003 4008 (I2C3)) 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 357. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0),
0x4005 4008 (I2C1), 0x4003 0008 (I2C2), 0x4003 4008 (I2C3)) 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

19.6.4 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 358. Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0), 0x4005 400C (I2C1), 0x4003 000C (I2C2), 0x4003 400C (I2C3)) 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

19.6.5 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 19.7.2 "Time-out".

Table 359. Time-out value register (TIMEOUT, address 0x4005 0010 (I2C0), 0x4005 4010 (I2C1), 0x4003 0010 (I2C2), 0x4003 4010 (I2C3)) 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 I2C function clocks and also a time-out resolution of 16 I2C function clocks.	0xF
15:4	TO	Time-out time value. Specifies the time-out interval value in increments of 16 I2C function clocks, as defined by the CLKDIV register. To change this value while I2C 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 I2C function clock. 0x001 = A time-out will occur after 32 counts of the I2C function clock. ... 0xFFF = A time-out will occur after 65,536 counts of the I2C function clock.	0xFFF
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

19.6.6 Clock Divider register

The CLKDIV register divides down the Peripheral Clock (PCLK) to produce the I2C function clock that is used to time various aspects of the I2C interface. The I2C function clock is used for some internal operations in the I2C block and to generate the timing required by the I2C bus specification, some of which are user configured in the MSTTIME register for Master operation and the SLVTIME register for Slave operation.

See Section 19.7.1.1 "Rate calculations" for details on bus rate setup.

Table 360. I2C Clock Divider register (CLKDIV, address 0x4005 0014 (I2C0), 0x4005 4014 (I2C1), 0x4003 0014 (I2C2), 0x4003 4014 (I2C3)) bit description

Bit	Symbol	Description	Reset value
15:0	DIVVAL	This field controls how the clock (PCLK) is used by the I2C functions that need an internal clock in order to operate. 0x0000 = PCLK is used directly by the I2C function. 0x0001 = PCLK is divided by 2 before use by the I2C function. 0x0002 = PCLK is divided by 3 before use by the I2C function. ... 0xFFFF = PCLK is divided by 65,536 before use by the I2C function.	0
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

19.6.7 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 354 for detailed descriptions of the interrupt flags.

Table 361. I2C Interrupt Status register (INTSTAT, address 0x4005 0018 (I2C0), 0x4005 4018 (I2C1), 0x4003 0018 (I2C2), 0x4003 4018 (I2C3)) 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

19.6.8 Master Control register

The MSTCTL register contains bits that control various functions of the I2C Master interface. Only write to this register when the master is pending (MSTPENDING = 1 in the STAT register, Table 354).

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 355), set the MSTDMA bit either with or after the MSTCONTINUE bit. MSTDMA can be cleared at any time.

Table 362. Master Control register (MSTCTL, address 0x4005 0020 (I2C0), 0x4005 4020 (I2C1), 0x4003 0020 (I2C2), 0x4003 4020 (I2C3)) 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 I2C 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 I2C 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 I2C 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 I2C master data operations. When this I2C 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

19.6.9 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 360.

Table 363. Master Time register (MSTTIME, address 0x4005 0024 (I2C0), 0x4005 4024 (I2C1), 0x4003 0024 (I2C2), 0x4003 4024 (I2C3)) 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^{2}C bus specification. I^{2}C 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^{2}C clock pre-divider.	
		0x1	3 clocks. Minimum SCL low time is 3 clocks of the I^{2}C clock pre-divider.	
		0x2	4 clocks. Minimum SCL low time is 4 clocks of the I^{2}C clock pre-divider.	
		0x3	5 clocks. Minimum SCL low time is 5 clocks of the I^{2}C clock pre-divider.	
		0x4	6 clocks. Minimum SCL low time is 6 clocks of the I^{2}C clock pre-divider.	
		0x5	7 clocks. Minimum SCL low time is 7 clocks of the I^{2}C clock pre-divider.	
		0x6	8 clocks. Minimum SCL low time is 8 clocks of the I^{2}C clock pre-divider.	
		0x7	9 clocks. Minimum SCL low time is 9 clocks of the I^{2}C clock pre-divider.	
3	-		Reserved.	0

UM11029

User manual **Rev. 1.6 — 8 December 2017** **351 of 530**

Table 363. Master Time register (MSTTIME, address 0x4005 0024 (I2C0), 0x4005 4024 (I2C1), 0x4003 0024 (I2C2), 0x4003 4024 (I2C3)) bit description ...continued

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

19.6.10 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 364. Master Data register (MSTDAT, address 0x4005 0028 (I2C0), 0x4005 4028 (I2C1), 0x4003 0028 (I2C2), 0x4003 4028 (I2C3)) 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

UM11029

User manual   **Rev. 1.6 — 8 December 2017**   **352 of 530**

19.6.11 Slave Control register

The SLVCTL register contains bits that control various functions of the I2C Slave interface. Only write to this register when the slave is pending (SLVPENDING = 1 in the STAT register, Table 354).

Remark: When in the slave address state (slave state 0, see Table 356), set the SLVDMA bit either with or after the SLVCONTINUE bit. SLVDMA can be cleared at any time.

Table 365. Slave Control register (SLVCTL, address 0x4005 0040 (I2C0), 0x4005 4040 (I2C1), 0x4003 0040 (I2C2), 0x4003 4040 (I2C3)) 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 I2C slave data transmission and reception.	
31:4	-		Reserved. Read value is undefined, only zero should be written.	NA

19.6.12 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 366. Slave Data register (SLVDAT, address 0x4005 0044 (I2C0), 0x4005 4044 (I2C1), 0x4003 0044 (I2C2), 0x4003 4044 (I2C3)) 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

19.6.13 Slave Address registers

The SLVADR[0:3] registers allow enabling and defining one of the addresses that can be automatically recognized by the I^{2}C 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 19.6.14).

The I^{2}C 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 367. **Slave Address registers (SLVADR[0:3], address 0x4005 0048 (SLVADR0) to 0x4005 0054 (SLVADR3) (I2C0), 0x4005 4048 (SLVADR0) to 0x4005 4054 (SLVADR3) (I2C1), 0x4003 0048 (SLVADR0) to 0x4003 0054 (SLVADR3) (I2C2), 0x4003 4048 (SLVADR0) to 0x4003 4054 (SLVADR3) (I2C3)) 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

19.6.14 Slave address Qualifier 0 register

The SLVQUAL0 register can alter how Slave Address 0 is interpreted.

Table 368. Slave address Qualifier 0 register (SLVQUAL0, address 0x4005 0058 (I2C0), 0x4005 4058 (I2C1), 0x4003 0058 (I2C2), 0x4003 4058 (I2C3)) 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.	0
			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]).	
31:8	-		Reserved. Read value is undefined, only zero should be written.	NA

19.6.15 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 369. Monitor data register (MONRXDAT, address 0x4005 0080 (I2C0), 0x4005 4080 (I2C1), 0x4003 0080 (I2C2), 0x4003 4080 (I2C3)) 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

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **355 of 530**

Table 369. Monitor data register (MONRXDAT, address 0x4005 0080 (I2C0), 0x4005 4080 (I2C1), 0x4003 0080 (I2C2), 0x4003 4080 (I2C3)) bit description

Bit	Symbol	Value	Description	Reset value
8	MONSTART		Monitor Received Start.	0
		0	No detect. The monitor function has not detected a Start event on the I2C bus.	
		1	Start detect. The monitor function has detected a Start event on the I2C bus.	
9	MONRESTART		Monitor Received Repeated Start.	0
		0	No start detect. The monitor function has not detected a Repeated Start event on the I2C bus.	
		1	Repeated start detect. The monitor function has detected a Repeated Start event on the I2C 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

19.7 Functional description

19.7.1 Bus rates and timing considerations

Due to the nature of the I2C 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 I2C bus could operate if nothing slows it down.

19.7.1.1 Rate calculations

SCL high time (in I2C function clocks) = (CLKDIV + 1) * (MSTSCLHIGH + 2)

SCL low time (in I2C function clocks) = (CLKDIV + 1) * (MSTSCLLOW + 2)

Nominal SCL rate = I2C 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 I2C divider (divval) must be ≤ 2 MHz. Table 370 shows the recommended settings for 400 KHz clock rate.

Table 370. Settings for 400 KHz clock rate

Input clock to I2C	DIVVAL for CLKDIV register	MSTSCLHIGH for MSTTIME register	MSTSCLLOW for MSTTIME register
30 MHz	14	0	1
24 MHz	14	0	0
24 MHz	11	0	1
12 MHz	5	0	1

19.7.2 Time-out

A time-out feature on an I2C 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 I2C 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 I2C 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 I2C bus within a system as part of a method to keep the bus running of problems occur.

The second type of I2C 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 I2C 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 I2C bus is considered busy.

19.7.3 Ten-bit addressing

Ten-bit addressing is accomplished by the I2C 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 I2C 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 I2C 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 19.6.14) 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.

19.7.4 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.

19.7.5 Interrupt handling

The I2C provides a single interrupt output that handles all interrupts for Master, Slave, and Monitor functions.

19.7.6 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.

20.1 How to read this chapter

The standard timer is available on all LPC84x devices.

20.2 Features

- 32-bit counter/timer with a programmable 32-bit prescaler. The timer includes external capture and match pin connections.
- Counter or timer operation.
- Up to four 32-bit captures can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt. The number of capture inputs that are actually available on device pins may vary by device.
- The timer and prescaler may be configured to be cleared on a designated capture event. This feature permits easy pulse-width measurement by clearing the timer on the leading edge of an input pulse and capturing the timer value on the trailing edge.
- Four 32-bit match registers that allow:
  - Continuous operation with optional interrupt generation on match.
  - Optional auto-reload from match shadow registers when counter is reset.
  - Stop timer on match with optional interrupt generation.
  - Reset timer on match with optional interrupt generation.
- Up to 4 external outputs corresponding to match registers with the following capabilities (the number of match outputs that are actually available on device pins may vary by device):
  - Set LOW on match.
  - Set HIGH on match.
  - Toggle on match.
  - Do nothing on match.
- Up to 4 match registers can be configured for PWM operation, allowing up to 3 single edged controlled PWM outputs. (The number of match outputs that are actually available on device pins may vary by device.)
- Up to 2 match registers can be used to generate DMA requests.

20.3 Basic configuration

- Set the appropriate bit to enable the clock to timer: CTIMER in the AHBCLKCTRL0 register (Section 8.6.22).
- Clear the timer reset using the PRESETCTRL1 register (Table 150 for CTIMER). Note that the bit position in the reset control register matches the bit position in the clock control register.
- Pins: Select timer pins through switch matrix.

- Interrupts: See register MCR (Table 378) and CCR (Table 380) for match and capture events. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register. For interrupt connections, see Table 108.
- DMA: Some timer match conditions can be used to generate timed DMA requests, see Table 297.

20.4 Applications

- Interval timer for counting internal events.
- Pulse Width Modulator via match outputs.
- Pulse Width Demodulator via capture input.
- Free running timer.

20.5 General description

The Counter/timer block is designed to count cycles of the APB bus clock or an externally supplied clock and can optionally generate interrupts or perform other actions at specified timer values based on four match registers. The counter/timer also includes capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, three match registers can be used to provide a single-edge controlled PWM output on the match output pins. One match register is used to control the PWM cycle length. All match registers can optionally be auto-reloaded from a companion shadow register whenever the counter is reset to zero. This permits modifying the match values for the next counter cycle without risk of disrupting the PWM waveforms during the current cycle. When enabled, match reload will occur whenever the counter is reset either due to a match event or a write to bit 1 of the Timer Control Register (TCR).

20.5.1 Capture inputs

The capture signal can be configured to load the Capture Register with the value in the counter/timer and optionally generate an interrupt. The capture signal is generated by one of the pins with a capture function. Each capture signal is connected to one capture channel of the timer.

The Counter/Timer block can select a capture signal as a clock source instead of the APB bus clock. For more details see Section 20.7.11.

20.5.2 Match outputs

When a match register equals the timer counter (TC), the corresponding match output can either toggle, go LOW, go HIGH, or do nothing. The External Match Register (EMR) and the PWM Control Register (PWMCON) control the functionality of this output.

20.5.3 Architecture

Figure 36 shows the block diagram for the timer.
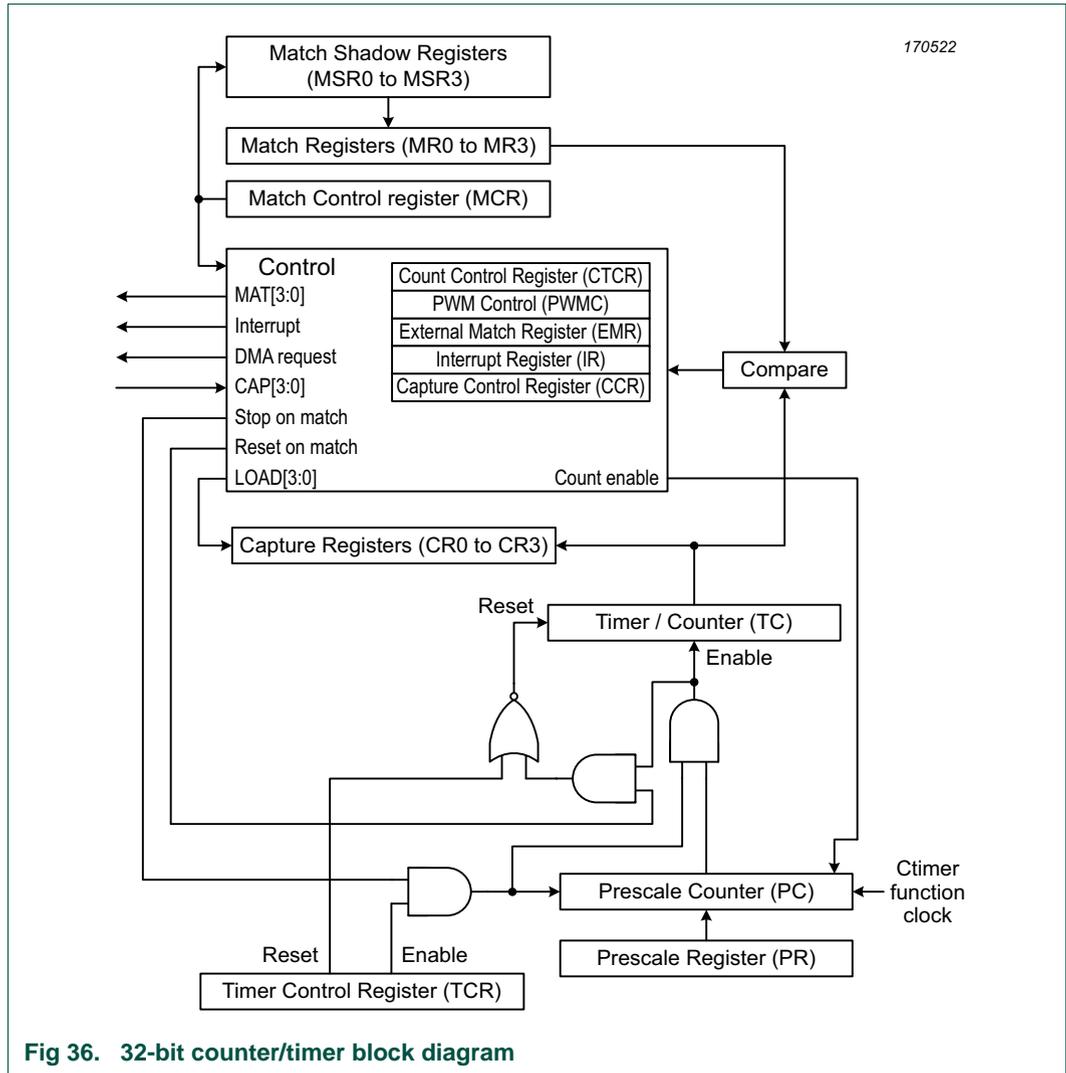
Fig 36. 32-bit counter/timer block diagram

20.6 Pin description

Table 371 gives a brief summary of each of the Timer/Counter related pins.

Table 371. Timer/Counter pin description

Function	Type	Connect to	Use register	Description
TO_CAPx	Input	Any pin	PINASSIGN14	Capture signals- A transition on a capture pin can be configured to load one of the Capture Registers with the value in the Timer Counter and optionally generate an interrupt. Capture functionality can be selected from a number of pins. Timer/Counter block can select a capture signal as a clock source instead of the APB bus clock. For more details see Section 20.7.11.
TO_MATx	Output	Any pin	PINASSIGN13	External Match Output - When a match register equals the timer counter (TC) this output can either toggle, go low, go high, or do nothing. The External Match Register (EMR) controls the functionality of this output. Match Output functionality can be selected on a number of pins in parallel.
TO_MATx	Output	Any pin	PINASSIGN14	External Match Output - When a match register equals the timer counter (TC) this output can either toggle, go low, go high, or do nothing. The External Match Register (EMR) controls the functionality of this output. Match Output functionality can be selected on a number of pins in parallel.

20.6.1 Multiple CAP and MAT pins

Software can select from multiple pins for the CAP or MAT functions in the switch matrix registers. Note that match conditions may be used internally without the use of a device pin.

20.7 Register description

The timer/counter contains the registers shown in Table 372.

Table 372. Register overview: CTIMER (register base addresses 0x4003 8000)

Name	Access	Offset	Description	Reset value[1]	Section
IR	R/W	0x00	Interrupt Register. The IR can be written to clear interrupts. The IR can be read to identify which of eight possible interrupt sources are pending.	0	20.7.1
TCR	R/W	0x04	Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	0	20.7.2
TC	R/W	0x08	Timer Counter. The 32 bit TC is incremented every PR+1 cycles of the APB bus clock. The TC is controlled through the TCR.	0	20.7.3
PR	R/W	0x0C	Prescale Register. When the Prescale Counter (PC) is equal to this value, the next clock increments the TC and clears the PC.	0	20.7.4
PC	R/W	0x10	Prescale Counter. The 32 bit PC is a counter which is incremented to the value stored in PR. When the value in PR is reached, the TC is incremented and the PC is cleared. The PC is observable and controllable through the bus interface.	0	20.7.5
MCR	R/W	0x14	Match Control Register. The MCR is used to control whether an interrupt is generated, whether the TC is reset when a Match occurs, and whether the match register is reloaded from its shadow register when the TC is reset.	0	20.7.6
MR0	R/W	0x18	Match Register 0. MR0 can be enabled through the MCR to reset the TC, stop both the TC and PC, and/or generate an interrupt every time MR0 matches the TC.	0	20.7.7
MR1	R/W	0x1C	Match Register 1. See MR0 description.	0	20.7.7
MR2	R/W	0x20	Match Register 2. See MR0 description.	0	20.7.7
MR3	R/W	0x24	Match Register 3. See MR0 description.	0	20.7.7
CCR	R/W	0x28	Capture Control Register. The CCR controls which edges of the capture inputs are used to load the Capture Registers and whether or not an interrupt is generated when a capture takes place.	0	20.7.8
CR0	RO	0x2C	Capture Register 0. CR0 is loaded with the value of TC when there is an event on the CAPn.0 input.	0	20.7.9
CR1	RO	0x30	Capture Register 1. See CR0 description.	0	20.7.9
CR2	RO	0x34	Capture Register 2. See CR0 description.	0	20.7.9
CR3	RO	0x38	Capture Register 3. See CR0 description.	0	20.7.9
EMR	R/W	0x3C	External Match Register. The EMR controls the match function and the external match pins.	0	20.7.10
CTCR	R/W	0x70	Count Control Register. The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	0	20.7.11
PWMC	R/W	0x74	PWM Control Register. The PWMCON enables PWM mode for the external match pins.	0	20.7.12
MSR0	R/W	0x78	Match 0 Shadow Register. If enabled, the Match 0 Register will be automatically reloaded with the contents of this register whenever the TC is reset to zero.	0	20.7.13

Table 372. Register overview: CTIMER (register base addresses 0x4003 8000)

Name	Access	Offset	Description	Reset value[1]	Section
MSR1	R/W	0x7C	Match 1 Shadow Register. If enabled, the Match 1 Register will be automatically reloaded with the contents of this register whenever the TC is reset to zero.	0	20.7.13
MSR2	R/W	0x80	Match 2 Shadow Register. If enabled, the Match 2 Register will be automatically reloaded with the contents of this register whenever the TC is reset to zero.	0	20.7.13
MSR3	R/W	0x84	Match 3 Shadow Register. If enabled, the Match 3 Register will be automatically reloaded with the contents of this register whenever the TC is reset to zero.	0	20.7.13

[1] Reset Value reflects the data stored in used bits only. It does not include reserved bits content.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **364 of 530**

20.7.1 Interrupt Register

The Interrupt Register consists of 4 bits for the match interrupts and 4 bits for the capture interrupts. If an interrupt is generated then the corresponding bit in the IR will be high. Otherwise, the bit will be low. Writing a logic one to the corresponding IR bit will reset the interrupt. Writing a zero has no effect. The act of clearing an interrupt for a timer match also clears any corresponding DMA request. Writing a zero has no effect.

Table 373. Interrupt Register (IR, offset 0x00) bit description

Bit	Symbol	Description	Reset Value
0	MR0INT	Interrupt flag for match channel 0.	0
1	MR1INT	Interrupt flag for match channel 1.	0
2	MR2INT	Interrupt flag for match channel 2.	0
3	MR3INT	Interrupt flag for match channel 3.	0
4	CR0INT	Interrupt flag for capture channel 0 event.	0
5	CR1INT	Interrupt flag for capture channel 1 event.	0
6	CR2INT	Interrupt flag for capture channel 2 event.	0
7	CR3INT	Interrupt flag for capture channel 3 event.	0
31:6	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.2 Timer Control Register

The Timer Control Register (TCR) is used to control the operation of the Timer/Counter.

Table 374. Timer Control Register (TCR, offset 0x04) bit description

Bit	Symbol	Value	Description	Reset value
0	CEN		Counter enable.	0
		0	Disabled.The counters are disabled.	
		1	Enabled. The Timer Counter and Prescale Counter are enabled.	
1	CRST		Counter reset.	0
		0	Disabled. Do nothing.	
		1	Enabled. The Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of the APB bus clock. The counters remain reset until TCR[1] is returned to zero.	
31:2	-	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.3 Timer Counter register

The 32-bit Timer Counter register is incremented when the prescale counter reaches its terminal count. Unless it is reset before reaching its upper limit, the Timer Counter will count up through the value 0xFFFF FFFF and then wrap back to the value 0x0000 0000. This event does not cause an interrupt, but a match register can be used to detect an overflow if needed.

Table 375. Timer counter register (TC, offset 0x08) bit description

Bit	Symbol	Description	Reset value
31:0	TCVAL	Timer counter value.	0

20.7.4 Prescale register

The 32-bit Prescale register specifies the maximum value for the Prescale Counter.

Table 376. Timer prescale register (PR, offset 0x0C) bit description

Bit	Symbol	Description	Reset value
31:0	PRVAL	Prescale counter value.	0

20.7.5 Prescale Counter register

The 32-bit Prescale Counter controls division of the APB bus clock by some constant value before it is applied to the Timer Counter. This allows control of the relationship of the resolution of the timer versus the maximum time before the timer overflows. The Prescale Counter is incremented on every APB bus clock. When it reaches the value stored in the Prescale register, the Timer Counter is incremented and the Prescale Counter is reset on the next APB bus clock. This causes the Timer Counter to increment on every APB bus clock when PR = 0, every 2 APB bus clocks when PR = 1, etc.

Table 377. Timer prescale counter register (PC, offset 0x10) bit description

Bit	Symbol	Description	Reset value
31:0	PCVAL	Prescale counter value.	0

20.7.6 Match Control Register

The Match Control Register is used to control what operations are performed when one of the Match Registers matches the Timer Counter.

Table 378. Match Control Register (MCR, offset 0x14) bit description

Bit	Symbol	Description	Reset Value
0	MR0I	Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC. 0 = disabled. 1 = enabled.	0
1	MR0R	Reset on MR0: the TC will be reset if MR0 matches it. 0 = disabled. 1 = enabled.	0
2	MR0S	Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC. 0 = disabled. 1 = enabled.	0
3	MR1I	Interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC. 0 = disabled. 1 = enabled. 0 = disabled. 1 = enabled.	0
4	MR1R	Reset on MR1: the TC will be reset if MR1 matches it. 0 = disabled. 1 = enabled.	0
5	MR1S	Stop on MR1: the TC and PC will be stopped and TCR[0] will be set to 0 if MR1 matches the TC. 0 = disabled. 1 = enabled.	0
6	MR2I	Interrupt on MR2: an interrupt is generated when MR2 matches the value in the TC. 0 = disabled. 1 = enabled.	0
7	MR2R	Reset on MR2: the TC will be reset if MR2 matches it. 0 = disabled. 1 = enabled.	0
8	MR2S	Stop on MR2: the TC and PC will be stopped and TCR[0] will be set to 0 if MR2 matches the TC. 0 = disabled. 1 = enabled.	0
9	MR3I	Interrupt on MR3: an interrupt is generated when MR3 matches the value in the TC. 0 = disabled. 1 = enabled.	0
10	MR3R	Reset on MR3: the TC will be reset if MR3 matches it. 0 = disabled. 1 = enabled.	0

Table 378. Match Control Register (MCR, offset 0x14) bit description

Bit	Symbol	Description	Reset Value
11	MR3S	Stop on MR3: the TC and PC will be stopped and TCR[0] will be set to 0 if MR3 matches the TC. 0 = disabled. 1 = enabled.	0
23:12	-	Reserved. Read value is undefined, only zero should be written.	-
24	MR0RL	Reload MR0 with the contents of the Match 0 Shadow Register when the TC is reset to zero (either via a match event or a write to bit 1 of the TCR). 0 = disabled. 1 = enabled.	0
25	MR1RL	Reload MR1 with the contents of the Match 1 Shadow Register when the TC is reset to zero (either via a match event or a write to bit 1 of the TCR). 0 = disabled. 1 = enabled.	0
26	MR2RL	Reload MR2 with the contents of the Match 2 Shadow Register when the TC is reset to zero (either via a match event or a write to bit 1 of the TCR). 0 = disabled. 1 = enabled.	0
27	MR3RL	Reload MR3 with the contents of the Match 3 Shadow Register when the TC is reset to zero (either via a match event or a write to bit 1 of the TCR). 0 = disabled. 1 = enabled.	0
31:28	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.7 Match Registers

The Match register values are continuously compared to the Timer Counter value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the Timer Counter, or stop the timer. Actions are controlled by the settings in the MCR register.

If the associated MRxRL bit in the Match Control Register is set, the Match Register will be automatically reloaded with the current contents of its corresponding Match Shadow register whenever the TC is cleared to zero. This transfer will take place on the same clock edge that clocks the TC to zero.

Note: The TC is typically reset in response to an occurrence of a match on the Match Register being used to set the cycle counter rate. A reset can also occur due to software writing a 1 to bit 1 of the Timer Control Register.

Table 379. Match registers (MR[0:3], offset [0x18:0x24]) bit description

Bit	Symbol	Description	Reset value
31:0	MATCH	Timer counter match value.	0

20.7.8 Capture Control Register

The Capture Control Register is used to control whether one of the four Capture Registers is loaded with the value in the Timer Counter when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges. In the description below, "n" represents the timer number, 0 or 1.

Note: If Counter mode is selected for a particular CAP input in the CTCR, the 3 bits for that input in this register should be programmed as 000, but capture and/or interrupt can be selected for the other 3 CAP inputs.

Table 380. Capture Control Register (CCR, offset 0x28) bit description

Bit	Symbol	Description	Reset Value
0	CAP0RE	Rising edge of capture channel 0: a sequence of 0 then 1 causes CR0 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
1	CAP0FE	Falling edge of capture channel 0: a sequence of 1 then 0 causes CR0 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
2	CAP0I	Generate interrupt on channel 0 capture event: a CR0 load generates an interrupt.	0
3	CAP1RE	Rising edge of capture channel 1: a sequence of 0 then 1 causes CR1 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
4	CAP1FE	Falling edge of capture channel 1: a sequence of 1 then 0 causes CR1 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
5	CAP1I	Generate interrupt on channel 1 capture event: a CR1 load generates an interrupt.	0
6	CAP2RE	Rising edge of capture channel 2: a sequence of 0 then 1 causes CR2 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
7	CAP2FE	Falling edge of capture channel 2: a sequence of 1 then 0 causes CR2 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
8	CAP2I	Generate interrupt on channel 2 capture event: a CR2 load generates an interrupt.	0
9	CAP3RE	Rising edge of capture channel 3: a sequence of 0 then 1 causes CR3 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
10	CAP3FE	Falling edge of capture channel 3: a sequence of 1 then 0 causes CR3 to be loaded with the contents of TC. 0 = disabled. 1 = enabled.	0
11	CAP3I	Generate interrupt on channel 3 capture event: a CR3 load generates an interrupt.	0
31:12	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.9 Capture Registers

Each capture register is associated with one capture channel and may be loaded with the counter/timer value when a specified event occurs on the signal defined for that capture channel. The signal could originate from an external pin or from an internal source. The settings in the Capture Control Register register determine whether the capture function is enabled, and whether a capture event happens on the rising edge of the associated signal, the falling edge, or on both edges.

Table 381. Capture registers (CR[0:3], offsets [0x2C:0x38]) bit description

Bit	Symbol	Description	Reset value
31:0	CAP	Timer counter capture value.	0

20.7.10 External Match Register

The External Match Register provides both control and status of the external match pins. In the descriptions below, "n" represents the timer number, 0 or 1, and "m" represent a Match number, 0 through 3.

Match events for Match 0 and Match 1 in the timer can cause a DMA request, see Section 20.8.2.

If the match outputs are configured as PWM output, the function of the external match register is determined by the PWM rules (Section 20.8.1 "Rules for single edge controlled PWM outputs" on page 373).

Table 382. External match register (EMR, offset 0x3C) bit description

Bit	Symbol	Value	Description	Reset value
0	EM0	-	External Match 0. This bit reflects the state of output MAT0, whether or not this output is connected to a pin. When a match occurs between the TC and MR0, this bit can either toggle, go LOW, go HIGH, or do nothing, as selected by EMR[5:4]. This bit is driven to the MAT pins if the match function is selected via IOCON. 0 = LOW. 1 = HIGH.	0
1	EM1	-	External Match 1. This bit reflects the state of output MAT1, whether or not this output is connected to a pin. When a match occurs between the TC and MR1, this bit can either toggle, go LOW, go HIGH, or do nothing, as selected by EMR[7:6]. This bit is driven to the MAT pins if the match function is selected via IOCON. 0 = LOW. 1 = HIGH.	0
2	EM2	-	External Match 2. This bit reflects the state of output MAT2, whether or not this output is connected to a pin. When a match occurs between the TC and MR2, this bit can either toggle, go LOW, go HIGH, or do nothing, as selected by EMR[9:8]. This bit is driven to the MAT pins if the match function is selected via IOCON. 0 = LOW. 1 = HIGH.	0
3	EM3	-	External Match 3. This bit reflects the state of output MAT3, whether or not this output is connected to a pin. When a match occurs between the TC and MR3, this bit can either toggle, go LOW, go HIGH, or do nothing, as selected by MR[11:10]. This bit is driven to the MAT pins if the match function is selected via IOCON. 0 = LOW. 1 = HIGH.	0
5:4	EMC0		External Match Control 0. Determines the functionality of External Match 0.	0
		0x0	Do Nothing.	
		0x1	Clear. Clear the corresponding External Match bit/output to 0 (MAT0 pin is LOW if pinned out).	
		0x2	Set. Set the corresponding External Match bit/output to 1 (MAT0 pin is HIGH if pinned out).	
		0x3	Toggle. Toggle the corresponding External Match bit/output.	
7:6	EMC1		External Match Control 1. Determines the functionality of External Match 1.	0
		0x0	Do Nothing.	
		0x1	Clear. Clear the corresponding External Match bit/output to 0 (MAT1 pin is LOW if pinned out).	
		0x2	Set. Set the corresponding External Match bit/output to 1 (MAT1 pin is HIGH if pinned out).	
		0x3	Toggle. Toggle the corresponding External Match bit/output.	
9:8	EMC2		External Match Control 2. Determines the functionality of External Match 2.	0
		0x0	Do Nothing.	
		0x1	Clear. Clear the corresponding External Match bit/output to 0 (MAT2 pin is LOW if pinned out).	
		0x2	Set. Set the corresponding External Match bit/output to 1 (MAT2 pin is HIGH if pinned out).	
		0x3	Toggle. Toggle the corresponding External Match bit/output.	
11:10	EMC3		External Match Control 3. Determines the functionality of External Match 3.	0
		0x0	Do Nothing.	
		0x1	Clear. Clear the corresponding External Match bit/output to 0 (MAT3 pin is LOW if pinned out).	
		0x2	Set. Set the corresponding External Match bit/output to 1 (MAT3 pin is HIGH if pinned out).	
		0x3	Toggle. Toggle the corresponding External Match bit/output.	
31:12	-	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.11 Count Control Register

The Count Control Register (CTCR) is used to select between Timer and Counter mode, and in Counter mode to select the pin and edge(s) for counting.

When Counter Mode is chosen as a mode of operation, the CAP input (selected by the CTCR bits 3:2) is sampled on every rising edge of the APB bus clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized: rising edge, falling edge, either of edges or no changes in the level of the selected CAP input. Only if the identified event occurs and the event corresponds to the one selected by bits 1:0 in the CTCR register, will the Timer Counter register be incremented.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the APB bus clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input cannot exceed one half of the APB bus clock. Consequently, duration of the HIGH/LOWLOW levels on the same CAP input in this case cannot be shorter than 1/APB bus clock.

Bits 7:4 of this register are also used to enable and configure the capture-clears-timer feature. This feature allows for a designated edge on a particular CAP input to reset the timer to all zeros. Using this mechanism to clear the timer on the leading edge of an input pulse and performing a capture on the trailing edge, permits direct pulse-width measurement using a single capture input without the need to perform a subtraction operation in software.

Table 383. Count Control Register (CTCR, offset 0x70) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	CTMODE		Counter/Timer Mode This field selects which rising APB bus clock edges can increment Timer's Prescale Counter (PC), or clear PC and increment Timer Counter (TC). Timer Mode: the TC is incremented when the Prescale Counter matches the Prescale Register.	0
		0x0	Timer Mode. Incremented every rising APB bus clock edge.	
		0x1	Counter Mode rising edge. TC is incremented on rising edges on the CAP input selected by bits 3:2.	
		0x2	Counter Mode falling edge. TC is incremented on falling edges on the CAP input selected by bits 3:2.	
		0x3	Counter Mode dual edge. TC is incremented on both edges on the CAP input selected by bits 3:2.	
3:2	CINSEL		Count Input Select When bits 1:0 in this register are not 0b00, these bits select which CAP pin is sampled for clocking. **Note:** If Counter mode is selected in the CTCR, the 3 bits for that input in the Capture Control Register (CCR) must be programmed as 000. However, capture and/or interrupt can be selected for the other 3 CAPn inputs in the same timer.	0
		0x0	Channel 0. CAP0	
		0x1	Channel 1. CAP1	
		0x2	Channel 2. CAP2	
		0x3	Channel 3. CAP3, not used	
4	ENCC	-	Setting this bit to 1 enables clearing of the timer and the prescaler when the capture-edge event specified in bits 7:5 occurs.	0

UM11029

User manual **Rev. 1.6 — 8 December 2017** **370 of 530**

Table 383. Count Control Register (CTCR, offset 0x70) bit description

Bit	Symbol	Value	Description	Reset Value
7:5	SELCC		Edge select. When bit 4 is 1, these bits select which capture input edge will cause the timer and prescaler to be cleared. These bits have no effect when bit 4 is low. Note that different part number and package variations may provide different capture input pin functions.	0
		0x0	Channel 0 Rising Edge. Rising edge of the signal on capture channel 0 clears the timer (if bit 4 is set).	
		0x1	Channel 0 Falling Edge. Falling edge of the signal on capture channel 0 clears the timer (if bit 4 is set).	
		0x2	Channel 1 Rising Edge. Rising edge of the signal on capture channel 1 clears the timer (if bit 4 is set).	
		0x3	Channel 1 Falling Edge. Falling edge of the signal on capture channel 1 clears the timer (if bit 4 is set).	
		0x4	Channel 2 Rising Edge. Rising edge of the signal on capture channel 2 clears the timer (if bit 4 is set).	
		0x5	Channel 2 Falling Edge. Falling edge of the signal on capture channel 2 clears the timer (if bit 4 is set).	
		0x6	Channel 3 Rising Edge. Rising edge of the signal on capture channel 3 clears the timer (if bit 4 is set).	
		0x7	Channel 3 Falling Edge. Falling edge of the signal on capture channel 3 clears the timer (if bit 4 is set).	
31:8	-	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.12 PWM Control Register

The PWM Control Register is used to configure the match outputs as PWM outputs. Each match output can be independently set to perform either as PWM output or as match output whose function is controlled by the External Match Register (EMR).

A maximum of three single edge controlled PWM outputs can be selected on the MATn.2:0 outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

Table 384. PWM Control Register (PWMC, offset 0x74)) bit description

Bit	Symbol	Value	Description	Reset value
0	PWMEN0		PWM mode enable for channel0.	0
		0	Match. CTIMER_MAT0 is controlled by EM0.	
		1	PWM. PWM mode is enabled for CTIMER_MAT0.	
1	PWMEN1		PWM mode enable for channel1.	0
		0	Match. CTIMER_MAT01 is controlled by EM1.	
		1	PWM. PWM mode is enabled for CTIMER_MAT1.	
2	PWMEN2		PWM mode enable for channel2.	0
		0	Match. CTIMER_MAT2 is controlled by EM2.	
		1	PWM. PWM mode is enabled for CTIMER_MAT2.	

Table 384. PWM Control Register (PWMC, offset 0x74)) bit description

Bit	Symbol	Value	Description	Reset value
3	PWMEN3		PWM mode enable for channel3. **Note:** It is recommended to use match channel 3 to set the PWM cycle.	0
		0	Match. CTIMER_MAT3 is controlled by EM3.	
		1	PWM. PWM mode is enabled for CTIMER_MAT3.	
31:4	-	-	Reserved. Read value is undefined, only zero should be written.	-

20.7.13 Match Shadow Registers

The Match Shadow registers contain the values that the corresponding Match Registers are (optionally) reloaded with at the start of each new counter cycle. Typically, the match that causes the counter to be reset (and instigates the match reload) will also be programmed to generate an interrupt or DMA request. Software or the DMA engine will then have one full counter cycle to modify the contents of the Match Shadow Register(s) before the next reload occurs.

Table 385. Match shadow registers (MSR[0:3], offset [0x78:0x84]) bit description

Bit	Symbol	Description	Reset value
31:0	SHADOW	Timer counter match shadow value.	0x0

20.8 Functional description

Figure 37 shows a timer configured to reset the count and generate an interrupt on match. The prescaler is set to 2 and the match register set to 6. At the end of the timer cycle where the match occurs, the timer count is reset. This gives a full length cycle to the match value. The interrupt indicating that a match occurred is generated in the next clock after the timer reached the match value.

Figure 38 shows a timer configured to stop and generate an interrupt on match. The prescaler is again set to 2 and the match register set to 6. In the next clock after the timer reaches the match value, the timer enable bit in TCR is cleared, and the interrupt indicating that a match occurred is generated.
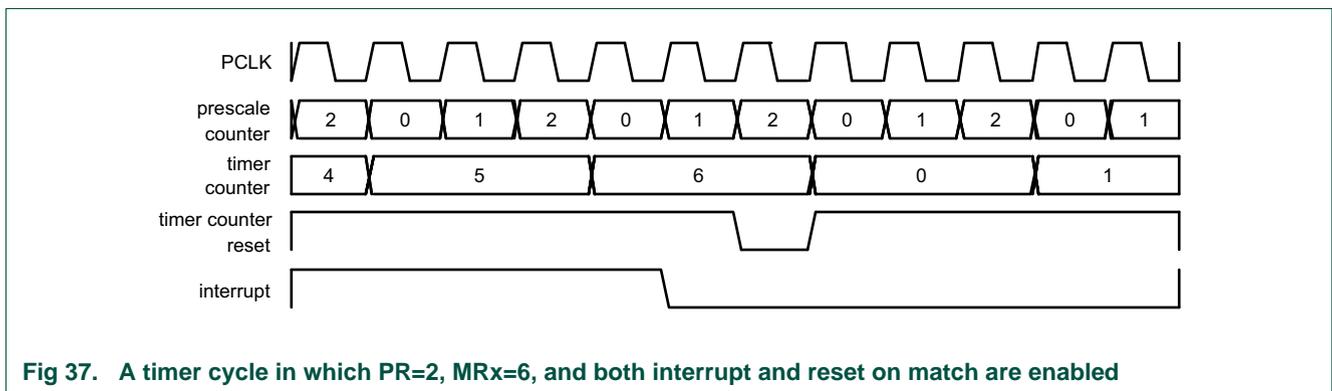
Fig 37. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled
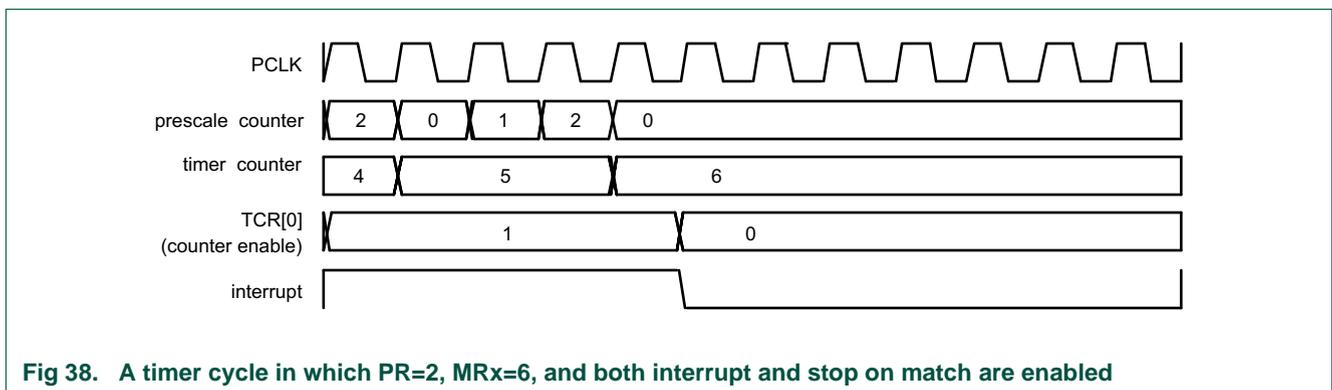
Fig 38. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled

20.8.1 Rules for single edge controlled PWM outputs

1. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value is equal to zero.

2. Each PWM output will go HIGH when its match value is reached. If no match occurs (i.e. the match value is greater than the PWM cycle length), the PWM output remains continuously LOW.

3. If a match value larger than the PWM cycle length is written to the match register, and the PWM signal is HIGH already, then the PWM signal will be cleared with the start of the next PWM cycle.

4. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to LOW on the next clock tick after the timer reaches the match value. Therefore, the PWM output will always consist of a one clock tick wide positive pulse with a period determined by the PWM cycle length (i.e. the timer reload value).

5. If a match register is set to zero, then the PWM output will go to HIGH the first time the timer goes back to zero and will stay HIGH continuously.

Note: When the match outputs are selected to perform as PWM outputs, the timer reset (MRnR) and timer stop (MRnS) bits in the Match Control Register MCR must be set to zero except for the match register setting the PWM cycle length. For this register, set the MRnR bit to one to enable the timer reset when the timer value matches the value of the corresponding match register.
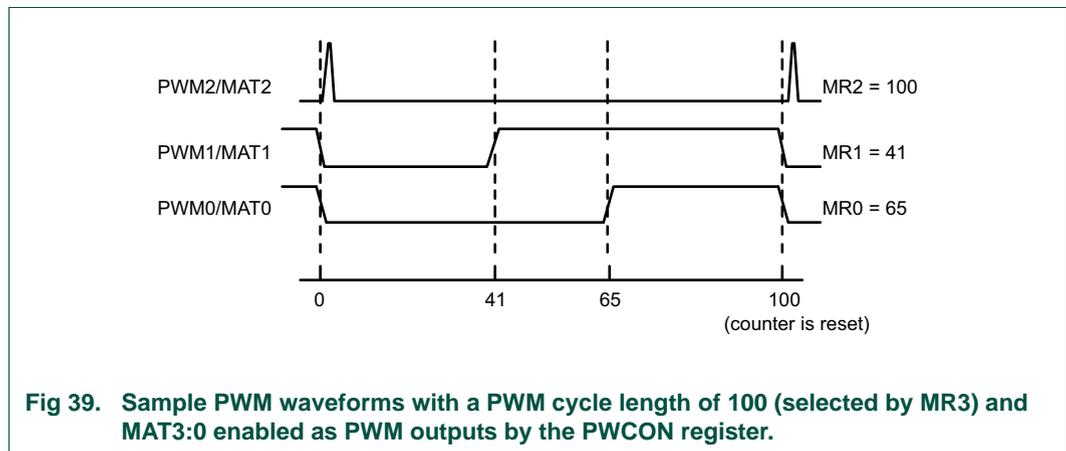
Fig 39.   Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWCON register.

20.8.2 DMA operation

DMA requests are generated by a match of the Timer Counter (TC) register value to either Match Register 0 (MR0) or Match Register 1 (MR1). This is not connected to the operation of the Match outputs controlled by the EMR register. Each match sets a DMA request flag, which is connected to the DMA controller. In order to have an effect, the DMA controller must be configured correctly.

When a timer is initially set up to generate a DMA request, the request may already be asserted before a match condition occurs. An initial DMA request may be avoided by having software write a one to the interrupt flag location, as if clearing a timer interrupt. See Section 20.7.1. A DMA request will be cleared automatically when it is acted upon by the DMA controller.

Note: because timer DMA requests are generated whenever the timer value is equal to the related Match Register value, DMA requests are always generated when the timer is running, unless the Match Register value is higher than the upper count limit of the timer. It is important not to select and enable timer DMA requests in the DMA block unless the timer is correctly configured to generate valid DMA requests.

21.1 How to read this chapter

The SCTimer/PWM is available on all LPC84x devices.

Remark: For a detailed description of SCTimer/PWM applications and code examples, see Ref. 5 "AN11538".

21.2 Features

- The SCTimer/PWM supports:
  - Five inputs: Four inputs with multiple connection options through the input MUX. The fifth input is from the clock generator sct_clk.
  - Seven outputs.
  - Eight match/capture registers.
  - Eight events.
  - Eight states.
- Counter/timer features:
  - Each SCTimer is configurable as two 16-bit counters or one 32-bit counter.
  - Counters clocked by system clock or selected input.
  - Configurable as up counters or up-down counters.
  - Configurable number of match and capture registers. Up to five match and capture registers total.
  - When there is a match and/or an input or output transition or level, create events to accomplish any or all of the following: stop, limit or halt the timer; change counting direction; set, clear or toggle outputs; change the state; capture the counter value; generate an interrupt or DMA request.
  - Counter value can be loaded into capture register triggered by a match or input/output toggle.
- PWM features:
  - Counters can be used in conjunction with match registers to toggle outputs and create time-proportional PWM signals.
  - PWM behavior can change based on the current state to create very complex, variable waveforms. In effect, states are a means of context switching for the entire SCTimer/PWM.
  - Up to 6 single-edge or dual-edge PWM outputs with independent duty cycle and common PWM cycle length.
- Event creation features:
  - The following conditions define an event: a counter match condition, an input (or output) condition such as an rising or falling edge or level, a combination of match and/or input/output condition. Event creation is qualified by States ("contexts").
  - In bidirectional mode, events can be enabled based on the count direction.

- – Selected events can limit, halt, start, or stop a counter or change its direction.
  - – Events trigger state changes, transitions, timer captures, interrupts, and DMA transactions.
  - – Match register 0 can be used as an automatic limit.
  - – In bidirectional mode, events can be enabled based on the count direction.
  - – Match events can be held until another qualifying event occurs.
- State control features:
  - – States have no pre-defined meaning. Entirely determined by the user. States provide a mechanism for context switching for the SCTimer/PWM including creation of complex state machines.
  - – The only function a State serves is to define which events can occur in that state.
  - – A state changes into some other state as a result of an event.
  - – Each event can be assigned to one or more states.
  - – State variable allows sequencing across multiple counter cycles.

21.3 Basic configuration

Configure the SCTimer/PWM as follows:

- Enable the clock to the SCTimer/PWM in the SYSAHBCLKCTRL register (Section 8.6.22) to enable the register interface and the peripheral clock.
- Clear the SCTimer/PWM peripheral reset using the PRESETCTRL register (Section 8.6.24).
- The SCTimer/PWM provides an interrupt to the NVIC, see Section 7.3.1.
- Use the INPUT MUX to connect the SCTimer/PWM inputs to external pins. See Table 386.
- Use the switch matrix registers to connect the SCTimer/PWM outputs to external pins. See Table 386.
- The SCTimer/PWM DMA request lines are connected to the DMA trigger inputs via the DMA_ITRIG_PINMUX registers. See Section 14.6.1.
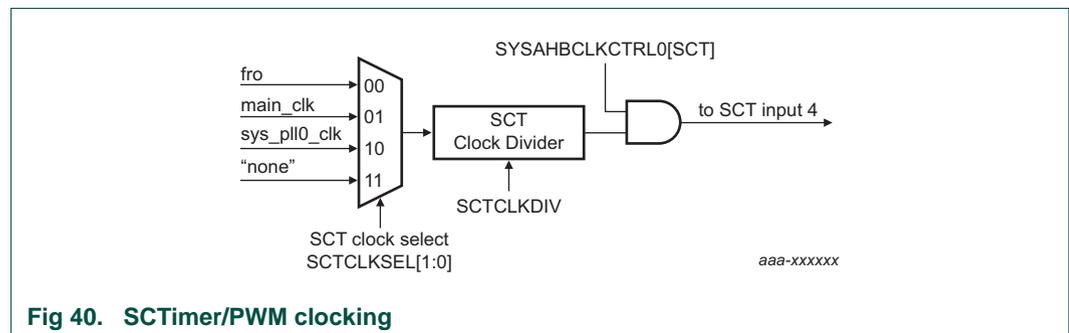
Fig 40.   SCTimer/PWM clocking

See Table 288 "SCT input mux registers 0 to 3 (SCT0_INMUX[0:3], address 0x4002 C020 (SCT0_INMUX0) to 0x4002 C02C (SCT0_INMUX3)) bit description" to configure the SCTimer/PWM input multiplexers.

Fig 41. SCTimer/PWM connections
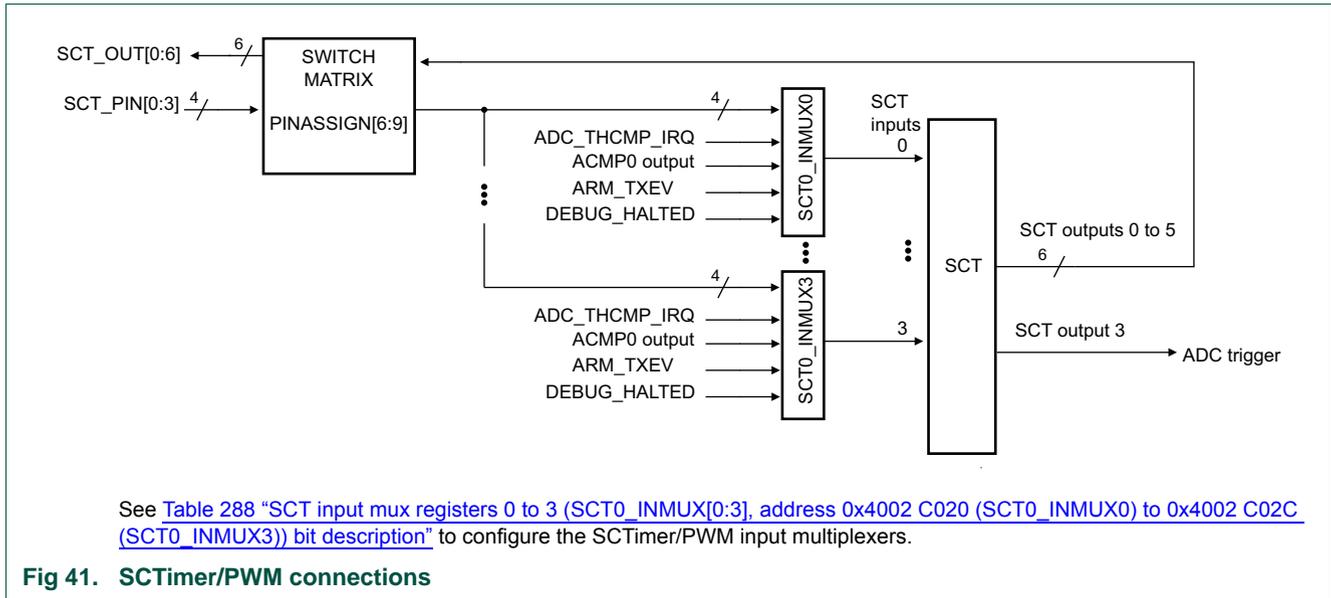
21.4 Pin description

Remark: Availability of inputs or outputs related to a particular peripheral function might be package dependent.

SCTimer/PWM input signals may come from external pins and internal signals. These are connected to the SCTimer/PWM via input multiplexers (Table 386).

SCTimer/PWM outputs can be routed to multiple places and can be connected to both a pin and an ADC trigger at the same time. See Table 386.

Table 386. SCTimer/PWM pin description

Function	Type	Connect to	Use register	Reference	Description
SCT input 0, SCT input 1, SCT input 2, SCT input 3, SCT input 4	external to pins or internal	one of the following: SCT_PIN[0:3] through the switch matrix, or internally to the analog comparator output, ADC threshold compare interrupt, ARM TXEV	SCT0_PERIPHINMUX PINASSIGN6 PINASSIGN7	Table 288, Table 188, Table 189	SCT capture and event inputs.
SCT_OUT0	external to pins	pins through the switch matrix	PINASSIGN7	Table 188	SCT match and PWM output.
SCT_OUT1, SCT_OUT2, SCT_OUT3, SCT_OUT4	external to pins and internal	pins through the switch matrix; to ADC trigger	PINASSIGN8	Table 189	SCT match and PWM outputs. SCT_OUT3 and SCT_OUT4 can be selected as ADC input trigger.
SCT_OUT5	external to pin	pin through the switch matrix	PINASSIGN9	Table 189	SCT match and PWM output.
SCT_OUT6	external to pin	pin through the switch matrix	PINASSIGN9	Table 189	SCT match and PWM output.

21.5 General description

The SCTimer/PWM is a powerful, flexible timer module capable of creating complex PWM waveforms and performing other advanced timing and control operations with minimal or no CPU intervention.

The SCTimer/PWM can operate as a single 32-bit counter or as two independent, 16-bit counters in unidirectional or bidirectional mode. As with most timers, the SCTimer/PWM supports a selection of match registers against which the count value can be compared, and capture registers where the current count value can be recorded when some pre-defined condition is detected.

An additional feature contributing to the versatility of the SCTimer/PWM is the concept of "events". The SCTimer/PWM module supports multiple separate events that can be defined by the user based on some combination of parameters including a match on one of the match registers, and/or a transition on one of the SCTimer/PWM inputs or outputs, the direction of count, and other factors.

Every action that the SCTimer/PWM block can perform occurs in direct response to one of these user-defined events without any software overhead. Any event can be enabled to:

- Start, stop, or halt the counter.
- Limit the counter which means to clear the counter in unidirectional mode or change its direction in bidirectional mode.
- Set, clear, or toggle any SCTimer/PWM output.
- Force a capture of the count value into any capture registers.
- Generate an interrupt or DMA request.

The SCTimer/PWM allows the user to group and filter events, thereby selecting some events to be enabled together while others are disabled. A group of enabled and disabled events can be described as a state, and several states with different sets of enabled and disabled events are allowed. Changing from one state to another is event driven as well and can therefore happen without software intervention. By defining these states, the SCTimer/PWM provides the means to run entire state machines in hardware with any desired level of complexity to accomplish complex waveform and timing tasks.

 In a simple system such as a classical timer/counter with capture and match capabilities. all events that could cause the timer to capture the timer value or toggle a match output are enabled and remain enabled at all times while the counter is running. In this case, no events are filtered and the system is described by one state that does not change. This is the default configuration of the SCTimer/PWM.

In a more complex system, two states could be set up that allow some events in one state and not in the other. An event itself, enabled in both states, can then be used, to move from one state to the other and back while filtering out events in either state. In such a two-state system different waveforms at the SCTimer/PWM output can be created depending on the event history. Changing between states is event-driven and happens without any intervention by the CPU.

Formally, the SCTimer/PWM can be programmed as state machine generator. The ability to perform switching between groups of events provides the SCTimer/PWM the unique capability to be utilized as a highly complex State Machine engine. Events identify the

occurrence of conditions that warrant state changes and determine the next state to move to. This provides an extremely powerful control tool - particularly when the SCTimer/PWM inputs and outputs are connected to other on-chip resources (comparators, ADC triggers, other timers etc.) in addition to general-purpose I/O.

In addition to events and states, the SCTimer/PWM provides other enhanced features:

- Four alternative clocking modes including a fully asynchronous mode.
- Selection of any SCTimer/PWM input as a clock source or a clock gate.
- Capability of selecting a "greater-than-or-equal-to" match condition for the purpose of event generation.

Fig 42. SCTimer/PWM block diagram
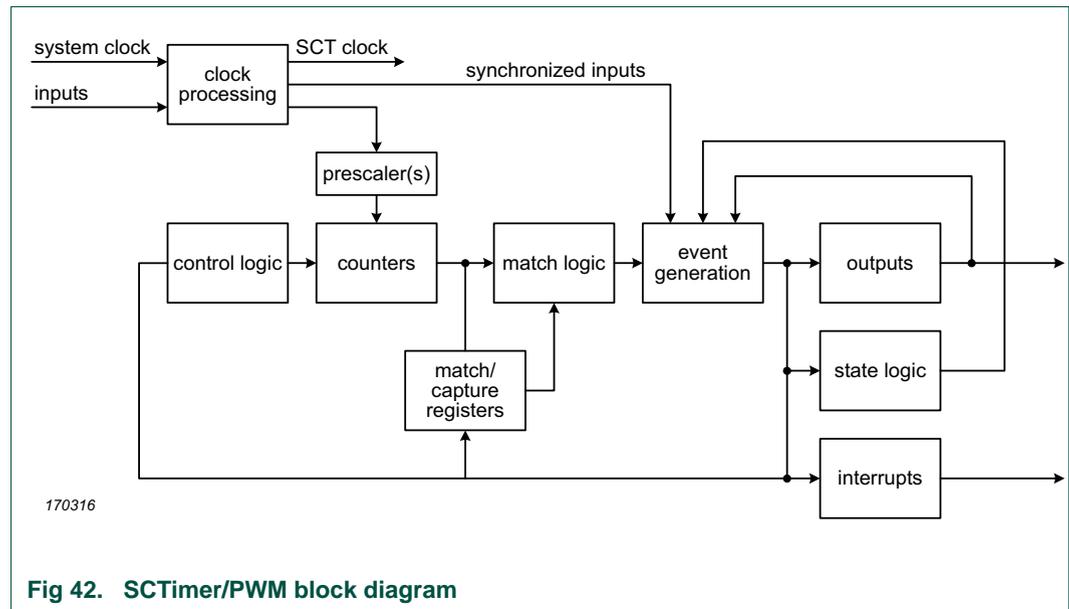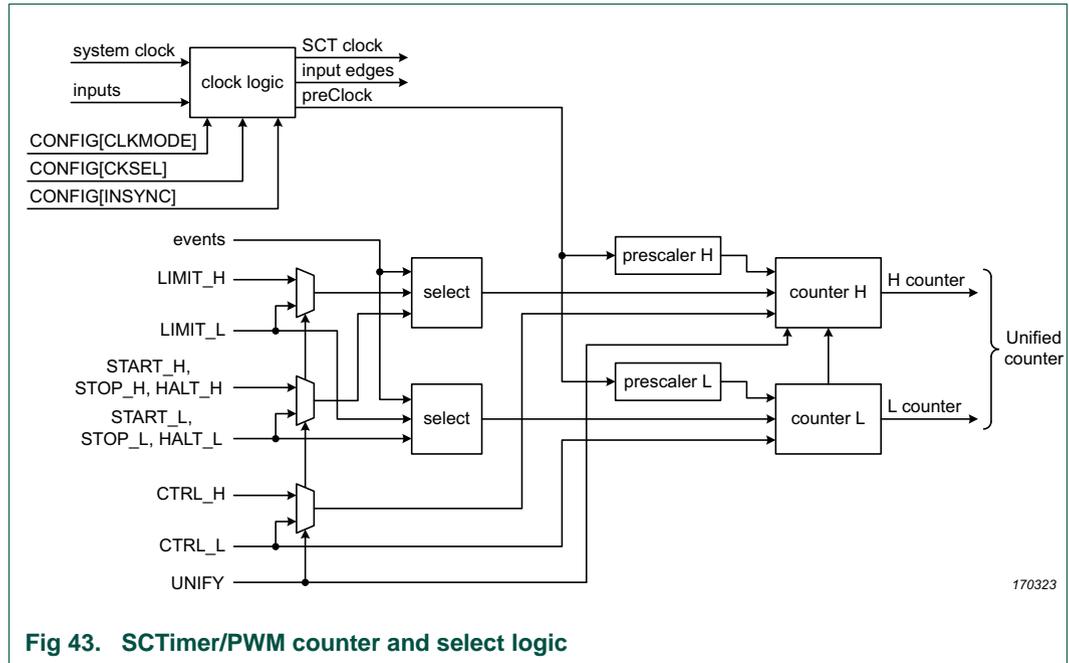
Fig 43. SCTimer/PWM counter and select logic

Remark: In this chapter, the term bus error indicates an SCTimer/PWM response that makes the processor take an exception.

21.6 Register description

The register addresses of the SCTimer/PWM are shown in Table 387. For most of the SCTimer/PWM registers, the register function depends on the setting of certain other register bits:

1. The UNIFY bit in the CONFIG register determines whether the SCTimer/PWM is used as one 32-bit register (for operation as one 32-bit counter/timer) or as two 16-bit counter/timers named L and H. The setting of the UNIFY bit is reflected in the register map:

   – UNIFY = 1: Only one register is used (for operation as one 32-bit counter/timer).

   – UNIFY = 0: Access the L and H registers by a 32-bit read or write operation or can be read or written to individually (for operation as two 16-bit counter/timers).

   Typically, the UNIFY bit is configured by writing to the CONFIG register before any other registers are accessed.

2. The REGMODEn bits in the REGMODE register determine whether each set of Match/Capture registers uses the match or capture functionality:

   – REGMODEn = 0: Registers operate as match and reload registers.

   – REGMODEn = 1: Registers operate as capture and capture control registers.

Table 387. Register overview: SCTimer/PWM (base address 0x5000 4000)

Name	Access	Address offset	Description	Reset value	Section
CONFIG	R/W	0x000	SCT configuration register	0x0000 7E00	21.6.2
CTRL	R/W	0x004	SCT control register	0x0004 0004	21.6.3
CTRL_L	R/W	0x004	SCT control register low counter 16-bit	0x0004 0004	21.6.3

Table 387. Register overview: SCTimer/PWM (base address 0x5000 4000) …*continued*

Name	Access	Address offset	Description	Reset value	Section
CTRL_H	R/W	0x006	SCT control register high counter 16-bit	0x0004 0004	21.6.3
LIMIT	R/W	0x008	SCT limit event select register	0x0000 0000	21.6.4
LIMIT_L	R/W	0x008	SCT limit event select register low counter 16-bit	0x0000 0000	21.6.4
LIMIT_H	R/W	0x00A	SCT limit event select register high counter 16-bit	0x0000 0000	21.6.4
HALT	R/W	0x00C	SCT halt events elect register	0x0000 0000	21.6.5
HALT_L	R/W	0x00C	SCT halt event select register low counter 16-bit	0x0000 0000	21.6.5
HALT_H	R/W	0x00E	SCT halt event select register high counter 16-bit	0x0000 0000	21.6.5
STOP	R/W	0x010	SCT stop event select register	0x0000 0000	21.6.6
STOP_L	R/W	0x010	SCT stop event select register low counter 16-bit	0x0000 0000	21.6.6
STOP_H	R/W	0x012	SCT stop event select register high counter 16-bit	0x0000 0000	21.6.6
START	R/W	0x014	SCT start event select register	0x0000 0000	21.6.7
START_L	R/W	0x014	SCT start event select register low counter 16-bit	0x0000 0000	21.6.7
START_H	R/W	0x016	SCT start event select register high counter 16-bit	0x0000 0000	21.6.7
COUNT	R/W	0x040	SCT counter register	0x0000 0000	21.6.8
COUNT_L	R/W	0x040	SCT counter register low counter 16-bit	0x0000 0000	21.6.8
COUNT_H	R/W	0x042	SCT counter register high counter 16-bit	0x0000 0000	21.6.8
STATE	R/W	0x044	SCT state register	0x0000 0000	21.6.9
STATE_L	R/W	0x044	SCT state register low counter 16-bit	0x0000 0000	21.6.9
STATE_H	R/W	0x046	SCT state register high counter 16-bit	0x0000 0000	21.6.9
INPUT	RO	0x048	SCT input register	0x0000 0000	21.6.10
REGMODE	R/W	0x04C	SCT match/capture mode register	0x0000 0000	21.6.11
REGMODE_L	R/W	0x04C	SCT match/capture mode register low counter 16-bit	0x0000 0000	21.6.11
REGMODE_H	R/W	0x04E	SCT match/capture registers mode register high counter 16-bit	0x0000 0000	21.6.11
OUTPUT	R/W	0x050	SCT output register	0x0000 0000	21.6.12
OUTPUTDIRCTRL	R/W	0x054	SCT output counter direction control register	0x0000 0000	21.6.13
RES	R/W	0x058	SCT conflict resolution register	0x0000 0000	21.6.14
DMAREQ0	R/W	0x05C	SCT DMA request 0 register	0x0000 0000	21.6.15
DMAREQ1	R/W	0x060	SCT DMA request 1 register	0x0000 0000	21.6.15
EVEN	R/W	0x0F0	SCT event interrupt enable register	0x0000 0000	21.6.16
EVFLAG	R/W	0x0F4	SCT event flag register	0x0000 0000	21.6.17
CONEN	R/W	0x0F8	SCT conflict interrupt enable register	0x0000 0000	21.6.18
CONFLAG	R/W	0x0FC	SCT conflict flag register	0x0000 0000	21.6.19
MATCH0 to MATCH7	R/W	0x100 to 0x11C	SCT match value register of match channels 0 to 7; REGMODE0 to REGMODE7 = 0	0x0000 0000	21.6.20
MATCH0_L to MATCH7_L	R/W	0x100 to 0x11C	SCT match value register of match channels 0 to 7; low counter 16-bit; REGMODE0_L to REGMODE7_L = 0	0x0000 0000	21.6.20
MATCH0_H to MATCH7_H	R/W	0x102 to 0x11E	SCT match value register of match channels 0 to 7; high counter 16-bit; REGMODE0_H to REGMODE7_H = 0	0x0000 0000	21.6.20

UM11029

User manual **Rev. 1.6 — 8 December 2017** **381 of 530**

Table 387. Register overview: SCTimer/PWM (base address 0x5000 4000) *…continued*

Name	Access	Address offset	Description	Reset value	Section
CAP0 to CAP7	R/W	0x100 to 0x11C	SCT capture register of capture channel 0 to 7; REGMODE0 to REGMODE7 = 1	0x0000 0000	21.6.21
CAP0_L to CAP7_L	R/W	0x100 to 0x11C	SCT capture register of capture channel 0 to 7; low counter 16-bit; REGMODE0_L to REGMODE7_L = 1	0x0000 0000	21.6.21
CAP0_H to CAP7_H	R/W	0x102 to 0x11E	SCT capture register of capture channel 0 to 7; high counter 16-bit; REGMODE0_H to REGMODE7_H = 1	0x0000 0000	21.6.21
MATCHREL0 to MATCHREL7	R/W	0x200 to 0x21C	SCT match reload value register 0 to 7; REGMODE0 = 0 to REGMODE7 = 0	0x0000 0000	21.6.22
MATCHREL0_L to MATCHREL7_L	R/W	0x200 to 0x21C	SCT match reload value register 0 to 7; low counter 16-bit; REGMODE0_L = 0 to REGMODE7_L = 0	0x0000 0000	21.6.22
MATCHREL0_H to MATCHREL7_H	R/W	0x202 to 0x21E	SCT match reload value register 0 to 7; high counter 16-bit; REGMODE0_H = 0 to REGMODE7_H = 0	0x0000 0000	21.6.22
CAPCTRL0 to CAPCTRL7	R/W	0x200 to 0x21C	SCT capture control register 0 to 7; REGMODE0 = 1 to REGMODE7 = 1	0x0000 0000	21.6.23
CAPCTRL0_L to CAPCTRL7_L	R/W	0x200 to 0x21C	SCT capture control register 0 to 7; low counter 16-bit; REGMODE0_L = 1 to REGMODE7_L = 1	0x0000 0000	21.6.23
CAPCTRL0_H to CAPCTRL7_H	R/W	0x202 to 0x21E	SCT capture control register 0 to 7; high counter 16-bit; REGMODE0 = 1 to REGMODE7 = 1	0x0000 0000	21.6.23
EV0_STATE	R/W	0x300	SCT event state register 0	0x0000 0000	21.6.24
EV0_CTRL	R/W	0x304	SCT event control register 0	0x0000 0000	21.6.25
EV1_STATE	R/W	0x308	SCT event state register 1	0x0000 0000	21.6.24
EV1_CTRL	R/W	0x30C	SCT event control register 1	0x0000 0000	21.6.25
EV2_STATE	R/W	0x310	SCT event state register 2	0x0000 0000	21.6.24
EV2_CTRL	R/W	0x314	SCT event control register 2	0x0000 0000	21.6.25
EV3_STATE	R/W	0x318	SCT event state register 3	0x0000 0000	21.6.24
EV3_CTRL	R/W	0x31C	SCT event control register 3	0x0000 0000	21.6.25
EV4_STATE	R/W	0x320	SCT event state register 4	0x0000 0000	21.6.24
EV4_CTRL	R/W	0x324	SCT event control register4	0x0000 0000	21.6.25
EV5_STATE	R/W	0x328	SCT event state register 5	0x0000 0000	21.6.24
EV5_CTRL	R/W	0x32C	SCT event control register 5	0x0000 0000	21.6.25
EV6_STATE	R/W	0x330	SCT event state register 6	0x0000 0000	21.6.24
EV6_CTRL	R/W	0x334	SCT event control register 6	0x0000 0000	21.6.25
EV7_STATE	R/W	0x338	SCT event state register 7	0x0000 0000	21.6.24
EV7_CTRL	R/W	0x33C	SCT event control register 7	0x0000 0000	21.6.25
OUT0_SET	R/W	0x500	SCT output 0 set register	0x0000 0000	21.6.26
OUT0_CLR	R/W	0x504	SCT output 0 clear register	0x0000 0000	21.6.27
OUT1_SET	R/W	0x508	SCT output 1 set register	0x0000 0000	21.6.26
OUT1_CLR	R/W	0x50C	SCT output 1 clear register	0x0000 0000	21.6.27
OUT2_SET	R/W	0x510	SCT output 2 set register	0x0000 0000	21.6.26
OUT2_CLR	R/W	0x514	SCT output 2 clear register	0x0000 0000	21.6.27

Table 387. Register overview: SCTimer/PWM (base address 0x5000 4000) *…continued*

Name	Access	Address offset	Description	Reset value	Section
OUT3_SET	R/W	0x518	SCT output 3 set register	0x0000 0000	21.6.26
OUT3_CLR	R/W	0x51C	SCT output 3 clear register	0x0000 0000	21.6.27
OUT4_SET	R/W	0x520	SCT output 4 set register	0x0000 0000	21.6.26
OUT4_CLR	R/W	0x524	SCT output 4 clear register	0x0000 0000	21.6.27
OUT5_SET	R/W	0x528	SCT output 5 set register	0x0000 0000	21.6.26
OUT5_CLR	R/W	0x52C	SCT output 5 clear register	0x0000 0000	21.6.27
OUT6_SET	R/W	0x530	SCT output 6 set register	0x0000 0000	21.6.26
OUT6_CLR	R/W	0x534	SCT output 6 clear register	0x0000 0000	21.6.27

21.6.1 Register functional grouping

Most SCTimer/PWM registers either configure an event or select an event for a specific action of the counter (or counters) and outputs. Figure 44 shows the registers and register bits that can be configured for each event.
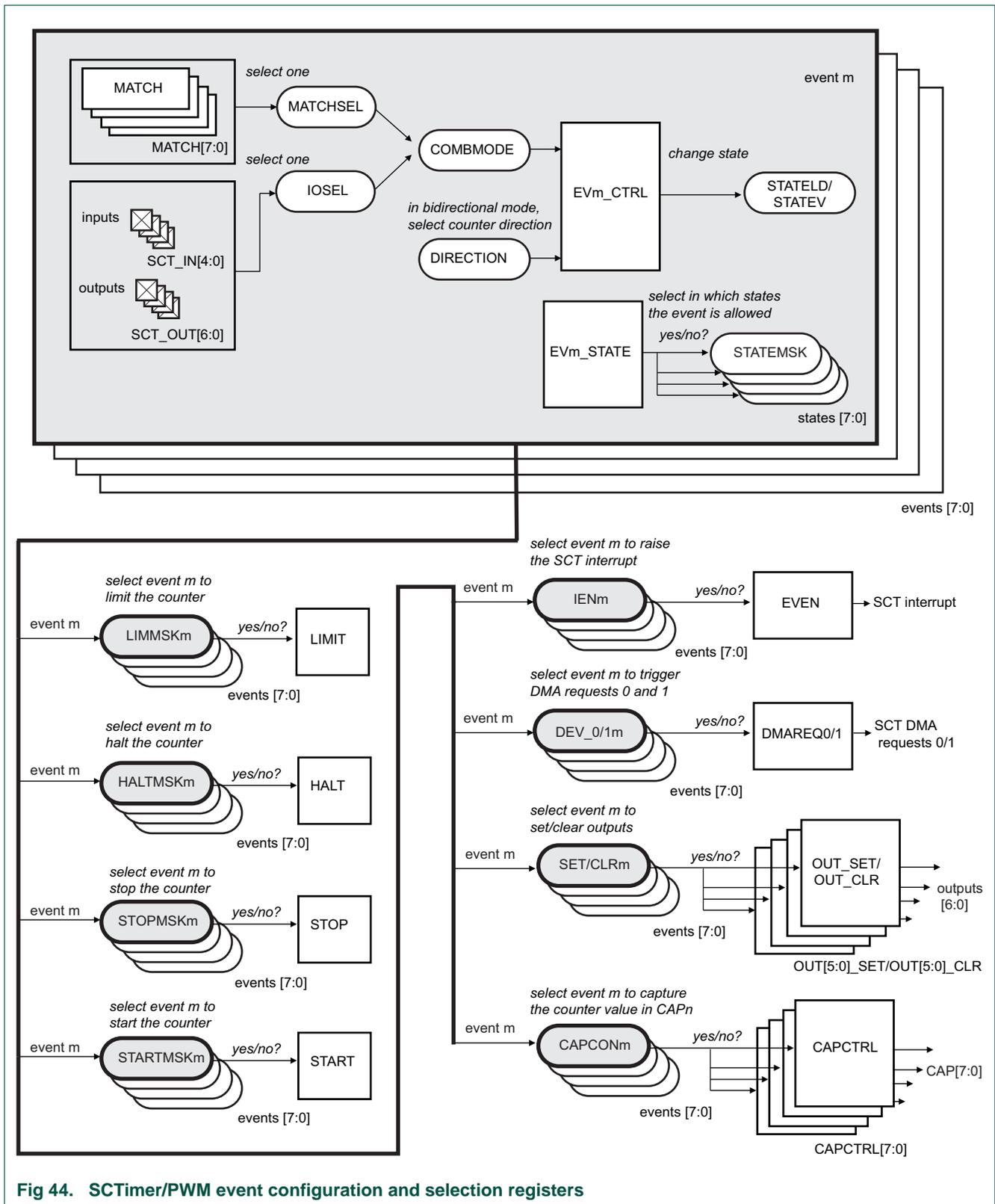
Fig 44. SCTimer/PWM event configuration and selection registers

UM11029

User manual **Rev. 1.6 — 8 December 2017** **384 of 530**

21.6.1.1 Counter configuration and control registers

The SCTimer/PWM contains two registers for configuring the SCTimer/PWM and monitor and control its operation by software.

- The configuration register (CONFIG) configures the SCTimer/PWM in single, 32-bit counter mode or in dual, 16-bit counter mode, configures the clocking and clock synchronization, and configures automatic limits and the use of reload registers.
- The control register (CTRL) allows to monitor and set the counter direction, and to clear, start, stop, or halt the 32-bit counter or each individual 16-bit counter if in dual-counter mode.

21.6.1.2 Event configuration registers

Each event is associated with two registers:

- One EVn_CTRL register per event to define what triggers the event.
- One EVn_STATE register per event to enable the event.

21.6.1.3 Match and capture registers

The SCTimer/PWM includes a set of registers to store the match or capture values. Each match register is associated with a match reload register which automatically reloads the match register at the beginning of each counter cycle. This register group includes the following registers:

- One REGMODE register per match/capture register to configure each match/capture register for either storing a match value or a capture value.
- A set of match/capture registers with each register, depending on the setting of REGMODE, either storing a match value or a counter value.
- One reload register for each match register.

21.6.1.4 Event select registers for the counter operations

This group contains the registers that select the events which affect the counter. Counter actions are limit, halt, and start or stop and apply to the unified counter or to the two 16-bit counters. Also included is the counter register with the counter value, or values in the dual-counter set-up. This register group includes the following registers:

- LIMIT selects the events that limit the counter.
- START and STOP select events that start or stop the counter.
- HALT selects events that halt the counter: HALT
- COUNT contains the counter value.

The LIMIT, START, STOP, and HALT registers each contain one bit per event that selects for each event whether the event limits, stops, starts, or halts the counter, or counters in dual-counter mode.

In the dual-counter mode, the events can be selected independently for each counter.

21.6.1.5 Event select registers for setting or clearing the outputs

This group contains the registers that select the events which affect the level of each SCTimer/PWM output. Also included are registers to manage conflicts that occur when events try to set or clear the same output. This register group includes the following registers:

- One OUTn_SET register for each output to select the events which set the output.
- One OUTn_CLR register for each output to select the events which clear the output.
- The conflict resolution register which defines an action when more than one event try to control an output at the same time.
- The conflict flag and conflict interrupt enable registers that monitor interrupts arising from output set and clear conflicts.
- The output direction control register that interchanges the set and clear output operation caused by an event in bidirectional mode.

The OUTn_SET and OUTn_CLR registers each contain one bit per event that selects whether the event changes the state a given output n.

In the dual-counter mode, the events can be selected independently for each output.

21.6.1.6 Event select registers for capturing a counter value

This group contains registers that select events which capture the counter value and store it in one of the CAP registers. Each capture register m has one associated CAPCTRLm register which in turn selects the events to capture the counter value.

21.6.1.7 Event select register for initiating DMA transfers

One register is provided for each of the two DMA requests to select the events that can trigger a DMA request.

The DMAREQn register contain one bit for each event that selects whether this event triggers a DMA request. An additional bit enables the DMA trigger when the match registers are reloaded.

21.6.1.8 Interrupt handling registers

The following registers provide flags that are set by events and select the events that when they occur request an interrupt.

- The event flag register provides one flag for each event that is set when the event occurs.
- The event flag interrupt enable register provides one bit for each event to be enabled for the SCTimer/PWM interrupt.

21.6.1.9 Registers for controlling SCTimer/PWM inputs and outputs by software

Two registers are provided that allow software (as opposed to events) to set input and outputs of the SCTimer/PWM:

- The SCTimer/PWM input register to read the state of any of the SCTimer/PWM inputs.
- The SCTimer/PWM output register to set or clear any of the SCTimer/PWM outputs or to read the state of the outputs.

21.6.2 SCTimer/PWM configuration register

This register configures the overall operation of the SCTimer/PWM. Write to this register before any other registers. Only word-writes are permitted to this register. Attempting to write a half-word value results in a bus error.

Table 388. SCTimer/PWM configuration register (CONFIG, offset 0x000) bit description

Bit	Symbol	Value	Description	Reset value
0	UNIFY		SCTimer/PWM operation	0
		0	The SCTimer/PWM operates as two 16-bit counters named COUNTER_L and COUNTER_H.	
		1	The SCTimer/PWM operates as a unified 32-bit counter.	
2:1	CLKMODE		SCTimer/PWM clock mode	0
		0x0	System Clock Mode. The system clock clocks the entire SCTimer/PWM module including the counter(s) and counter prescalers.	
		0x1	Sampled System Clock Mode. The system clock clocks the SCTimer/PWM module, but the counter and prescalers are only enabled to count when the designated edge is detected on the input selected by the CKSEL field. The minimum width of the positive and negative phases of the clock input must each be greater than one full period of the bus/system clock. This mode is the high-performance, sampled-clock mode.	
		0x2	SCTimer/PWM Input Clock Mode. The input/edge selected by the CKSEL field clocks the SCTimer/PWM module, including the counters and prescalers, after first being synchronized to the system clock. The minimum width of the positive and negative phases of the clock input must each be greater than one full period of the bus/system clock.	
		0x3	Asynchronous Mode. The entire SCTimer/PWM module is clocked directly by the input/edge selected by the CKSEL field. In this mode, the SCTimer/PWM outputs are switched synchronously to the SCTimer/PWM input clock - not the system clock. The input clock rate must be at least half the system clock rate and can be the same or faster than the system clock.	
6:3	CKSEL		SCTimer/PWM clock select. The specific functionality of the designated input/edge is dependent on the CLKMODE bit selection in this register.	0
		0x0	Rising edges on input 0.	
		0x1	Falling edges on input 0.	
		0x2	Rising edges on input 1.	
		0x3	Falling edges on input 1.	
		0x4	Rising edges on input 2.	
		0x5	Falling edges on input 2.	
		0x6	Rising edges on input 3.	
		0x7	Falling edges on input 3.	
		0x8	Rising edges on input 4. SCT clock selected by the SYSCON SCTCLKSEL and SCTCLKDIV registers.	
		0x9	Falling edges on input 4. SCT clock selected by the SYSCON SCTCLKSEL and SCTCLKDIV registers.	
7	NORELAOD_L	-	A 1 in this bit prevents the lower match registers from being reloaded from their respective reload registers. Setting this bit eliminates the need to write to the reload registers MATCHREL if the match values are fixed. Software can write to set or clear this bit at any time. This bit applies to both the higher and lower registers when the UNIFY bit is set.	0

Table 388. SCTimer/PWM configuration register (CONFIG, offset 0x000) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
8	NORELOAD_H	-	A 1 in this bit prevents the higher match registers from being reloaded from their respective reload registers. Setting this bit eliminates the need to write to the reload registers MATCHREL if the match values are fixed. Software can write to set or clear this bit at any time. This bit is not used when the UNIFY bit is set.	0
12:9	INSYNC	-	Synchronization for input N (bit 9 = input 0, bit 10 = input 1,..., bit 12 = input 3); all other bits are reserved. A 1 in one of these bits subjects the corresponding input to synchronization to the SCTimer/PWM clock, before it is used to create an event. This synchronization injects a two SCT-clock delay in the input path. Clearing this bit bypasses synchronization on the corresponding input. This bit may be cleared for faster input response time if both of the following conditions are met (for all Clock Modes): The corresponding input is already synchronous to the SCTimer/PWM clock. The SCTimer/PWM clock frequency does not exceed 100 MHz. Note: The SCTimer/PWM clock is the bus/system clock for CKMODE 0-2 or the selected, asynchronous input clock for CKMODE3. Alternatively, for CKMODE2 only, it is also allowable to bypass synchronization if both of the following conditions are met: The corresponding input is synchronous to the designated CKMODE2 input clock. The CKMODE2 input clock frequency is less than one-third the frequency of the bus/system clock.	1
16:13	-	-	Reserved.	-
17	AUTOLIMIT_L	-	This bit applies to the lower registers when the UNIFY bit = 0, and both the higher and lower registers when the UNIFY bit is set. Software can write to set or clear this bit at any time. A one in this bit causes a match on match register 0 to be treated as a de-facto LIMIT condition without the need to define an associated event. As with any LIMIT event, this automatic limit causes the counter to be cleared to zero in unidirectional mode or to change the direction of count in bi-directional mode.	0
18	AUTOLIMIT_H	-	This bit applies to the upper registers when the UNIFY bit = 0, and is not used when the UNIFY bit is set. Software can write to set or clear this bit at any time. A one in this bit will cause a match on match register 0 to be treated as a de-facto LIMIT condition without the need to define an associated event. As with any LIMIT event, this automatic limit causes the counter to be cleared to zero in unidirectional mode or to change the direction of count in bi-directional mode.	0
31:19	-	-	Reserved	-

21.6.3 SCTimer/PWM control register

If bit UNIFY = 1 in the CONFIG register, only the _L bits are used.

If bit UNIFY = 0 in the CONFIG register, this register can be written to as two registers CTRL_L and CTRL_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

All bits in this register can be written to when the counter is stopped or halted. When the counter is running, the only bits that can be written are STOP or HALT. (Other bits can be written in a subsequent write after HALT is set to 1.)

Remark: If CLKMODE = 0x3 is selected, wait at least 12 system clock cycles between a write access to the H, L or unified version of this register and the next write access. This restriction does not apply when writing to the HALT bit or bits and then writing to the CTRL register again to restart the counters - for example because software must update the MATCH register, which is only allowed when the counters are halted.

Remark: If the SCTimer/PWM is operating as two 16-bit counters, events can only modify the state of the outputs when neither counter is halted. This is true regardless of what triggered the event.

Table 389. SCTimer/PWM control register (CTRL, offset 0x004) bit description

Bit	Symbol	Value	Description	Reset value
0	DOWN_L	-	This read-only bit is 1 when the L or unified counter is counting down. Hardware sets this bit when the counter is counting up, counter limit occurs, and BIDIR = 1.Hardware clears this bit when the counter is counting down and a limit condition occurs or when the counter reaches 0.	0
1	STOP_L	-	When this bit is 1 and HALT is 0, the L or unified counter does not run, but I/O events related to the counter can occur. If a designated start event occurs, this bit is cleared and counting resumes.	0
2	HALT_L	-	When this bit is 1, the L or unified counter does not run and no events can occur. A reset sets this bit. When the HALT_L bit is one, the STOP_L bit is cleared. It is possible to remove the halt condition while keeping the SCTimer/PWM in the stop condition (not running) with a single write to this register to simultaneously clear the HALT bit and set the STOP bit. **Remark:** Once set, only software can clear this bit to restore counter operation. This bit is set on reset.	1
3	CLRCTR_L	-	When the counter is halted (not just stopped), writing a 1 to this bit will clear the L or unified counter. This bit always reads as 0.	0
4	BIDIR_L		L or unified counter direction select	0
		0	Up. The counter counts up to a limit condition, then is cleared to zero.	
		1	Up-down. The counter counts up to a limit, then counts down to a limit condition or to 0.	
12:5	PRE_L	-	Specifies the factor by which the SCTimer/PWM clock is prescaled to produce the L or unified counter clock. The counter clock is clocked at the rate of the SCTimer/PWM clock divided by PRE_L+1. **Remark:** Clear the counter (by writing a 1 to the CLRCTR bit) whenever changing the PRE value.	0
15:13	-	-	Reserved	-
16	DOWN_H	-	This read-only bit is 1 when the H counter is counting down. Hardware sets this bit when the counter is counting, a counter limit condition occurs, and BIDIR is 1. Hardware clears this bit when the counter is counting down and a limit condition occurs or when the counter reaches 0.	0
17	STOP_H	-	When this bit is 1 and HALT is 0, the H counter does not, run but I/O events related to the counter can occur. If such an event matches the mask in the Start register, this bit is cleared and counting resumes.	0

Table 389. SCTimer/PWM control register (CTRL, offset 0x004) bit description

Bit	Symbol	Value	Description	Reset value
18	HALT_H	-	When this bit is 1, the H counter does not run and no events can occur. A reset sets this bit. When the HALT_H bit is one, the STOP_H bit is cleared.	1
			It is possible to remove the halt condition while keeping the SCTimer/PWM in the stop condition (not running) with a single write to this register to simultaneously clear the HALT bit and set the STOP bit.	
			Remark: Once set, this bit can only be cleared by software to restore counter operation. This bit is set on reset.	
19	CLRCTR_H	-	When the counter is halted (not just stopped), writing a 1 to this bit will clear the H counter. This bit always reads as 0.	0
20	BIDIR_H		Direction select	0
		0	The H counter counts up to its limit condition, then is cleared to zero.	
		1	The H counter counts up to its limit, then counts down to a limit condition or to 0.	
28:21	PRE_H	-	Specifies the factor by which the SCTimer/PWM clock is prescaled to produce the H counter clock. The counter clock is clocked at the rate of the SCTimer/PWM clock divided by PRELH+1.	0
			Remark: Clear the counter (by writing a 1 to the CLRCTR bit) whenever changing the PRE value.	
31:29	-	-	Reserved	-

21.6.4 SCTimer/PWM limit event select register

The running counter can be limited by an event. When any of the events selected in this register occur, the counter is cleared to zero from its current value or changes counting direction if in bidirectional mode.

Each bit of the register is associated with a different event (bit 0 with event 0, etc.). Setting a bit causes its associated event to serve as a LIMIT event. When any limit event occurs, the counter is reset to zero in unidirectional mode or changes its direction of count in bidirectional mode. To define the actual limiting event (a match, an I/O pin toggle, etc.), see the EVn_CTRL register.

Remark: Counting up to all ones or counting down to zero is always equivalent to a limit event occurring.

Note that in addition to using this register to specify events that serve as limits, it is also possible to automatically cause a limit condition whenever a match register 0 match occurs. This eliminates the need to define an event for the sole purpose of creating a limit. The AUTOLIMITL and AUTOLIMITH bits in the configuration register enable/disable this feature (see Table 388).

If UNIFY = 1 in the CONFIG register, only the _L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers LIMIT_L and LIMIT_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

Table 390. SCTimer/PWM limit event select register (LIMIT, offset 0x008) bit description

Bit	Symbol	Description	Reset value
7:0	LIMMSK_L	If bit n is one, event n is used as a counter limit for the L or unified counter (event 0 = bit 0, event 1 = bit 1, event 7 = bit 7).	0
15:8	-	Reserved.	-
23:16	LIMMSK_H	If bit n is one, event n is used as a counter limit for the H counter (event 0 = bit 16, event 1 = bit 17, event 7 = bit 23).	0
31:24	-	Reserved.	-

21.6.5 SCTimer/PWM halt event select register

The running counter can be disabled (halted) by an event. When any of the events selected in this register occur, the counter stops running and all further events are disabled.

Each bit of the register is associated with a different event (bit 0 with event 0, etc.). Setting a bit will cause its associated event to serve as a HALT event. To define the actual events that cause the counter to halt (a match, an I/O pin toggle, etc.), see the EVn_CTRL registers.

Remark: A HALT condition can only be removed when software clears the HALT bit in the CTRL register (Table 389).

If UNIFY = 1 in the CONFIG register, only the L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers HALT_L and HALT_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

Table 391. SCTimer/PWM halt event select register (HALT, offset 0x00C) bit description

Bit	Symbol	Description	Reset value
7:0	HALTMSK_L	If bit n is one, event n sets the HALT_L bit in the CTRL register (event 0 = bit 0, event 1 = bit 1, event 7 = bit 7).	0
15:8	-	Reserved.	-
23:16	HALTMSK_H	If bit n is one, event n sets the HALT_H bit in the CTRL register (event 0 = bit 16, event 1 = bit 17, event 7 = bit 23).	0
31:24	-	Reserved.	-

21.6.6 SCTimer/PWM stop event select register

The running counter can be stopped by an event. When any of the events selected in this register occur, counting is suspended, that is the counter stops running and remains at its current value. Event generation remains enabled, and any event selected in the START register such as an I/O event or an event generated by the other counter can restart the counter.

UM11029
© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **391 of 530**

This register specifies which events stop the counter. Each bit of the register is associated with a different event (bit 0 with event 0, etc.). Setting a bit will cause its associated event to serve as a STOP event. To define the actual event that causes the counter to stop (a match, an I/O pin toggle, etc.), see the EVn_CTRL register.

Remark: Software can stop and restart the counter by writing to the CTRL register.

If UNIFY = 1 in the CONFIG register, only the _L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers STOPT_L and STOP_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

Table 392. SCTimer/PWM stop event select register (STOP, offset 0x010) bit description

Bit	Symbol	Description	Reset value
7:0	STOPMSK_L	If bit n is one, event n sets the STOP_L bit in the CTRL register (event 0 = bit 0, event 1 = bit 1, event 7 = bit 7).	0
15:8	-	Reserved.	-
23:16	STOPMSK_H	If bit n is one, event n sets the STOP_H bit in the CTRL register (event 0 = bit 16, event 1 = bit 17, event 7= bit 23).	0
31:24	-	Reserved.	-

21.6.7 SCTimer/PWM start event select register

The stopped counter can be re-started by an event. When any of the events selected in this register occur, counting is restarted from the current counter value.

Each bit of the register is associated with a different event (bit 0 with event 0, etc.). Setting a bit will cause its associated event to serve as a START event. When any START event occurs, hardware will clear the STOP bit in the Control Register CTRL. Note that a START event has no effect on the HALT bit. Only software can remove a HALT condition. To define the actual event that starts the counter (an I/O pin toggle or an event generated by the other running counter in dual-counter mode), see the EVn_CTRL register.

If UNIFY = 1 in the CONFIG register, only the _L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers START_L and START_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

Table 393. SCTimer/PWM start event select register (START, offset 0x014) bit description

Bit	Symbol	Description	Reset value
7:0	STARTMSK_L	If bit n is one, event n clears the STOP_L bit in the CTRL register (event 0 = bit 0, event 1 = bit 1, event 7 = bit 7).	0
15:8	-	Reserved.	-
23:16	STARTMSK_H	If bit n is one, event n clears the STOP_H bit in the CTRL register (event 0 = bit 16, event 1 = bit 17, event 7 = bit 23).	0
31:24	-	Reserved.	-

UM11029

User manual **Rev. 1.6 — 8 December 2017** **392 of 530**

21.6.8 SCTimer/PWM counter register

If UNIFY = 1 in the CONFIG register, the counter is a unified 32-bit register and both the _L and _H bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers COUNT_L and COUNT_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation. In this case, the L and H registers count independently under the control of the other registers.

Writing to the COUNT_L, COUNT_H, or unified register is only allowed when the corresponding counter is halted (HALT bits are set to 1 in the CTRL register). Attempting to write to the counter when it is not halted causes a bus error. Software can read the counter registers at any time.

Table 394. SCTimer/PWM counter register (COUNT, offset 0x040) bit description

Bit	Symbol	Description	Reset value
15:0	CTR_L	When UNIFY = 0, read or write the 16-bit L counter value. When UNIFY = 1, read or write the lower 16 bits of the 32-bit unified counter.	0
31:16	CTR_H	When UNIFY = 0, read or write the 16-bit H counter value. When UNIFY = 1, read or write the upper 16 bits of the 32-bit unified counter.	0

21.6.9 SCTimer/PWM state register

Each group of enabled and disabled events is assigned a number called the state variable. For example, a state variable with a value of 0 could have events 0, 2, and 3 enabled and all other events disabled. A state variable with the value of 1 could have events 1, 4, and 5 enabled and all others disabled.

Remark: The EVm_STATE registers define which event is enabled in each group.

Software can read the state associated with a counter at any time. Writing to the STATE_L, STATE_H, or unified register is only allowed when the corresponding counter is halted (HALT bits are set to 1 in the CTRL register).

The state variable is the main feature that distinguishes the SCTimer/PWM from other counter/timer/ PWM blocks. Events can be made to occur only in certain states. Events, in turn, can perform the following actions:

- set and clear outputs
- limit, stop, and start the counter
- cause interrupts and DMA requests
- modify the state variable

The value of a state variable is completely under the control of the application. If an application does not use states, the value of the state variable remains zero, which is the default value.

A state variable can be used to track and control multiple cycles of the associated counter in any desired operational sequence. The state variable is logically associated with a state machine diagram which represents the SCTimer/PWM configuration. See Section 21.6.24 and 21.6.25 for more about the relationship between states and events.

The STATELD/STADEV fields in the event control registers of all defined events set all possible values for the state variable. The change of the state variable during multiple counter cycles reflects how the associated state machine moves from one state to the next.

If UNIFY = 1 in the CONFIG register, only the _L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers STATE_L and STATE_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

Table 395. SCTimer/PWM state register (STATE, offset 0x044) bit description

Bit	Symbol	Description	Reset value
4:0	STATE_L	State variable.	0
15:5	-	Reserved.	-
20:16	STATE_H	State variable.	0
31:21	-	Reserved.	-

21.6.10 SCTimer/PWM input register

Software can read the state of the SCTimer/PWM inputs in this read-only register in slightly different forms.

1. The AIN bit displays the state of the input captured on each rising edge of the SCTimer/PWM clock This corresponds to a nearly direct read-out of the input but can cause spurious fluctuations in case of an asynchronous input signal.

2. The SIN bit displays the form of the input as it is used for event detection. This may include additional stages of synchronization, depending on what is specified for that input in the INSYNC field in the CONFIG register:

   – If the INSYNC bit is set for the input, the input is triple-synchronized to the SCTimer/PWM clock resulting in a stable signal that is delayed by three SCTimer/PWM clock cycles.

   – If the INSYNC bit is not set, the SIN bit value is identical to the AIN bit value.

Table 396. SCTimer/PWM input register (INPUT, offset 0x048) bit description

Bit	Symbol	Description	Reset value
0	AIN0	Input 0 state. Input 0 state on the last SCTimer/PWM clock edge.	-
1	AIN1	Input 1 state. Input 1 state on the last SCTimer/PWM clock edge.	-
2	AIN2	Input 2 state. Input 2 state on the last SCTimer/PWM clock edge.	-
3	AIN3	Input 3 state. Input 3 state on the last SCTimer/PWM clock edge.	-
4	AIN4	Input 4 state. Input 4 state on the last SCTimer/PWM clock edge.	-
15:5	-	Reserved.	-

Table 396. SCTimer/PWM input register (INPUT, offset 0x048) bit description

Bit	Symbol	Description	Reset value
16	SIN0	Input 0 state. Input 0 state following the synchronization specified by INSYNC0.	-
17	SIN1	Input 1 state. Input 1 state following the synchronization specified by INSYNC1.	-
18	SIN2	Input 2 state. Input 2 state following the synchronization specified by INSYNC2.	-
19	SIN3	Input 3 state. Input 3 state following the synchronization specified by INSYNC3.	-
20	SIN4	Input 4 state. Input 4 state following the synchronization specified by INSYNC0.	-
31:21	-	Reserved	-

21.6.11 SCTimer/PWM match/capture mode register

If UNIFY = 1 in the CONFIG register, only the _L bits of this register are used. In this case, REGMODE_H is not used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers REGMODE_L and REGMODE_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation. The _L bits/registers control the L match/capture registers, and the _H bits/registers control the H match/capture registers.

The SCTimer/PWM contains multiple Match/Capture registers. The Register Mode register selects whether each register acts as a Match register (see Section 21.6.20) or as a Capture register (see Section 21.6.21). Each Match/Capture register has an accompanying register which functions as a Reload register when the primary register is used as a Match register (Section 21.6.22) or as a Capture-Control register (event select) when the register is used as a capture register (Section 21.6.23). REGMODE_H is used only when the UNIFY bit is 0.

Table 397. SCTimer/PWM match/capture mode register (REGMODE, offset 0x04C) bit description

Bit	Symbol	Description	Reset value
7:0	REGMOD_L	Each bit controls one match/capture register (register 0 = bit 0, register 1 = bit 1,..., register = bit 7). 0 = register operates as match register. 1 = register operates as capture register.	0
15:8	-	Reserved.	-
23:16	REGMOD_H	Each bit controls one match/capture register (register 0 = bit 16, register 1 = bit 17,..., register 7 = bit 23). 0 = register operates as match registers. 1 = register operates as capture registers.	0
31:24	-	Reserved.	-

21.6.12 SCTimer/PWM output register

Each SCTimer/PWM output has a corresponding bit in this register to allow software to control the output state directly or read its current state.

While the counter is running, outputs are set, cleared, or toggled only by events. However, using this register, software can write to any of the output registers when both counters are halted to control the outputs directly. Writing to the OUT register is only allowed when all counters (L-counter, H-counter, or unified counter) are halted (HALT bits are set to 1 in the CTRL register).

Software can read this register at any time to sense the state of the outputs.

Table 398. SCTimer/PWM output register (OUTPUT, offset 0x050) bit description

Bit	Symbol	Description	Reset value
6:0	OUT	Writing a 1 to bit n forces the corresponding output HIGH. Writing a 0 forces the corresponding output LOW (output 0 = bit 0, output 1 = bit 1,..., output 6 = bit 6).	0
31:6	-	Reserved	-

21.6.13 SCTimer/PWM bidirectional output control register

For bidirectional mode, this register specifies (for each output) the impact of the counting direction on the meaning of set and clear operations on the output (see Section 21.6.26 and Section 21.6.27). The purpose of this register is to facilitate the creation of center-aligned output waveforms without the need to define additional events.

Table 399. SCTimer/PWM bidirectional output control register (OUTPUTDIRCTRL, offset 0x054) bit description

Bit	Symbol	Value	Description	Reset value
1:0	SETCLR0		Set/clear operation on output 0.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	
3:2	SETCLR1		Set/clear operation on output 1.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	
5:4	SETCLR2		Set/clear operation on output 2.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	
7:6	SETCLR3		Set/clear operation on output 3.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	

Table 399. SCTimer/PWM bidirectional output control register (OUTPUTDIRCTRL, offset 0x054) bit description

Bit	Symbol	Value	Description	Reset value
9:8	SETCLR4		Set/clear operation on output 4.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	
11:10	SETCLR5		Set/clear operation on output 5.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	
13:12	SETCLR6		Set/clear operation on output 6.	0
		0x0	Set and clear do not depend on the direction of any counter.	
		0x1	Set and clear are reversed when counter L or the unified counter is counting down.	
		0x2	Set and clear are reversed when counter H is counting down. Do not use if UNIFY = 1.	
31:14	-	-	Reserved	-

21.6.14 SCTimer/PWM conflict resolution register

The output conflict resolution register specifies what action should be taken if multiple events (or even the same event) dictate that a given output should be both set and cleared at the same time.

To enable an event to toggle an output each time the event occurs, set the bits for that event in both the OUTn_SET and OUTn_CLR registers and set the On_RES value to 0x3 in this register.

Table 400. SCTimer/PWM conflict resolution register (RES, offset 0x058) bit description

Bit	Symbol	Value	Description	Reset value
1:0	O0RES		Effect of simultaneous set and clear on output 0.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR0 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output (or set based on the SETCLR0 field).	
		0x3	Toggle output.	
3:2	O1RES		Effect of simultaneous set and clear on output 1.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR1 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output (or set based on the SETCLR1 field).	
		0x3	Toggle output.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **397 of 530**

Table 400. SCTimer/PWM conflict resolution register (RES, offset 0x058) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
5:4	O2RES		Effect of simultaneous set and clear on output 2.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR2 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output n (or set based on the SETCLR2 field).	
		0x3	Toggle output.	
7:6	O3RES		Effect of simultaneous set and clear on output 3.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR3 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output (or set based on the SETCLR3 field).	
		0x3	Toggle output.	
9:8	O4RES		Effect of simultaneous set and clear on output 4.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR4 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output (or set based on the SETCLR4 field).	
		0x3	Toggle output.	
11:10	O5RES		Effect of simultaneous set and clear on output 5.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR5 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output (or set based on the SETCLR5 field).	
		0x3	Toggle output.	
13:12	O6RES		Effect of simultaneous set and clear on output 6.	0
		0x0	No change.	
		0x1	Set output (or clear based on the SETCLR6 field in the OUTPUTDIRCTRL register).	
		0x2	Clear output (or set based on the SETCLR6 field).	
		0x3	Toggle output.	
31:14	-	-	Reserved	-

21.6.15 SCTimer/PWM DMA request 0 and 1 registers

The SCTimer/PWM includes two DMA request outputs. These registers enable the DMA requests to be triggered when a particular event occurs or when counter Match registers are loaded from its Reload registers. The DMA request registers are word-write only. Attempting to write a half-word value to these registers result in a bus error.

Event-triggered DMA requests are particularly useful for launching DMA activity to or from other peripherals under the control of the SCTimer/PWM.

Table 401. SCTimer/PWM DMA 0 request register (DMAREQ0, offset 0x05C) bit description

Bit	Symbol	Description	Reset value
5:0	DEV_0	If bit n is one, event n triggers DMA request 0 (event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7).	0
29:6	-	Reserved	-
30	DRL0	A 1 in this bit triggers DMA request 0 when it loads the Match_L/Unified registers from the Reload_L/Unified registers.	0
31	DRQ0	This read-only bit indicates the state of DMA Request 0. Note that if the related DMA channel is enabled and properly set up, it is unlikely that software will see this flag, it will be cleared rapidly by the DMA service. The flag remaining set could point to an issue with DMA setup.	0

Table 402. SCTimer/PWM DMA 1 request register (DMAREQ1, offset 0x060) bit description

Bit	Symbol	Description	Reset value
5:0	DEV_1	If bit n is one, event n triggers DMA request 1 (event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7).	0
29:6	-	Reserved	-
30	DRL1	A 1 in this bit triggers DMA request 1 when it loads the Match L/Unified registers from the Reload L/Unified registers.	0
31	DRQ1	This read-only bit indicates the state of DMA Request 1. Note that if the related DMA channel is enabled and properly set up, it is unlikely that software will see this flag, it will be cleared rapidly by the DMA service. The flag remaining set could point to an issue with DMA setup.	0

21.6.16 SCTimer/PWM event interrupt enable register

This register enables flags to request an interrupt if the FLAGn bit in the SCTimer/PWM event flag register (Section 21.6.17) is also set.

Table 403. SCTimer/PWM event interrupt enable register (EVEN, offset 0x0F0) bit description

Bit	Symbol	Description	Reset value
7:0	IEN	The SCTimer/PWM requests an interrupt when bit n of this register and the event flag register are both one (event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7).	0
31:8	-	Reserved	-

21.6.17 SCTimer/PWM event flag register

This register records events. Writing ones to this register clears the corresponding flags and negates the SCTimer/PWM interrupt request if all enabled flag register bits are zero.

Table 404. SCTimer/PWM event flag register (EVFLAG, offset 0x0F4) bit description

Bit	Symbol	Description	Reset value
7:0	FLAG	Bit n is one if event n has occurred since reset or a 1 was last written to this bit (event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7).	0
31:8	-	Reserved	-

21.6.18 SCTimer/PWM conflict interrupt enable register

This register enables the no-change conflict events specified in the SCTimer/PWM conflict resolution register to generate an interrupt request.

Table 405. SCTimer/PWM conflict interrupt enable register (CONEN, offset 0x0F8) bit description

Bit	Symbol	Description	Reset value
5:0	NCEN	The SCTimer/PWM requests an interrupt when bit n of this register and the SCTimer/PWM conflict flag register are both one (output 0 = bit 0, output 1 = bit 1,..., output 5 = bit 5).	0
31:6	-	Reserved	-

21.6.19 SCTimer/PWM conflict flag register

This register records a no-change conflict occurrence and provides details of a bus error. Writing ones to the NCFLAG bits clears the corresponding read bits and negates the SCTimer/PWM interrupt request if all enabled Flag bits are zero.

Table 406. SCTimer/PWM conflict flag register (CONFLAG, offset 0x0FC) bit description

Bit	Symbol	Description	Reset value
5:0	NCFLAG	Bit n is one if a no-change conflict event occurred on output n since reset or a 1 was last written to this bit (output 0 = bit 0, output 1 = bit 1,..., output 5 = bit 5).	0
29:6	-	Reserved.	-
30	BUSERRL	The most recent bus error from this SCTimer/PWM involved writing CTR L/Unified, STATE L/Unified, MATCH L/Unified, or the Output register when the L/U counter was not halted. A word write to certain L and H registers can be half successful and half unsuccessful.	0
31	BUSERRH	The most recent bus error from this SCTimer/PWM involved writing CTR H, STATE H, MATCH H, or the Output register when the H counter was not halted.	0

21.6.20 SCTimer/PWM match registers 0 to 7 (REGMODEn bit = 0)

Match registers are compared to the counters to help create events. When the UNIFY bit is 0, the L and H registers are independently compared to the L and H counters. When UNIFY is 1, the combined L and H registers hold a 32-bit value that is compared to the unified counter. A Match can only occur in a clock in which the counter is running (STOP and HALT are both 0).

Match registers can be read at any time. Writing to the MATCH_L, MATCH_H, or unified register is only allowed when the corresponding counter is halted (HALT bits are set to 1 in the CTRL register). Match events occur in the SCTimer/PWM clock in which the counter is (or would be) incremented to the next value. When a Match event limits its counter as described in Section 21.6.4, the value in the Match register is the last value of the counter before it is cleared to zero (or decremented if BIDIR is 1).

There is no "write-through" from Reload registers to Match registers. Before starting a counter, software can write one value to the Match register used in the first cycle of the counter and a different value to the corresponding Match Reload register used in the second cycle.

Table 407. SCTimer/PWM match registers 0 to 7 (MATCH[0:7], offset 0x100 (MATCH0) to 0x11C (MATCH7)) bit description (REGMODEn bit = 0)

Bit	Symbol	Description	Reset value
15:0	MATCHn_L	When UNIFY = 0, read or write the 16-bit value to be compared to the L counter. When UNIFY = 1, read or write the lower 16 bits of the 32-bit value to be compared to the unified counter.	0
31:16	MATCHn_H	When UNIFY = 0, read or write the 16-bit value to be compared to the H counter. When UNIFY = 1, read or write the upper 16 bits of the 32-bit value to be compared to the unified counter.	0

21.6.21 SCTimer/PWM capture registers 0 to 7 (REGMODEn bit = 1)

These registers allow software to record the counter values upon occurrence of the events selected by the corresponding Capture Control registers occurred.

Table 408. SCTimer/PWM capture registers 0 to 7 (CAP[0:7], offset 0x100 (CAP0) to 0x11C (CAP7)) bit description (REGMODEn bit = 1)

Bit	Symbol	Description	Reset value
15:0	CAPn_L	When UNIFY = 0, read the 16-bit counter value at which this register was last captured. When UNIFY = 1, read the lower 16 bits of the 32-bit value at which this register was last captured.	0
31:16	CAPn_H	When UNIFY = 0, read the 16-bit counter value at which this register was last captured. When UNIFY = 1, read the upper 16 bits of the 32-bit value at which this register was last captured.	0

21.6.22 SCTimer/PWM match reload registers 0 to 7 (REGMODEn bit = 0)

A Match register (L, H, or unified 32-bit) is loaded from its corresponding Reload register at the start of each new counter cycle, that is

- when BIDIR = 0 and the counter is cleared to zero upon reaching it limit condition.
- when BIDIR = 1 and the counter counts down to 0.

In either case, reloading does not occur if the corresponding NORELOAD bit is set in the CFG register.

Table 409. SCTimer/PWM match reload registers 0 to 7 (MATCHREL[0:7], offset 0x200 (MATCHREL0) to 0x21C (MATCHREL7)) bit description (REGMODEn bit = 0)

Bit	Symbol	Description	Reset value
15:0	RELOADn_L	When UNIFY = 0, specifies the 16-bit value to be loaded into the MATCHn_L register. When UNIFY = 1, specifies the lower 16 bits of the 32-bit value to be loaded into the MATCHn register.	0
31:16	RELOADn_H	When UNIFY = 0, specifies the 16-bit to be loaded into the MATCHn_H register. When UNIFY = 1, specifies the upper 16 bits of the 32-bit value to be loaded into the MATCHn register.	0

21.6.23 SCTimer/PWM capture control registers 0 to 7 (REGMODEn bit = 1)

If UNIFY = 1 in the CONFIG register, only the _L bits are used.

If UNIFY = 0 in the CONFIG register, this register can be written to as two registers CAPCTRLn_L and CAPCTRLn_H. Both the L and H registers can be read or written individually or in a single 32-bit read or write operation.

The capture registers can be loaded with the current counter value when any of the specified events occur.

Each Capture Control register (L, H, or unified 32-bit) controls which events cause loading of the corresponding Capture register from the counter.

Table 410. SCTimer/PWM capture control registers 0 to 7 (CAPCTRL[0:7], offset 0x200 (CAPCTRL0) to 0x21C (CAPCTRL7)) bit description (REGMODEn bit = 1)

Bit	Symbol	Description	Reset value
7:0	CAPCONn_L	If bit m is one, event m causes the CAPn_L (UNIFY = 0) or the CAPn (UNIFY = 1) register to be loaded (event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7).	0
15:8	-	Reserved.	-
23:16	CAPCONn_H	If bit m is one, event m causes the CAPn_H (UNIFY = 0) register to be loaded (event 0 = bit 16, event 1 = bit 17,..., event 7 = bit 23).	0
31:24	-	Reserved.	-

21.6.24 SCTimer/PWM event enable registers 0 to 7

Each event can be enabled in some contexts (or states) and disabled in others. Each event defined in the EV_CTRL register has one associated event enable register that can enable or disable the event for each available state.

An event n is completely disabled when its EVn_STATE register contains all zeros, since it is masked regardless of the current state.

In simple applications that do not use states, writing 0x01 (or any other value with a 1 in bit 0) will enable the event. Since the state doesn't change (that is, the state variable always remains at its reset value of 0), setting bit 0 permanently enables this event. Conversely, clearing bit 0 will disable the event.

Table 411. SCTimer/PWM event state mask registers 0 to 7 (EV[0:7]_STATE, offset 0x300 (EV0_STATE) to 0x338 (EV7_STATE)) bit description

Bit	Symbol	Description	Reset value
7:0	STATEMSKn	If bit m is one, event n (n= 0 to 7) happens in state m of the counter selected by the HEVENT bit (m = state number; state 0 = bit 0, state 1= bit 1,..., state 7 = bit 7).	0
31:8	-	Reserved.	-

21.6.25 SCTimer/PWM event control registers 0 to 7

This register defines the conditions for an event to occur based on the counter values or input and output states.Once the event is configured, it can be selected to trigger multiple actions (for example stop the counter and toggle an output) unless the event is blocked in the current state of the SCT or the counter is halted. To block a particular event from occurring, use the EV_STATE register. To block all events for a given counter, set the HALT bit in the CTRL register or select an event to halt the counter.

An event can be programmed to occur based on a selected input or output edge or level and/or based on its counter value matching a selected match register. In bidirectional mode, events can also be enabled based on the direction of count.

When the UNIFY bit is 0, each event is associated with a particular counter by the HEVENT bit in its event control register. An event is permanently disabled when its event state mask register contains all 0s.

Each event can modify its counter STATE value. If more than one event associated with the same counter occurs in a given clock cycle, only the state change specified for the highest-numbered event among them takes place. Other actions dictated by any simultaneously occurring events all take place.

Table 412. SCTimer/PWM event control register 0 to 7 (EV[0:7]_CTRL, offset 0x304 (EV0_CTRL) to 0x33C (EV7_CTRL)) bit description

Bit	Symbol	Value	Description	Reset value
3:0	MATCHSEL	-	Selects the Match register associated with this event (if any). A match can occur only when the counter selected by the HEVENT bit is running.	0
4	HEVENT		Select L/H counter. Do not set this bit if UNIFY = 1.	0
		0	Selects the L state and the L match register selected by MATCHSEL.	
		1	Selects the H state and the H match register selected by MATCHSEL.	
5	OUTSEL		Input/output select	0
		0	Selects the inputs elected by IOSEL.	
		1	Selects the outputs selected by IOSEL.	
9:6	IOSEL	-	Selects the input or output signal number (0 to 3 for inputs or 0 to 5 for outputs) associated with this event (if any). Do not select an input in this register, if CLKMODE is 1x. In this case the clock input is an implicit ingredient of every event.	0

User manual **Rev. 1.6 — 8 December 2017** **403 of 530**

Table 412. SCTimer/PWM event control register 0 to 7 (EV[0:7]_CTRL, offset 0x304 (EV0_CTRL) to 0x33C (EV7_CTRL)) bit description

Bit	Symbol	Value	Description	Reset value
11:10	IOCOND		Selects the I/O condition for event n. (The detection of edges on outputs lag the conditions that switch the outputs by one SCTimer/PWM clock). In order to guarantee proper edge/state detection, an input must have a minimum pulse width of at least one SCTimer/PWM clock period .	0
		0x0	LOW	
		0x1	Rise	
		0x2	Fall	
		0x3	HIGH	
13:12	COMBMODE		Selects how the specified match and I/O condition are used and combined.	0
		0x0	OR. The event occurs when either the specified match or I/O condition occurs.	
		0x1	MATCH. Uses the specified match only.	
		0x2	IO. Uses the specified I/O condition only.	
		0x3	AND. The event occurs when the specified match and I/O condition occur simultaneously.	
14	STATELD		This bit controls how the STATEV value modifies the state selected by HEVENT when this event is the highest-numbered event occurring for that state.	0
		0	STATEV value is added into STATE (the carry-out is ignored).	
		1	STATEV value is loaded into STATE.	
19:15	STATEV	-	This value is loaded into or added to the state selected by HEVENT, depending on STATELD, when this event is the highest-numbered event occurring for that state. If STATELD and STATEV are both zero, there is no change to the STATE value.	0
20	MATCHMEM	-	If this bit is one and the COMBMODE field specifies a match component to the triggering of this event, then a match is considered to be active whenever the counter value is GREATER THAN OR EQUAL TO the value specified in the match register when counting up, LESS THEN OR EQUAL TO the match value when counting down. If this bit is zero, a match is only be active during the cycle when the counter is equal to the match value.	0
22:21	DIRECTION		Direction qualifier for event generation. This field only applies when the counters are operating in BIDIR mode. If BIDIR = 0, the SCTimer/PWM ignores this field. Value 0x3 is reserved.	0
		0x0	Direction independent. This event is triggered regardless of the count direction.	
		0x1	Counting up. This event is triggered only during up-counting when BIDIR = 1.	
		0x2	Counting down. This event is triggered only during down-counting when BIDIR = 1.	
31:23	-	-	Reserved	-

21.6.26 SCTimer/PWM output set registers 0 to 6

Each SCT output can be set on the occurrence of one or more specified events.

There is one output set register for each SCTimer/PWM output which selects which events can set that output. Each bit of an output set register is associated with a different event (bit 0 with event 0, etc.).

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **404 of 530**

Note that it is possible to reverse the action specified by "SET" and "CLR" when counting down in bi-directional mode depending on the setting of the SETCLRn field in the OUTPUTDIRCTRL register. To define the creation of the actual event(s) that sets an output (a match, an I/O pin toggle, etc.), see the EVn_CTRL register.

Remark: If the SCTimer/PWM is operating as two 16-bit counters, events can only modify the state of the outputs when neither counter is halted. This is true regardless of what triggered the event.

Table 413. SCTimer/PWM output set register (OUT[0:6]_SET, offset 0x500 (OUT0_SET) to 0x530 (OUT6_SET) bit description

Bit	Symbol	Description	Reset value
7:0	SET	A 1 in bit m selects event m to set output n (or clear it if SETCLRn = 0x1 or 0x2) event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7. When the counter is used in bidirectional mode, it is possible to reverse the action specified by the output set and clear registers when counting down, See the OUTPUTCTRL register.	0
31:8	-	Reserved	-

21.6.27 SCTimer/PWM output clear registers 0 to 6

Each SCT output can be cleared on the occurrence of one or more specified events.

There is one register for each SCTimer/PWM output which selects which events can clear that output. Each bit of an output clear register is associated with a different event (bit 0 with event 0, etc.).

Note that it is possible to reverse the action specified by "SET" and "CLR" when counting down in bi-directional mode depending on the setting of the SETCLRn field in the OUTPUTDIRCTRL register. To define the creation of the actual event(s) that sets an output (a match, an I/O pin toggle, etc.), see the EVn_CTRL register.

Remark: If the SCTimer/PWM is operating as two 16-bit counters, events can only modify the state of the outputs when neither counter is halted. This is true regardless of what triggered the event.

Table 414. SCTimer/PWM output clear register (OUT[0:6]_CLR, offset 0x504 (OUT0_CLR) to 0x534 (OUT6_CLR)) bit description

Bit	Symbol	Description	Reset value
7:0	CLR	A 1 in bit m selects event m to clear output n (or set it if SETCLRn = 0x1 or 0x2) event 0 = bit 0, event 1 = bit 1,..., event 7 = bit 7. When the counter is used in bidirectional mode, it is possible to reverse the action specified by the output set and clear registers when counting down, See the OUTPUTCTRL register.	0
31:8	-	Reserved	-

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **405 of 530**

21.7 Functional description

21.7.1 Match logic
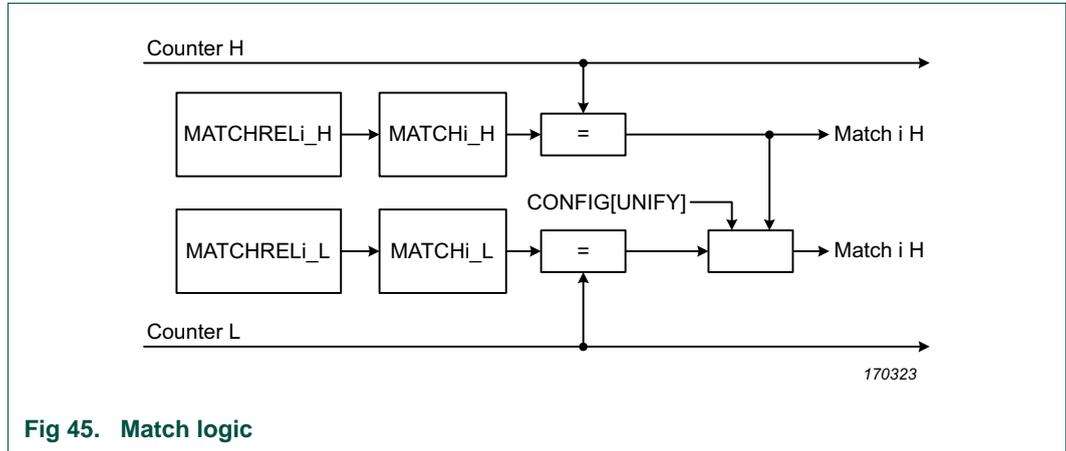
Fig 45.   Match logic

21.7.2 Capture logic
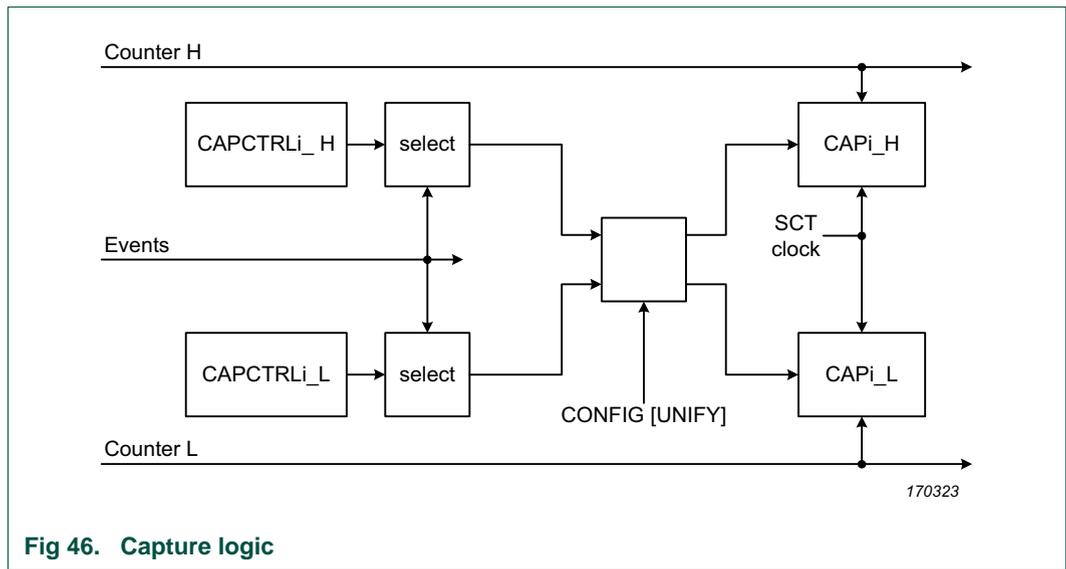
Fig 46.   Capture logic

21.7.3 Event selection

State variables allow control of the SCTimer/PWM across more than one cycle of the counter. Counter matches, input/output edges, and state values are combined into a set of general-purpose events that can switch outputs, request interrupts, and change state values.
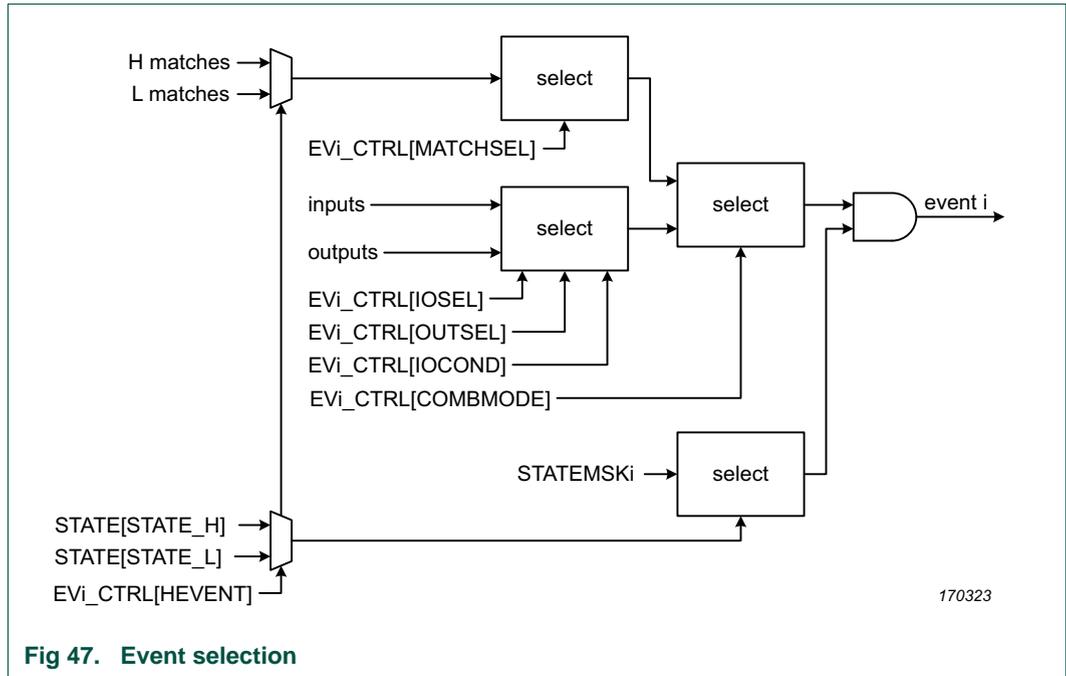
Fig 47. Event selection

21.7.4 Output generation

Figure 48 shows one output slice of the SCTimer/PWM.
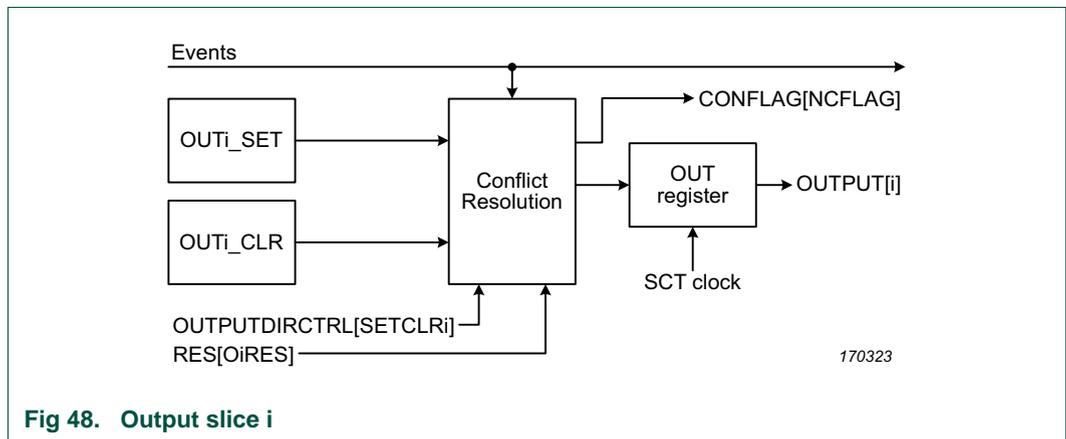
Fig 48. Output slice i

21.7.5 State logic

The SCTimer/PWM can be configured as a timer/counter with multiple programmable states. The states are user-defined through the events that can be captured in each particular state. In a multi-state SCTimer/PWM. the SCTimer/PWM can change from one state to another state when a user-defined event triggers a state change. The state change is triggered through each event's EV_CTRL register in one of the following ways:

- The event can increment the current state number by a new value.
- The event can write a new state value.

If an event increments the state number beyond the number of available states, the SCTimer/PWM enters a locked state in which all further events are ignored while the counter is still running. Software must intervene to change out of this state.

Software can capture the counter value (and potentially create an interrupt and write to all outputs) when the event moving the SCTimer/PWM into a locked state occurs. Later, while the SCTimer/PWM is in the locked state, software can read the counter again to record the time passed since the locking event and can also read the state variable to obtain the current state number

If the SCTimer/PWM registers an event that forces an abort, putting the SCTimer/PWM in a locked state can be a safe way to record the time that has passed since the abort event while no new events are allowed to occur. Since multiple states (any state number between the maximum implemented state and 31) are locked states, multiple abort or error events can be defined each incrementing the state number by a different value.

21.7.6 Interrupt generation

The SCTimer/PWM generates one interrupt to the NVIC.
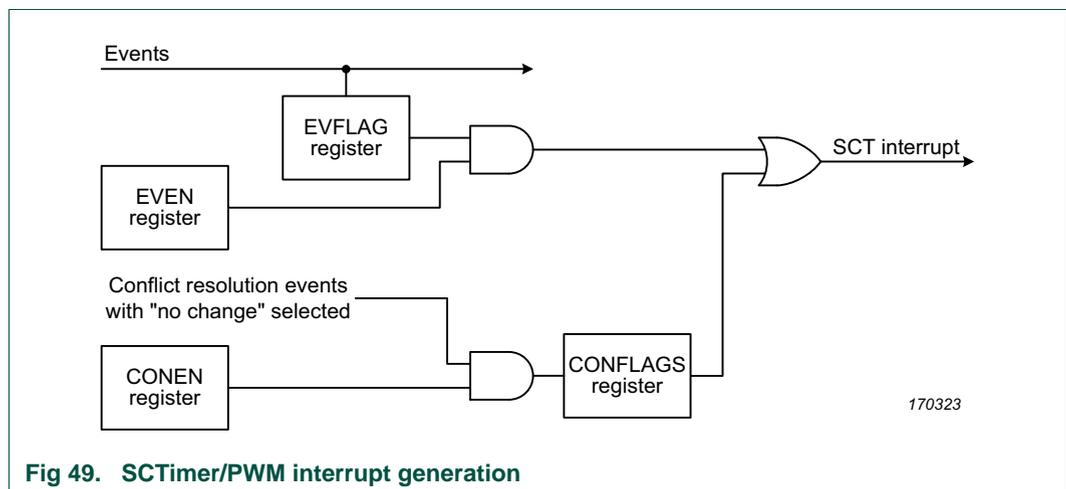
Fig 49. SCTimer/PWM interrupt generation

21.7.7 Clearing the prescaler

When enabled by a non-zero PRE field in the Control register, the prescaler acts as a clock divider for the counter, like a fractional part of the counter value. The prescaler is cleared whenever the counter is cleared or loaded for any of the following reasons:

- Hardware reset
- Software writing to the counter register
- Software writing a 1 to the CLRCTR bit in the control register
- an event selected by a 1 in the counter limit register when BIDIR = 0

When BIDIR is 0, a limit event caused by an I/O signal can clear a non-zero prescaler. However, a limit event caused by a Match only clears a non-zero prescaler in one special case as described Section 21.7.8.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **408 of 530**

A limit event when BIDIR is 1 does not clear the prescaler. Rather it clears the DOWN bit in the Control register, and decrements the counter on the same clock if the counter is enabled in that clock.

21.7.8 Match vs. I/O events

Counter operation is complicated by the prescaler and by clock mode 01 in which the SCTimer/PWM clock is the bus clock. However, the prescaler and counter are enabled to count only when a selected edge is detected on a clock input.

- The prescaler is enabled when the clock mode is not 01, or when the input edge selected by the CLKSEL field is detected.
- The counter is enabled when the prescaler is enabled, and (PRELIM=0 or the prescaler is equal to the value in PRELIM).

An I/O component of an event can occur in any SCTimer/PWM clock when its counter HALT bit is 0. In general, a Match component of an event can only occur in an SCTimer/PWM clock when its counter HALT and STOP bits are both 0 and the counter is enabled.

Table 415 shows when the various kinds of events can occur.

Table 415. Event conditions

COMBMODE	IOMODE	Event can occur on clock:
IO	Any	Event can occur whenever HALT = 0 (type A).
MATCH	Any	Event can occur when HALT = 0 and STOP = 0 and the counter is enabled (type C).
OR	Any	From the IO component: Event can occur whenever HALT = 0 (A). From the match component: Event can occur when HALT = 0 and STOP = 0 and the counter is enabled (C).
AND	LOW or HIGH	Event can occur when HALT = 0 and STOP = 0 and the counter is enabled (C).
AND	RISE or FALL	Event can occur whenever HALT = 0 (A).

21.7.9 SCTimer/PWM operation

In its simplest, single-state configuration, the SCTimer/PWM operates as an event controlled unidirectional or bidirectional counter. Events can be configured to be counter match events, an input or output level, transitions on an input or output pin, or a combination of match and input/output behavior. In response to an event, the SCTimer/PWM output or outputs can transition, or the SCTimer/PWM can perform other actions such as creating an interrupt or starting, stopping, or resetting the counter. Multiple simultaneous actions are allowed for each event. Furthermore, any number of events can trigger one specific action of the SCTimer/PWM.

An action or multiple actions of the SCTimer/PWM uniquely define an event. A state is defined by which events are enabled to trigger an SCTimer/PWM action or actions in any stage of the counter. Events not selected for this state are ignored.

In a multi-state configuration, states change in response to events. A state change is an additional action that the SCTimer/PWM can perform when the event occurs. When an event is configured to change the state, the new state defines a new set of events resulting in different actions of the SCTimer/PWM. Through multiple cycles of the counter, events can change the state multiple times and thus create a large variety of event controlled transitions on the SCTimer/PWM outputs and/or interrupts.

Once configured, the SCTimer/PWM can run continuously without software intervention and can generate multiple output patterns entirely under the control of events.

- To configure the SCTimer/PWM, see Section 21.7.10.
- To start, run, and stop the SCTimer/PWM, see Section 21.7.11.
- To configure the SCTimer/PWM as simple event controlled counter/timer, see Section 21.7.12.

21.7.10 Configure the SCTimer/PWM

To set up the SCTimer/PWM for multiple events and states, perform the following configuration steps:

21.7.10.1 Configure the counter

1. Configure the L and H counters in the CONFIG register by selecting two independent 16-bit counters (L counter and H counter) or one combined 32-bit counter in the UNIFY field.
2. Select the SCTimer/PWM clock source in the CONFIG register (fields CLKMODE and CLKSEL) from any of the inputs or an internal clock.

21.7.10.2 Configure the match and capture registers

1. Select how many match and capture registers the application uses (total of up to 5):
   – In the REGMODE register, select for each of the 5 match/capture register pairs whether the register is used as a match register or capture register.
2. Define match conditions for each match register selected:
   – Each match register MATCH sets one match value, if a 32-bit counter is used, or two match values, if the L and H 16-bit counters are used.

– Each match reload register MATCHRELOAD sets a reload value that is loaded into the match register when the counter reaches a limit condition or the value 0.

21.7.10.3 Configure events and event responses

1. Define when each event can occur in the following way in the EVn_CTRL registers (up to 6, one register per event):

   – Select whether the event occurs on an input or output changing, on an input or output level, a match condition of the counter, or a combination of match and input/output conditions in field COMBMODE.

   – For a match condition:

   Select the match register that contains the match condition for the event to occur. Enter the number of the selected match register in field MATCHSEL.

   If using L and H counters, define whether the event occurs on matching the L or the H counter in field HEVENT.

   – For an SCTimer/PWM input or output level or transition:

   Select the input number or the output number that is associated with this event in fields IOSEL and OUTSEL.

   Define how the selected input or output triggers the event (edge or level sensitive) in field IOCOND.

2. Define what the effect of each event is on the SCTimer/PWM outputs in the OUTn_SET or OUTn_CLR registers (up to 4 outputs, one register per output):

   – For each SCTimer/PWM output, select which events set or clear this output. More than one event can change the output, and each event can change multiple outputs.

3. Define how each event affects the counter:

   – Set the corresponding event bit in the LIMIT register for the event to set an upper limit for the counter.

   When a limit event occurs in unidirectional mode, the counter is cleared to zero and begins counting up on the next clock edge.

   When a limit event occurs in bidirectional mode, the counter begins to count down from the current value on the next clock edge.

   – Set the corresponding event bit in the HALT register for the event to halt the counter. If the counter is halted, it stops counting and no new events can occur. The counter operation can only be restored by clearing the HALT_L and/or the HALT_H bits in the CTRL register.

   – Set the corresponding event bit in the STOP register for the event to stop the counter. If the counter is stopped, it stops counting. However, an event that is configured as a transition on an input/output can restart the counter.

   – Set the corresponding event bit in the START register for the event to restart the counting. Only events that are defined by an input changing can be used to restart the counter.

4. Define which events contribute to the SCTimer/PWM interrupt:

   – Set the corresponding event bit in the EVEN and the EVFLAG registers to enable the event to contribute to the SCTimer/PWM interrupt.

21.7.10.4 Configure multiple states

1. In the EVn_STATE register for each event (up to 6 events, one register per event), select the state or states (up to 2) in which this event is allowed to occur. Each state can be selected for more than one event.

2. Determine how the event affects the system state:

   In the EVn_CTRL registers (up to 6 events, one register per event), set the new state value in the STATEV field for this event. If the event is the highest numbered in the current state, this value is either added to the existing state value or replaces the existing state value, depending on the field STATELD.

   Remark: If there are higher numbered events in the current state, this event cannot change the state.

   If the STATEV and STATELD values are set to zero, the state does not change.

21.7.10.5 Miscellaneous options

- There are a certain (selectable) number of capture registers. Each capture register can be programmed to capture the counter contents when one or more events occur.

- If the counter is in bidirectional mode, the effect of set and clear of an output can be made to depend on whether the counter is counting up or down by writing to the OUTPUTDIRCTRL register.

21.7.11 Run the SCTimer/PWM

1. Configure the SCTimer/PWM (see Section 21.7.10 "Configure the SCTimer/PWM").

2. Write to the STATE register to define the initial state. By default the initial state is state 0.

3. To start the SCTimer/PWM, write to the CTRL register:

   – Clear the counters.

   – Clear or set the STOP_L and/or STOP_H bits.

     Remark: The counter starts counting once the STOP bit is cleared as well. If the STOP bit is set, the SCTimer/PWM waits instead for an event to occur that is configured to start the counter.

   – For each counter, select unidirectional or bidirectional counting mode (field BIDIR_L and/or BIDIR_H).

   – Select the prescale factor for the counter clock (CTRL register).

   – Clear the HALT_L and/or HALT_H bit. By default, the counters are halted and no events can occur.

4. To stop the counters by software at any time, stop or halt the counter (write to STOP_L and/or STOP_H bits or HALT_L and/or HALT_H bits in the CTRL register).

   – When the counters are stopped, both an event configured to clear the STOP bit or software writing a zero to the STOP bit can start the counter again.

   – When the counter are halted, only a software write to clear the HALT bit can start the counter again. No events can occur.

   – When the counters are halted, software can set any SCTimer/PWM output HIGH or LOW directly by writing to the OUT register.

The current state can be read at any time by reading the STATE register.

To change the current state by software (that is independently of any event occurring), set the HALT bit and write to the STATE register to change the state value. Writing to the STATE register is only allowed when the counter is halted (the HALT_L and/or HALT_H bits are set) and no events can occur.

21.7.12 Configure the SCTimer/PWM without using states

The SCTimer/PWM can be used as standard counter/timer with external capture inputs and match outputs without using the state logic. To operate the SCTimer/PWM without states, configure the SCTimer/PWM as follows:

- Write zero to the STATE register (zero is the default).
- Write zero to the STATELD and STATEV fields in the EVCTRL registers for each event.
- Write 0x1 to the EVn_STATE register of each event. Writing 0x1 enables the event.

  In effect, the event is allowed to occur in a single state which never changes while the counter is running.

21.7.13 SCTimer/PWM PWM Example

Figure 50 shows a simple application of the SCTimer/PWM using two sets of match events (EV0/1 and EV3/4) to set/clear SCTimer/PWM output 0. The timer is automatically reset whenever it reaches the MAT0 match value.

In the initial state 0, match event EV0 sets output 0 to HIGH and match event EV1 clears output 0. The SCTimer/PWM input 0 is monitored: If input0 is found LOW by the next time the timer is reset(EV2), the state is changed to state 1, and EV3/4 are enabled, which create the same output but triggered by different match values. If input 0 is found HIGH by the next time the timer is reset, the associated event (EV5) causes the state to change back to state 0where the events EV0 and EV1 are enabled.

The example uses the following SCTimer/PWM configuration:

- 1 input
- 1 output
- 5 match registers
- 6 events and match 0 used with autolimit function
- 2 states
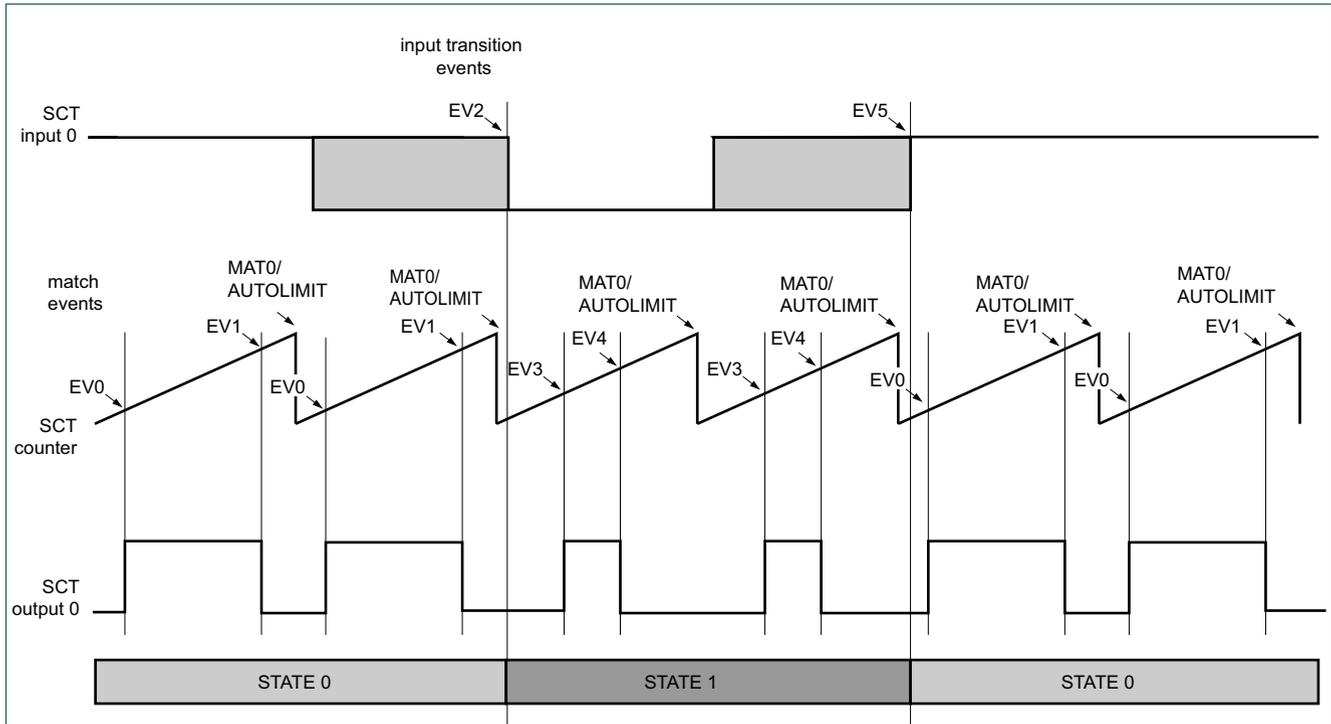
Fig 50. SCTimer/PWM configuration example

This application of the SCTimer/PWM uses the following configuration (all register values not listed in Table 416 are set to their default values):

Table 416. SCTimer/PWM configuration example

Configuration	Registers	Setting
Counter	CONFIG	Uses one counter (UNIFY = 1).
	CONFIG	Enable the autolimit for MAT0. (AUTOLIMIT = 1.)
	CTRL	Uses unidirectional counter (BIDIR_L = 0).
Clock base	CONFIG	Uses default values for clock configuration.
Match/Capture registers	REGMODE	Configure one match register for each match event by setting REGMODE_L bits 0,1, 2, 3, 4 to 0. This is the default.
Define match values	MATCH0/1/2/3/4	Set a match value MATCH0/1/2/4/5_L in each register. The match 0 register serves as an automatic limit event that resets the counter. without using an event. To enable the automatic limit, set the AUTOLIMIT bit in the CONFIG register.
Define match reload values	MATCHREL0/1/2/3/4	Set a match reload value RELOAD0/1/2/3/4_L in each register (same as the match value in this example).
Define when event 0 occurs	EV0_CTRL	• Set COMBMODE = 0x1. Event 0 uses match condition only. • Set MATCHSEL = 1. Select match value of match register 1. The match value of MAT1 is associated with event 0.
Define when event 1 occurs	EV1_CTRL	• Set COMBMODE = 0x1. Event 1 uses match condition only. • Set MATCHSEL = 2 Select match value of match register 2. The match value of MAT2 is associated with event 1.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **414 of 530**

Table 416. SCTimer/PWM configuration example

Configuration	Registers	Setting
Define when event 2 occurs	EV2_CTRL	• Set COMBMODE = 0x3. Event 2 uses match condition and I/O condition. • Set IOSEL = 0. Select input 0. • Set IOCOND = 0x0. Input 0 is LOW. • Set MATCHSEL = 0. Chooses match register 0 to qualify the event.
Define how event 2 changes the state	EV2_CTRL	Set STATEV bits to 1 and the STATED bit to 1. Event 2 changes the state to state 1.
Define when event 3 occurs	EV3_CTRL	• Set COMBMODE = 0x1. Event 3 uses match condition only. • Set MATCHSEL = 0x3. Select match value of match register 3. The match value of MAT3 is associated with event 3.
Define when event 4 occurs	EV4_CTRL	• Set COMBMODE = 0x1. Event 4 uses match condition only. • Set MATCHSEL = 0x4. Select match value of match register 4.The match value of MAT4 is associated with event 4.
Define when event 5 occurs	EV5_CTRL	• Set COMBMODE = 0x3. Event 5 uses match condition and I/O condition. • Set IOSEL = 0. Select input 0. • Set IOCOND = 0x3. Input 0 is HIGH. • Set MATCHSEL = 0. Chooses match register 0 to qualify the event.
Define how event 5 changes the state	EV5_CTRL	Set STATEV bits to 0 and the STATED bit to 1. Event 5 changes the state to state 0.
Define by which events output 0 is set	OUT0_SET	Set SET0 bits 0 (for event 0) and 3 (for event 3) to one to set the output when these events 0 and 3 occur.
Define by which events output 0 is cleared	OUT0_CLR	Set CLR0 bits 1 (for events 1) and 4 (for event 4) to one to clear the output when events 1 and 4 occur.
Configure states in which event 0 is enabled	EV0_STATE	Set STATEMSK0 bit 0 to 1. Set all other bits to 0. Event 0 is enabled in state 0.
Configure states in which event 1 is enabled	EV1_STATE	Set STATEMSK1 bit 0 to 1. Set all other bits to 0. Event 1 is enabled in state 0.
Configure states in which event 2 is enabled	EV2_STATE	Set STATEMSK2 bit 0 to 1. Set all other bits to 0. Event 2 is enabled in state 0.
Configure states in which event 3 is enabled	EV3_STATE	Set STATEMSK3 bit 1 to 1. Set all other bits to 0. Event 3 is enabled in state 1.
Configure states in which event 4 is enabled	EV4_STATE	Set STATEMSK4 bit 1 to 1. Set all other bits to 0. Event 4 is enabled in state 1.
Configure states in which event 5 is enabled	EV5_STATE	Set STATEMSK5 bit 1 to 1. Set all other bits to 0. Event 5 is enabled in state 1.

22.1 How to read this chapter

The watchdog timer is identical on all LPC84x 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 WatchDog 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 147.
- Enable the WWDT clock source (the watchdog oscillator) in the PDRUNCFG register (Table 173). 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 170).
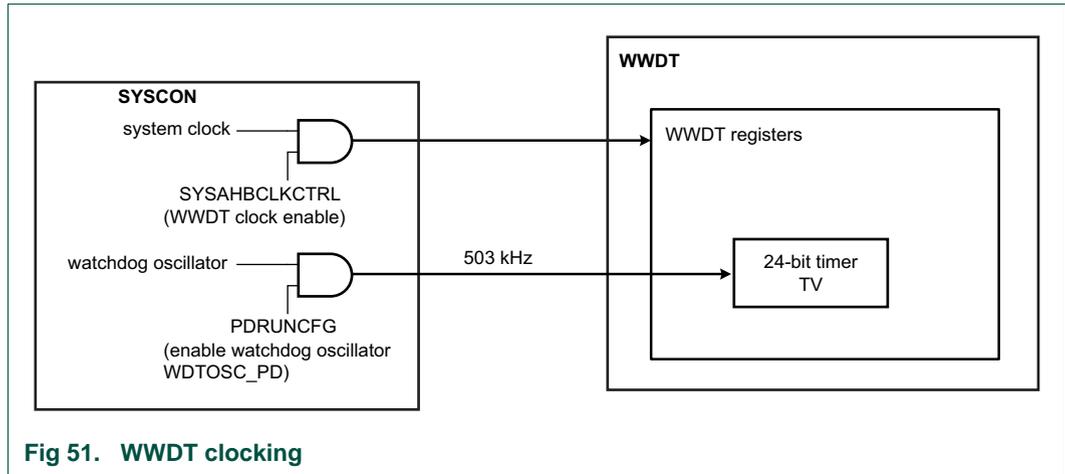
Fig 51. WWDT clocking

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 52. The synchronization logic (PCLK - WDCLK) is not shown in the block diagram.

Fig 52. Windowed Watchdog timer block diagram
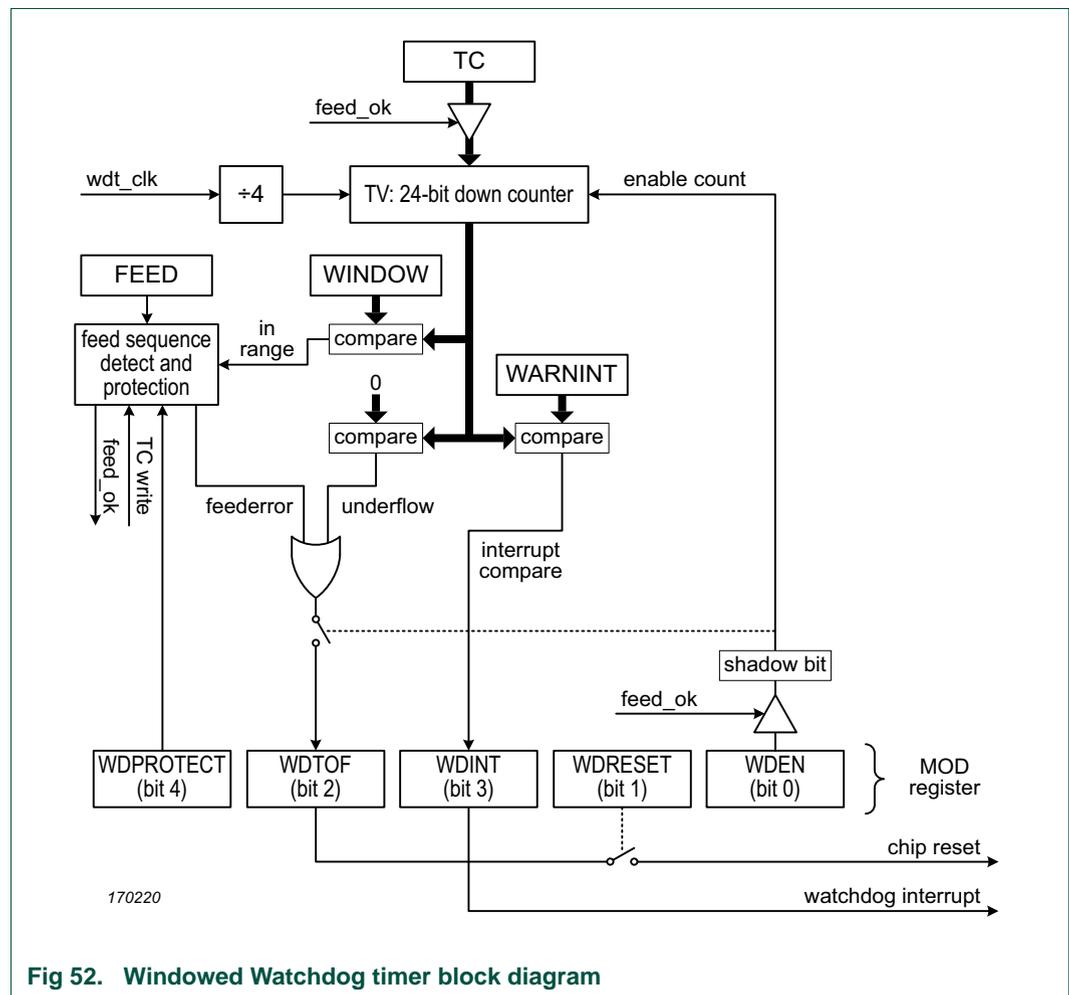
22.5.2 Clocking and power control

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 watchdog 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 watchdog 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 293).

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 417.

The reset value reflects the data stored in used bits only. It does not include the content of reserved bits.

Table 417. 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 418
TC	R/W	0x004	Watchdog timer constant register. This 24-bit register determines the time-out value.	0xFF	Table 420
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 421
TV	RO	0x00C	Watchdog timer value register. This 24-bit register reads out the current value of the Watchdog timer.	0xFF	Table 422
-	-	0x010	Reserved	-	-
WARNINT	R/W	0x014	Watchdog Warning Interrupt compare value.	0	Table 423
WINDOW	R/W	0x018	Watchdog Window compare value.	0xFF FFFF	Table 424

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 418. 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 418. 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 watchdog 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 0 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 watchdog 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 170 "Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description"

Table 419. 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 420. 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 421. 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 422. 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 423. 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **423 of 530**

Table 424. 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.

WDCLK / 4

Watchdog Counter   125A 1259 1258 1257

Early Feed Event

Watchdog Reset

Conditions:
WINDOW   = 0x1200
WARNINT   = 0x3FF
TC   = 0x2000
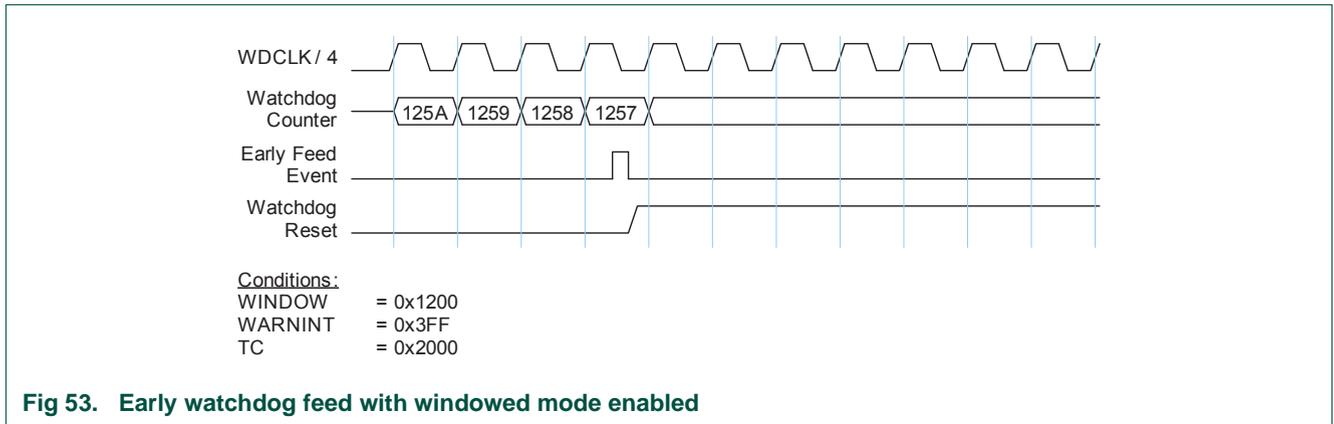
Fig 53. Early watchdog feed with windowed mode enabled

WDCLK / 4

Watchdog Counter   1201 1200 11FF 11FE 11FD 11FC 2000 1FFF 1FFE 1FFD 1FFC

Correct Feed Event

Watchdog Reset

Conditions:
WDWINDOW = 0x1200
WDWARNINT = 0x3FF
WDTC   = 0x2000
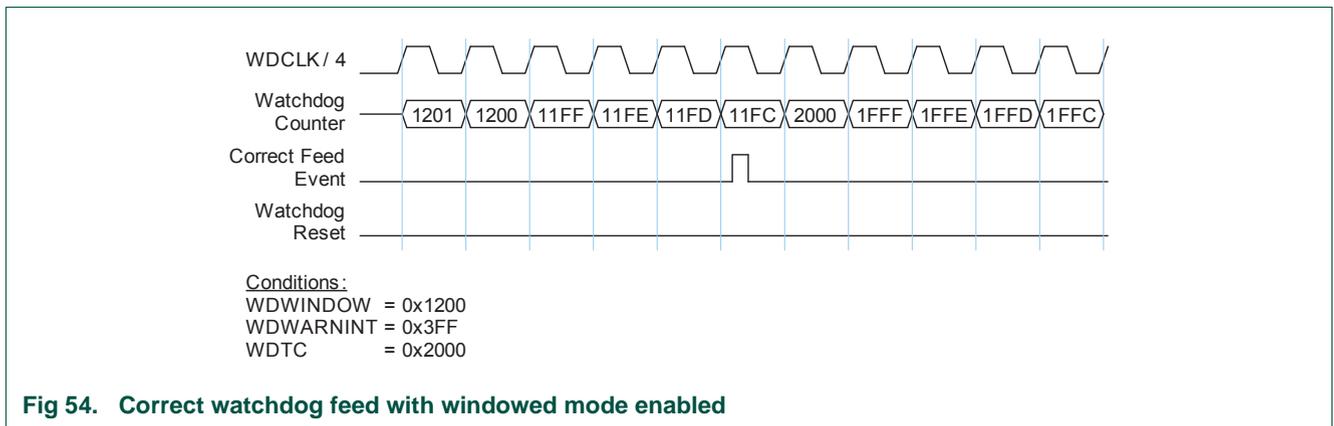
Fig 54. Correct watchdog feed with windowed mode enabled

WDCLK / 4

Watchdog Counter   0403 0402 0401 0400 03FF 03FE 03FD 03FC 03FB 03FA 03F9

Watchdog Interrupt

Conditions:
WINDOW   = 0x1200
WARNINT   = 0x3FF
TC   = 0x2000
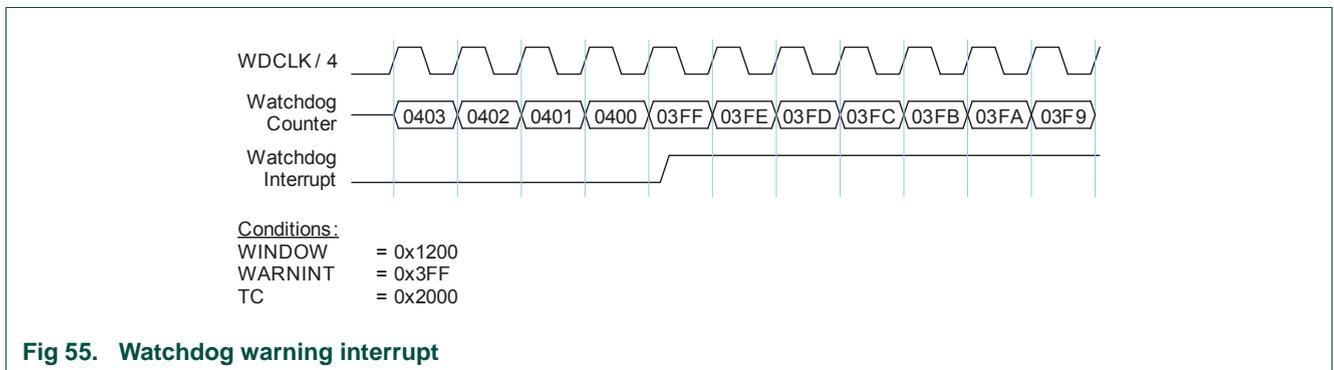
Fig 55. Watchdog warning interrupt

23.1 How to read this chapter

The self-wake-up timer is available on all LPC84x 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 low-power oscillator, or the WKTCLKIN pin. The 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 147) to enable the clock to the register interface.
- Clear the WKT reset using the PRESETCTRL register (Table 149).
- The WKT interrupt is connected to interrupt #15 in the NVIC. See Table 108.
- Enable the low power oscillator in the PMU (Table 295).
- Enable the FRO and FRO output in the PDRUNCFG register if used as the clock source for the timer (Table 173).
- To use an external clock source for the self-wake-up timer, enable the clock input for pin PIO0_28 in the DPDCRTL register (Table 295) and enable the external clock option in the self-wake-up timer CTRL register (see Table 426). 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 295.
- 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 295.
- See Section 15.7.1 to enable the various power down modes.

Turn off the low-power oscillator in deep power-down mode if not used for wake-up timing.

Fig 56. WKT clocking
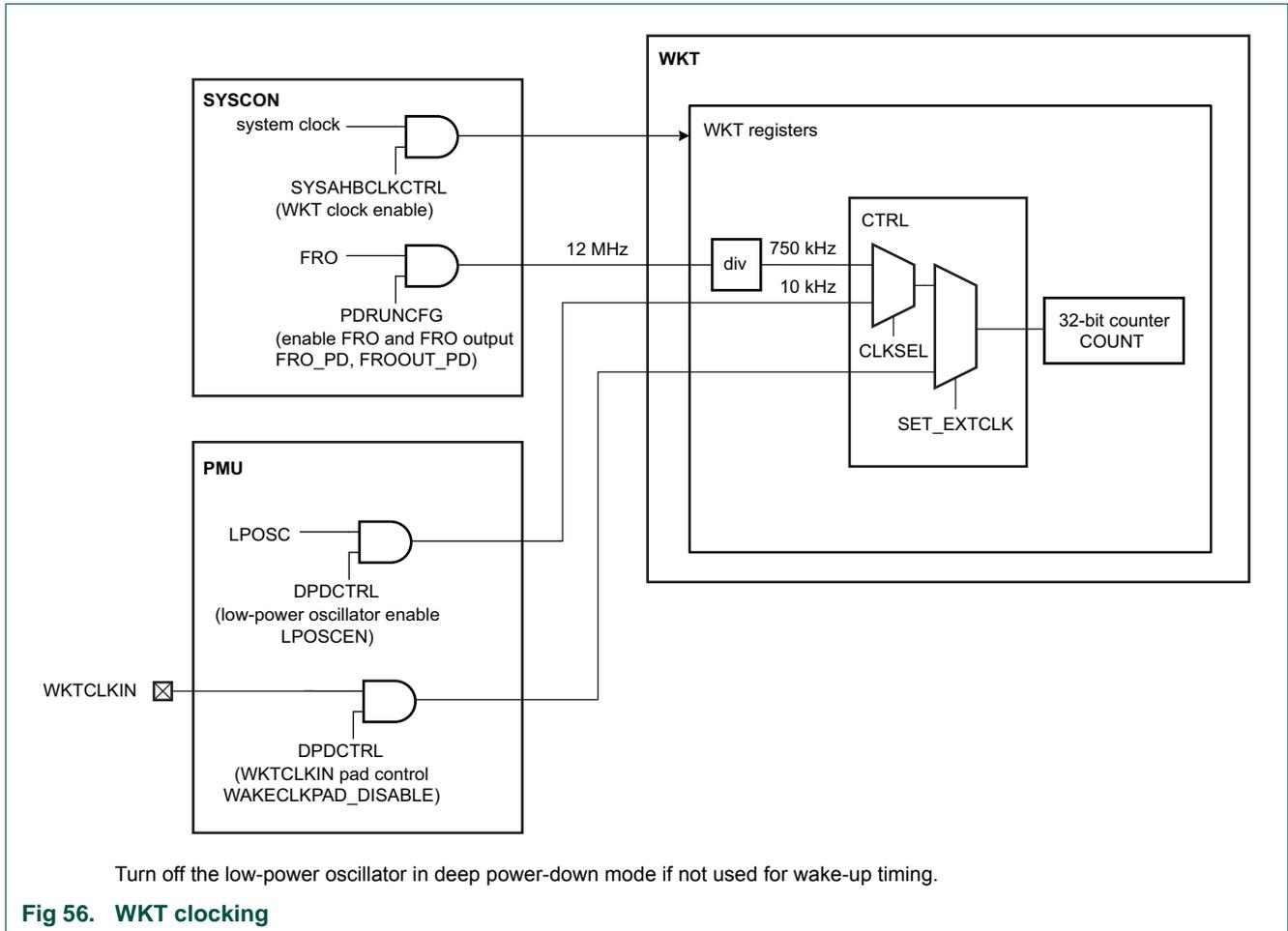
23.4 Pin description

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 426).

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 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 425. 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 426
COUNT	R/W	0xC	Counter register.	-	Table 427

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 426. 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 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 low-power oscillator must be enabled. The oscillator must also be set to remain active in Deep power-down if needed.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **427 of 530**

Table 426. 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 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 427. 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.	-

24.1 How to read this chapter

The MRT is available on all LPC84x parts.

24.2 Features

- 31-bit interrupt timer
- Four channels independently counting down from individually set values
- Repeat, bus-stall, and one-shot interrupt modes

24.3 Basic configuration

Configure the MRT using the following registers:

- In the SYSAHBCLKCTRL register, set bit 10 (Table 147) to enable the clock to the register interface.
- Clear the MRT reset using the PRESETCTRL register (Table 149).
- The global MRT interrupt is connected to interrupt #10 in the NVIC.

Fig 57.   MRT clocking
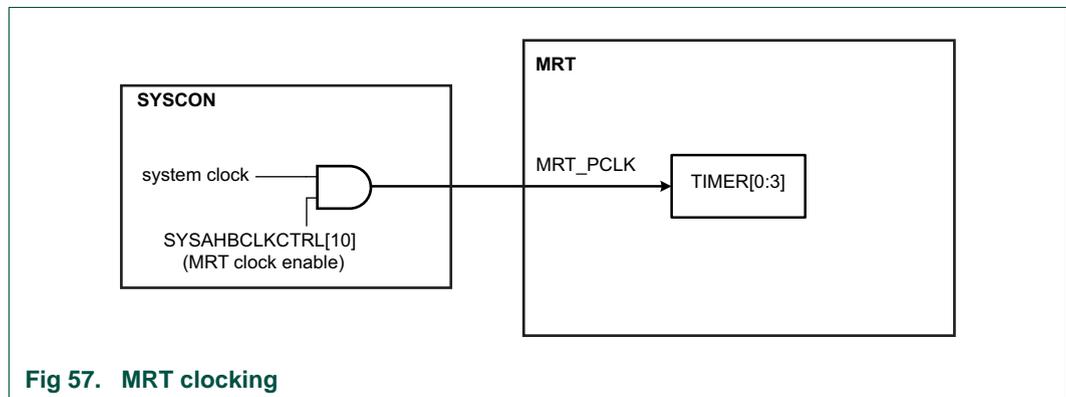
24.4 Pin description

The MRT has no configurable pins.

24.5 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 24.5.1.

- One-shot interrupt mode. See Section 24.5.2.
- Bus-stall mode.

The modes for each timer are set in the timer's control register. See Table 431.

ZEROS — MUX — -1 DEC — D Q TIMER — IRQ_GEN — IRQ0

D Q INTVAL

BUS

CONTROL

STAT

CHANNEL0

IRQ[1:3]

CHANNEL[1:3]
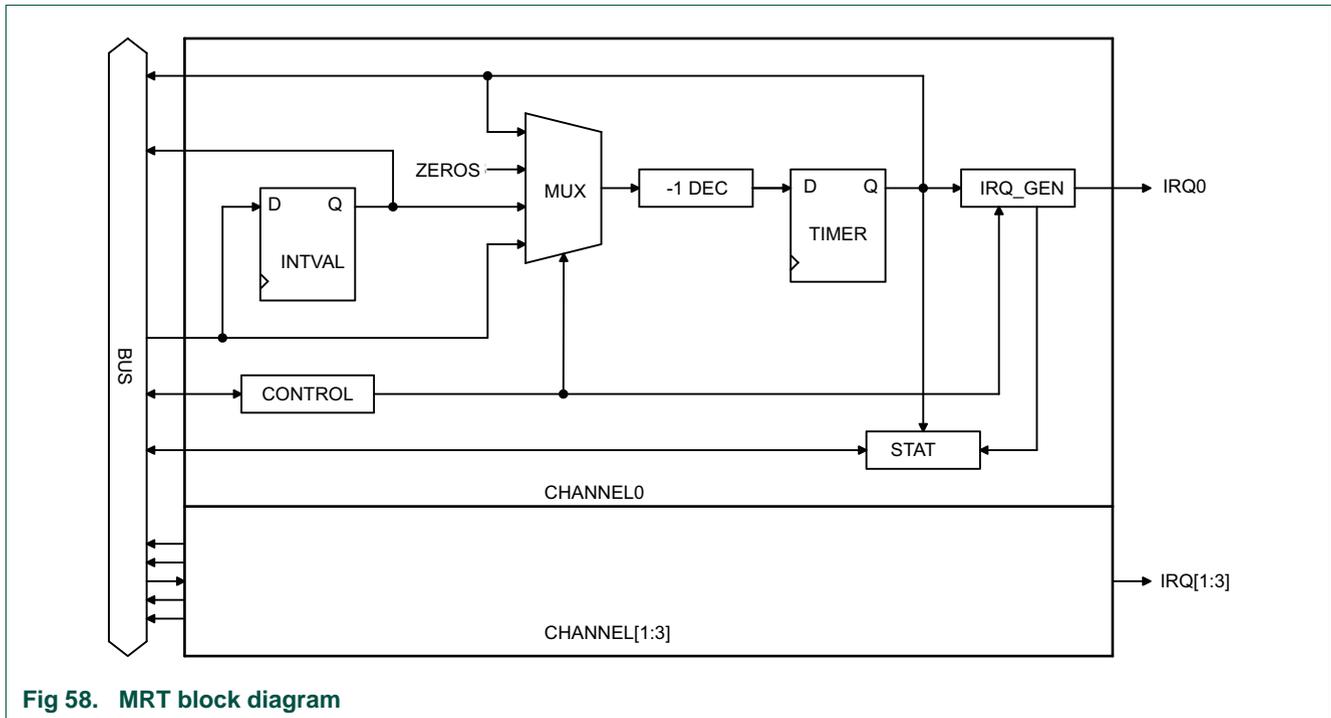
Fig 58.   MRT block diagram

24.5.1 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.

24.5.2 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.

24.5.3 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.

24.6 Register description

The reset values shown in Table 428 are POR reset values.

Table 428. 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 429
TIMER0	R	0x4	MRT0 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 430
CTRL0	R/W	0x8	MRT0 Control register. This register controls the MRT0 modes.	0	Table 431
STAT0	R/W	0xC	MRT0 Status register.	0	Table 432
INTVAL1	R/W	0x10	MRT1 Time interval value register. This value is loaded into the TIMER1 register.	0	Table 429
TIMER1	R/W	0x14	MRT1 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 430
CTRL1	R/W	0x18	MRT1 Control register. This register controls the MRT1 modes.	0	Table 431
STAT1	R/W	0x1C	MRT1 Status register.	0	Table 432
INTVAL2	R/W	0x20	MRT2 Time interval value register. This value is loaded into the TIMER2 register.	0	Table 429
TIMER2	R/W	0x24	MRT2 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 430
CTRL2	R/W	0x28	MRT2 Control register. This register controls the MRT2 modes.	0	Table 431
STAT2	R/W	0x2C	MRT2 Status register.	0	Table 432
INTVAL3	R/W	0x30	MRT3 Time interval value register. This value is loaded into the TIMER3 register.	0	Table 429
TIMER3	R/W	0x34	MRT3 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	Table 430
CTRL3	R/W	0x38	MRT3 Control register. This register controls the MRT modes.	0	Table 431
STAT3	R/W	0x3C	MRT3 Status register.	0	Table 432
IDLE_CH	R	0xF4	Idle channel register. This register returns the number of the first idle channel.	0	Table 433
IRQ_FLAG	R/W	0xF8	Global interrupt flag register	0	Table 434

24.6.1 Time interval register

This register contains the MRT load value and controls how the timer is reloaded. The load value is IVALUE -1.

Table 429. 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.	

24.6.2 Timer register

The timer register holds the current timer value. This register is read-only.

Table 430. 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 (0x00FF FFFF).	0x00FF FFFF
31	-	Reserved.	0

24.6.3 Control register

The control register configures the mode for each MRT and enables the interrupt.

Table 431. 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

24.6.4 Status register

This register indicates the status of each MRT.

Table 432. 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

24.6.5 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 433. 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

24.6.6 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 434. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **435 of 530**

25.1 How to read this chapter

The SysTick timer is available on all LPC84x parts.

25.2 Features

- Simple 24-bit timer.
- Uses dedicated exception vector.
- Clocked internally by the system clock or the system clock/2.

25.3 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 436). 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 436).
3. The calibration value of the SysTick timer is contained in the SYSTCKCAL register in the system configuration block SYSCON (see Table 165).

25.4 Pin description

The SysTick has no configurable pins.

25.5 General description

The block diagram of the SysTick timer is shown in Figure 59.
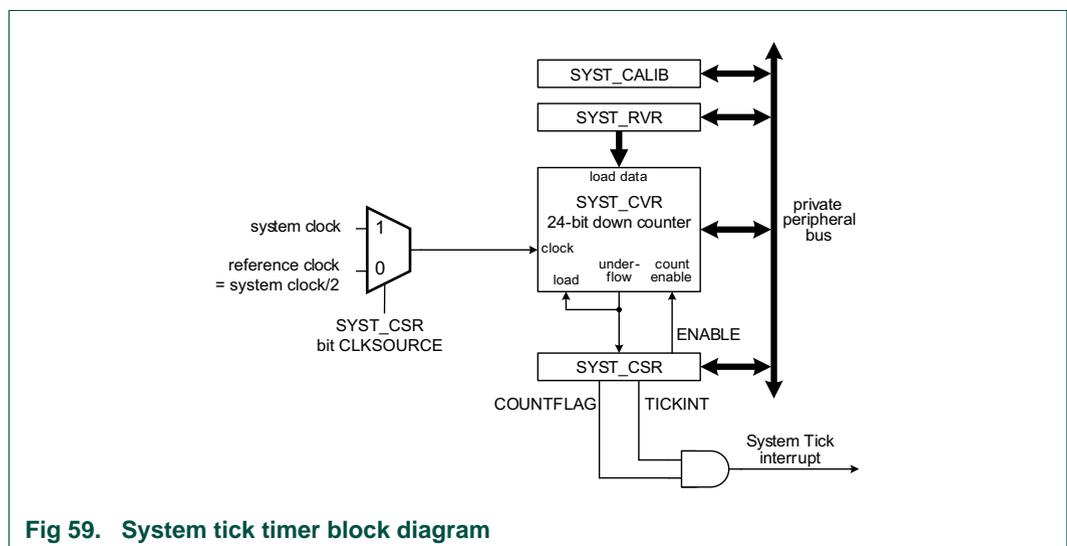
Fig 59.   System tick timer block diagram

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. 6 for details.

25.6 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. 6.

Table 435. 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	0
SYST_CVR	R/W	0x018	System Timer Current value register	0
SYST_CALIB	R/W	0x01C	System Timer Calibration value register	0x4

[1] Reset Value reflects the data stored in used bits only. It does not include content of reserved bits.

25.6.1 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. 6.

This register determines the clock source for the system tick timer.

Table 436. 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

25.6.2 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 437. 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.	0
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

25.6.3 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 438. 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.	0
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

25.6.4 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 165).

Table 439. System Timer Calibration value register (SYST_CALIB, 0xE000 E01C) bit description

Bit	Symbol	Value	Description	Reset value
23:0	TENMS		See Ref. 6.	0x4
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. 6.	0
31	NOREF		See Ref. 6.	0

25.7 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.

25.7.1 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 = 200 000 -1 = 199 999 = 0x00030D3F.

26.1 How to read this chapter

The Capacitive Touch is available on LPC845M301JBD64, LPC845M301JBD48, and LPC845M301JHI48 devices only.

26.2 Features

The Capacitive Touch supports:

- Up to nine mutual-capacitance touch sensors.
- Both GPIO port pin, and analog comparator measurement methods are available.
- DMA for continuous sequential polling of all sensors with no CPU intervention.
- Wake up from sleep, deep-sleep, and power-down modes.

26.3 Introduction

The Capacitive Touch module measures the change in capacitance of an electrode plate when an earth-ground connected object (for example, the finger or stylus) is brought within close proximity. Simply stated, the module delivers a small charge to an X capacitor (a mutual capacitance touch sensor), then transfers that charge to a larger Y capacitor (the measurement capacitor), and counts the number of iterations necessary for the voltage across the Y capacitor to cross a predetermined threshold.

The finger or stylus will impact the fringe capacitance field (between mutual electrodes of the X capacitor), effectively adding to, or subtracting from, the built up charge on the X capacitor. Once the threshold is crossed, the Y capacitor is discharged, and the process is repeated for the next X sensor. The number of iterations necessary to cross the threshold with an untouched sensor is used as the baseline count. For a given system and its grounding characteristics, a touched sensor triggers at either a lower or a higher count than an untouched sensor.

Once a calibrated and configured system operates normally, each X sensor generates either a 'no-touch' or a 'touch' event every time that sensor is polled. These events can be used to generate interrupts, and/or DMA activity. The system requires re-calibration when environmental factors change, including temperature and humidity, which affect fringe capacitance fields.

26.4 Quick setup guide

- Enable the bus clock to the Capacitive Touch module by setting the appropriate bit in one of the SYSAHBCLKCTRL registers in SYSCON.
- Take the module out of reset by setting the appropriate bit in one of the PRESETCTRL registers in SYSCON.
- Provide a function clock (FCLK) to the module by writing to the CAPTCLKSEL register in SYSCON.

- Choose a divider value to produce 4 MHz from the given FCLK, and program the divider into the FDIV field of the module in the CTRL register.

- Enable the fixed-pin functions CAPT_YH, CAPT_YL, and any of CAPT_X0, CAPT_X1, etc. that will be used, by writing to the appropriate PINENABLE register(s) in the SWM.

- Disable pull-ups and pull-downs on those pads by writing to the appropriate registers in the IOCON.

- If the analog comparator measurement method is used, enable one of the ACMP_I fixed pin functions by writing to the appropriate PINENABLE register(s) in the SWM. If the device does not provide a shared CAPT_YH and ACMP_I pad, the package pins must be wire-OR'd externally.

- Identify to the module the set of X pins to use by writing to the XPINSEL field in the control register. The maximum number of X pins available for a given device can be read from the XMAX field in the status register and is equal to XMAX+1.

- In many systems a finger-touched sensor triggers at a lower count than an untouched sensor. If touch triggers lower than no-touch, set the TCHLOWER bit in the POLL_TCNT register.

- Set the threshold, and various other timing and counting parameters, by writing to the POLL_TCNT register (an initial POLL_NOW polling operation can be launched in order for the baseline no-touch trigger point to be determined).

- Enable interrupts and/or DMA requests in the CTRL register, as required, and start polling.

26.5 General description

26.5.1 Pin usage

The Capacitive Touch module uses one standard GPIO pin for YL and up to sixteen (depending on product family and part number) standard GPIOs for X0 through XMAX.

One standard GPIO is used for YH on products with nine or fewer X pins. For products with greater than nine X pins, there is a second YH GPIO pin.

YH, YL, and X functions are typically enabled on their pins using the switch matrix or IOCON, depending on the product family. Additionally, the set of X pins that the application will use must be enabled, or identified to the module by writing '1's to their bit positions in the XPINSEL field of the control register.

On devices with an analog comparator, if the analog comparator measurement method is used, an analog comparator input must be enabled via the switch matrix or IOCON, and also wire-OR'd externally to the YH pin. Some devices may provide a shared YH and ACMP_IN pin, in which case the external connection is not necessary.

Figure 60 shows the connections necessary for a basic button configuration using two sensors. The same connection scheme applies to slider and touch-wheel sensors. Additional sensors are added in parallel, each with its own X pin.
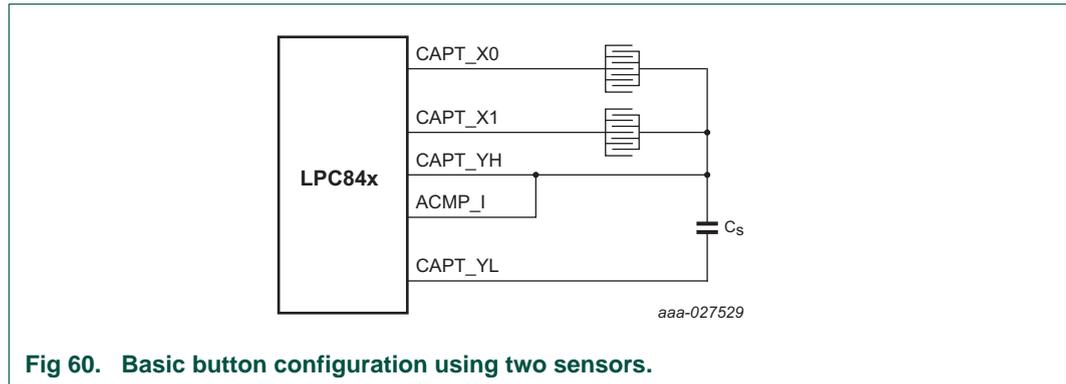
Fig 60. Basic button configuration using two sensors.

26.5.2 Basic measurement algorithm

The basic measurement algorithm, represents the states of the hardware module when any given sensor is polled.

Step state:

1. 0 Reset
2. 1 Charge
3. 2 Transfer
4. 3 Measure

In all states, X channels not identified to the module (that is, disabled in the XPINSEL field of the CTRL register) is kept in High-Z or is held low, depending on the value written to the XPINUSE field of the CTRL register.

Between measurements, the module remains in Step 0.

During a measurement, steps 1 to 3 are repeated until the ACMP (or YH port pin) reaches threshold or the module times out (too many iterations).

On completion, the module returns to Step 0 before any new measurements are taken.

26.5.3 Function clock divider

The function clock (FCLK) provided to the module is divided internally by the function clock divider, to produce the divided FCLK. All clocking and counting discussed fin this chapter is in reference to the divided FCLK, unless otherwise noted.

Divided FCLK can be any of FCLK-divided-by-1 through FCLK-divided-by-16. A good rule-of-thumb is to provide an FCLK frequency so that the divided FCLK can be 4 MHz or less.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **442 of 530**

26.6 Timing and counting parameters

26.6.1 Threshold count

The THRESHOLD COUNT (TCNT) sets the count boundary between touch and no-touch. The software configures the module to consider a touch to occur at either a higher or a lower count than the threshold count, based on the system specifics.

26.6.2 Time-out count

If the time-out count reaches before the YH port pin or the analog comparator crosses the zero-to-one voltage threshold, there is a time-out event and the X measurement is terminated. This should be set to a value much greater than the larger of the expected touch / no-touch counts. Under normal circumstances the time-out count should never be reached because each X sensor measurement triggers either a touch or a no-touch event before reaching time-out.

26.6.3 Poll counter

The poll counter sets the time delay between polling rounds (successive sets of X measurements). After each polling round completes, the module will wait 4096 x POLL divided FCLKs before starting the next polling round. Measuring too often can be inefficient and unnecessary because finger movement is relatively slow.

26.6.4 Measurement delay

The measurement delay specifies the number of divided FCLKs the module will wait after entering Measure Voltage state (Section 26.5.2, step 3), before sampling the YH port pin or analog comparator output. This gives the analog comparator time to react to the transferred charge.

26.6.5 Reset delay

The reset delay specifies the number of divided FCLKs the module will remain in Reset or Draining Cap. state (Section 26.5.2, step 0) before starting the next X measurement in the polling round. Larger capacitors may need more time to drain and/or the unselected Xs may need more time to go back to their resting state.

26.7 Modes of Operation

26.7.1 Measurement methods

The Capacitive Touch module can use two distinct methods of measuring the voltage across the measurement capacitor, based on the setting of the TRIGGER bit in the Control register.

26.7.1.1 YH port pin measurement

In Measure Voltage state (Section 26.5.2, step 3), the module samples the YH port pin (which is in input mode). Until the voltage on the pad increases above the V_{IH} (HIGH-level input voltage) of the pad, the module will sample '0', above VIH it will sample '1' (that is, the YH port pin has "triggered").

26.7.1.2 Analog comparator measurement

In Measure Voltage state (Section 26.5.2, step 3), the module samples the analog comparator output, which is connected internally to the module. The analog comparator must be enabled and properly configured, and one of the comparator analog inputs must be enabled and connected to the YH port pin. On some devices the YH port pin and an analog comparator input may share a pad or pin. Otherwise, the YH port pin and the analog comparator input pin must be connected externally (wire-OR'd).

Until the voltage on the analog comparator input increases above the configured threshold of the comparator, the module will sample '0', above that it will sample '1' (that is, the analog comparator has "triggered").

The voltage threshold of the comparator may be set lower than the VIH of the YH port pin, which allows for faster sensing than the YH port pin measurement method.

26.7.2 Polling modes

The Capacitive Touch module has the following polling modes, based on the POLLMODE field in the Control register.

26.7.2.1 Inactive

No measurements are taken, no polls are performed. The module remains in the Reset / Draining Cap. state (Section 26.5.2, step 0). Counters are inactive. Entering this mode will immediately terminate any polling / measurement in progress. Entering this mode does not affect the five interrupt flags in the Status register.

26.7.2.2 Poll Now

Immediately launches (ignoring Poll Delay) a one-time-only, simultaneous poll of all X pins that are enabled in the XPINSEL field of the Control register, then stops, returning to Reset / Draining Cap. state (Section 26.5.2, step 0). Note that all enabled X pins are activated concurrently, rather than walked one-at-a-time.

Useful for baselining threshold count and time-out count.

26.7.2.3 Continuous

Polling rounds are continuously performed, by walking through the enabled X pins. Each polling round is preceded by the time delay specified by the poll counter in the POLL_TCNT register (which can be zero).

26.7.2.3.1 Low power polling

The software may choose to take certain steps to conserve power, before putting the device into Deep-sleep / Power-down, with the intention of using a touch event to wake up the processor.

The software would do the following before going into a low-power state:

- Switch FCLK to a low-frequency, low-power clock source that stays on in deep-sleep or power-down mode, and adjust the Function clock divider, as well as the fields in the POLL_TCNT register, for this new frequency. Polling once every 250 ms or so may be sufficient for detecting a wake-up touch.

- Power-down the analog comparator, and switch to YH port pin measurement method.

- Choose the set of X pins in XPINSEL that will be polled for wake-up.
- Make sure that only the YESTOUCH interrupt is enabled.
- Enable the Capacitive Touch wake-up source in the SYSCON.
- Put the device into deep sleep or power down mode.
- A touch event will wake up the CPU and cause an interrupt. Software would then revert to the regular configuration.

26.7.3 Polling Types

26.7.3.1 Normal

Normal polling (TYPE = 0) treats all enabled X elements individually, and they are scanned one-at-a-time in ascending order. This is used for buttons, basic sliders, etc.

26.8 Touch Data

The Touch information for the most recent measurement can be read by either the CPU or the DMA engine from the Touch Data register. The information available are:

- COUNT: The count value reached at trigger or time-out. If the measurement resulted in a time-out, COUNT will contain the time-out count value minus one, that is, $2^{TOUT} - 1$.
- XVAL: For normal polling (TYPE = 0) in continuous mode (POLLMODE = 2), XVAL contains the index of the X pin for the current measurement. For a multiple-pin measurement in poll now mode (POLLMODE = 1), XVAL will contain the index of the lowest enabled X pin.
- ISTOUCH: Set if the trigger is due to a touch event, cleared if due to a no-touch event.
- ISTO: Set if the measurement resulted in a time-out event.
- SEQ: Sequence number, a 4-bit rolling counter that increments after all enabled X pins are scanned in a polling round.
- CHANGE: An indicator that will be set for one bus clock at the end of an X measurement, while the touch data are changing. Touch data read while this bit is high are invalid.

26.9 Interrupts and DMA

26.9.1 Interrupts

The Capacitive Touch module has five types of interrupts, each with a unique flag in the status register.

26.9.1.1 Yes touch

The YESTOUCH flag is set when the YH port pin or analog comparator triggers at a count value that is greater than TCNT (if the TCHLOWER bit equals '0'), or a count value that is less-than-or-equal-to TCNT (if the TCHLOWER bit equals '1').

26.9.1.2 No touch

The NOTOUCH flag is set when the YH port pin or analog comparator triggers at a count value that is less-than-or-equal-to TCNT (if the TCHLOWER bit equals '0'), or a count value that is greater than TCNT (if the TCHLOWER bit equals '1').

26.9.1.3 Poll done

The POLLDONE flag is set at the end of each polling round, or when a Poll Now completes.

26.9.1.4 Timeout

The TIMEOUT flag is set if the count reaches the time-out count value before the YH port pin or analog comparator crosses the zero-to-one voltage threshold.

26.9.1.5 Overrun

The OVERRUN flag is set if the Touch Data register is updated before the software has read the previous data, and the trigger is due to a touch event (ISTOUCH = '1'). The OVERRUN flag will not be set on no-touch events (ISTOUCH = '0', in which case touch data are silently overwritten) or on TIMEOUTs. If the WAIT bit in the Control register equals '1', the OVERRUN flag will not be set.

26.9.2 DMA

Based on the setting of the DMA field in the Control register, the Capacitive Touch module can generate DMA requests, so Touch Data register contents can be read by the DMA engine instead of the CPU.

26.9.2.1 Request on touch

DMA requests are generated on touch events. In this case, the DMA engine can be configured to read only the 16 lower bits of the Touch Data register, since the upper 16 bits are not needed.

26.9.2.2 Request on both touch and no touch

DMA requests are generated on both touch and no-touch events.

26.9.2.3 Request on both plus time-out

DMA requests are generated on touch, no-touch, and time-out events.

26.10 Register description

Table 440. Register overview: base address 0x4006 0000

Name	Access	Address offset	Description	Reset value	Reference
CTRL	R/W	0x000	Control register. Contains control and configuration fields.	0x0	26.10.1
STATUS	RO	0x004	Status register. Contains the interrupt flags, busy, and XMAX information.	Part-number dependent	26.10.2
POLL_TCNT	R/W	0x008	Poll and measurement counter register. This sets up the polling counter and measurement counter rules.	0x0	26.10.3
INTENSET	R/W	0x010	Interrupt enable read and set. Write '1' to set.	0x0	26.10.4
INTENCLR	WO	0x004	Interrupt enable clear. Write '1' to clear.	0x0	26.10.5
INTSTAT	RO	0x018	Interrupt status. Logical AND of interrupt flags in STATUS register and corresponding bits in INTENSET register.	0x0	26.10.6
TOUCH	RO	0x20	Touch data register. TOUCH contains details of the most recent X sensor interrogation.	0x0	26.10.7
ID	RO	0xFFC	Block ID	0xE100 0000	26.10.8

26.10.1 Control register

The control register is used to specify the modes and parameters for the operation of the module.

It is recommended that the POLLMODE field is set to 0x0 before making changes to this register. No writes to this register should be made while INCHANGE (bit 15) equals '1'.

Table 441. Control register (CTRL, offset 0x000) bit description

Bit	Symbol	Value	Description	Reset value	Access
1:0	POLLMODE		Selects the method of polling. This field may only change from 0x0 to another value. Therefore, if software wishes to change between two active polling modes, it must first write 0x0 before selecting the new mode.	0x0	R/W
		0x0	Inactive.		
		0x1	Poll Now.		
		0x2	Continuous.		
		0x3	Reserved.		
3:2	TYPE		Selects the polling type and sensor arrangement.	0x0	R/W
		0x0	Normal.		
		0x1	Reserved.		
		0x2	Reserved.		
		0x3	Reserved.		
4	TRIGGER		Selects the measurement method.	0x0	R/W
		0x0	YH port pin measurement.		
		0x1	Analog comparator measurement.		

Table 441. Control register (CTRL, offset 0x000) bit description

Bit	Symbol	Value	Description	Reset value	Access
5	WAIT		Controls when the next X measurement in the sequence may commence.	0x0	R/W
		0x0	The next X measurement starts at the normal time.		
		0x1	When the ISTOUCH bit in the TOUCH register equals '1', the module will wait until the TOUCH register is read before starting the next measurement. Other-wise, measurements continue.		
7:6	DMA		Controls how DMA triggers are generated.	0x0	R/W
		0x0	No DMA.		
		0x1	Trigger on touch.		
		0x2	Trigger on both touch and no-touch.		
		0x3	Trigger on both plus time-out.		
11:8	FDIV	0x0 – 0xF	Function clock divider. The function clock is divided by FDIV+1 to produce the divided FCLK for the module.	0x0	R/W
13:12	XPINUSE		Determines how X pins enabled in the XPINSEL field are controlled when not active.	0x0	R/W
		0x0	Inactive X pins are High-Z.		
		0x1	Inactive X pins are driven low.		
		0x2	Reserved.		
		0x2	Reserved.		
14	-	-	Reserved.	-	-
15	INCHANGE		Shows the status of the most recent update to the control register.		
		0x0	The last change has propagated.	0x0	RO
		0x1	The last change has not propagated. Propagation time is dependent on synchronization between the bus clock and divided FCLK domains. Wait for IN-CHANGE = 0 before updating this register.		
31:16	XPINSEL		Selects which of the available X pins are enabled. Writing '1' to a bit enables the corresponding X pin, '0' disables. Bit 16 (XPINSEL[0]) controls X0. Bit 17 (XPINSEL[1]) controls X1. Bit 18 (XPINSEL[2]) controls X2.	-	

26.10.2 Status register

The status register indicates the status of the module in terms of touch events and polling. The interrupt flags are bits in this register, which are cleared by writing '1' to their bit position.

Table 442. Status register (STATUS, offset 0x004) bit description

Bit	Symbol	Description	Reset value	Access
0	YESTOUCH	Set if a touch has been detected, based on the count at which trigger occurred.	0	R/W1-to- clear
1	NOTOUCH	Set if a no-touch has been detected, based on the count at which trigger occurred.	0	R/W1-to- clear
2	POLLDONE	Set at the end of a polling round, or when a POLLNOW completes.	0	R/W1-to- clear
3	TIMEOUT	Set if the count reaches the time-out count value before a trigger occurs.	0	R/W1-to- clear
4	OVERRUN	Set if the Touch Data register has been up-dated before software has read the previous data, and the ISTOUCH bit in Touch Data equals '1'. Will not be set if ISTOUCH = '0' in which case touch data are silently overwritten.	0	R/W1-to- clear
7:5	-	Reserved.	-	-
8	BUSY	Set while a poll is currently in progress, otherwise cleared.	0	RO
15:9	-	Reserved.	-	-
19:16	XMAX	The maximum number of X pins available for a given device is equal to XMAX+1.	Part-number dependent	RO

26.10.3 Poll and measurement counter register

The poll and measurement counter register contains the counting, delay, and threshold information, and is the basis for all measurements.

Make changes to this register while the POLLMODE field in the control register equals 0x0.

Table 443. Poll and Measurement Counter Register (POLL_TCNT, offset 0x008) bit description

Bit	Symbol	Value	Description	Reset value	Access
11:0	TCNT		Sets the count boundary in divided FCLKs between touch and no-touch, based on the TCHLOWER bit (Bit 31).	0x000	R/W
15:12	TOUT		Sets the count value at which a time-out event occurs if a measurement has not triggered. The time-out count value is calculated as 2^{TOUT}. TOUT must be less than 13, so the legal val-ues are 0, 1, ... 12.	0x0	R/W
23:16	POLL		Sets the time delay between polling rounds (successive sets of X measurements). After each polling round completes, the module will wait 4096 x POLL divided FCLKs before starting the next polling round.	0x00	R/W
25:24	MDELAY		Specifies the time delay after entering step 3 (measure voltage state), before sampling the YH port pin or analog comarator output.	0x0	R/W
		0x0	Don't wait.		
		0x1	Wait 3 divided FCLKs.		
		0x2	Wait 5 divided FCLKs.		
		0x3	Wait 9 divided FCLKs.		

Table 443. Poll and Measurement Counter Register (POLL_TCNT, offset 0x008) bit description

Bit	Symbol	Value	Description	Reset value	Access
27:26	RDELAY		Specifies the number of divided FCLKs the module will remain in step 0 before starting the next X measurement in the polling round.	0x0	R/W
		0x0	1 divided FCLK.		
		0x1	2 divided FCLKs.		
		0x2	4 divided FCLKs.		
		0x3	8 divided FCLKs.		
30:28	-		Reserved.	-	-
31	TCHLOW-ER		Specifies whether a touched sensor triggers at a lower or higher count than an untouched sensor. TOUCHLOWER = 0: Trigger at count > TCNT is a touch. Trigger at count ≤ TCNT is a no-touch. TOUCHLOWER = 1: Trigger at count ≤ TCNT is a touch. Trigger at count > TCNT is a no-touch.	0x0	R/W

26.10.4 Interrupt Enable Read and Set Register

The INTENSET register is used to set interrupt enables for individual interrupt flags in the STATUS register. An enabled interrupt can contribute to the module's interrupt request. Reading INTENSET returns the current values of the interrupt enables

Interrupt enables are set by writing '1' to their bit position in INTENSET. Writing '0' has no effect. Use INTENCLR to clear interrupt enables.

Table 444. Interrupt Enable Read and Set Register (INTENSET, offset 0x010) bit description

Bit	Symbol	Value	Description	Reset value	Access
0	YESTOUCH			0x0	R/W1-to-set
		0	Disabled		
		1	enabled		
1	NOTOUCH			0x0	R/W1-to-set
		0	Disabled		
		1	enabled		
2	POLLDONE			0x0	R/W1-to-set
		0	Disabled		
		1	enabled		
3	TIMEOUT			0x0	R/W1-to-set
		0	Disabled		
		1	enabled		
4	OVERRUN			0x0	R/W1-to-set
		0	Disabled		
		1	enabled		

26.10.5 Interrupt Enable Clear Register

The write-only INTENCLR register is used to clear interrupt enables for individual interrupt flags in the STATUS register.

Interrupt enables are cleared by writing '1' to their bit position in INTENCLR. Writing '0' has no effect.

Table 445. Interrupt Enable Clear Register (INTENCLR, offset 0x014) bit description

Bit	Symbol	Description	Reset value	Access
0	YESTOUCH	Writing '1' clears this interrupt enable.	0x0	W1-to-clear
1	NOTOUCH	Writing '1' clears this interrupt enable.	0x0	W1-to-clear
2	POLLDONE	Writing '1' clears this interrupt enable.	0x0	W1-to-clear
3	TIMEOUT	Writing '1' clears this interrupt enable.	0x0	W1-to-clear
4	OVERRUN	Writing '1' clears this interrupt enable.	0x0	W1-to-clear

26.10.6 Interrupt Status Register

The INTSTAT register is used to view the status of enabled interrupts only. Reading INTSTAT returns the logical AND of the interrupt flags in the STATUS register with their corresponding interrupt enable bits.

Table 446. Interrupt status register (INTSTAT, offset 0x018) bit description

Bit	Symbol	Description	Reset value	Access
0	YESTOUCH	0 = no interrupt, 1 = enabled and pending.	0x0	R/O
1	NOTOUCH	0 = no interrupt, 1 = enabled and pending.	0x0	R/O
2	POLLDONE	0 = no interrupt, 1 = enabled and pending.	0x0	R/O
3	TIMEOUT	0 = no interrupt, 1 = enabled and pending.	0x0	R/O
4	OVERRUN	0 = no interrupt, 1 = enabled and pending.	0x0	R/O

26.10.7 Touch Data Register

The TOUCH register contains the data from the most recent X measurement.

Table 447. Touch data register (TOUCH, offset 0x020) bit description

Bit	Symbol	Description	Reset value	Access
11:0	COUNT	Contains the count value reached at trigger or time-out.	0x0	R/O
15:12	XVAL	Contains the index of the X pin for the current measurement, or lowest X for a multiple-pin poll now measurement.	0x0	R/O
16	ISTOUCH	'1' if the trigger is due to a touch event, '0' if the trigger is due to a no-touch event.	0x0	R/O
17	ISTO	'1' if the measurement resulted in a time-out event, '0' otherwise.	0x0	R/O
19:18	-	Reserved.	-	-
23:20	SEQ	Contains the 4-bit sequence number, which increments at the end of each polling round.	0x0	R/O
30:24	-	Reserved.	-	-
31	CHANGE	Will be '1' for one bus clock at the end of each X measurement while the data are changing, otherwise '0'. Touch data read while this bit is '1' are invalid.		

26.10.8 ID register

The ID register identifies the type and revision of the module. A generic SW driver can make use of this information register to implement module-type or implementation-specific behavior.

Table 448. ID register (ID, offset 0xFFC) bit description

Bit	Symbol	Description	Reset value	Access
7:0	APERTURE	Aperture: encoded as (aperture size/4K) -1, so 0x00 is a 4 K aperture.	0x00	R/O
11:8	MINOR_REV	Minor revision of module implementation, starting at 0. Software compatibility is expected between minor revisions.	0x0	R/O
15:12	MAJOR_REV	Major revision of module implementation, starting at 0. There may not be software compatibility between major revisions.	0x0	R/O
31:16	ID	Unique module identifier for this IP block.	0xE100	R/O

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 449. Pinout summary

Package	ADC channels available
HVQFN33	ADC_0 to ADC_11
HVQFN48	ADC_0 to ADC_11
LQFP48	ADC_0 to ADC_11
LQFP64	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 1.2 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 173. 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 454 and Section 27.7.5.
- Use the SYSAHBCLKCTRL register (Table 147) 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 108.
- The ADC analog inputs are enabled in the switch matrix block.See Table 196.
- The power to the ADC block is controlled by the PDRUNCFG register in the SYSCON block. See Table 173.

- 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 454.

- Configure the ADC for the appropriate analog supply voltage using the TRM register (Table 467). The default setting assumes $V_{DDA} \geq 2.7$ V.

aaa-028398
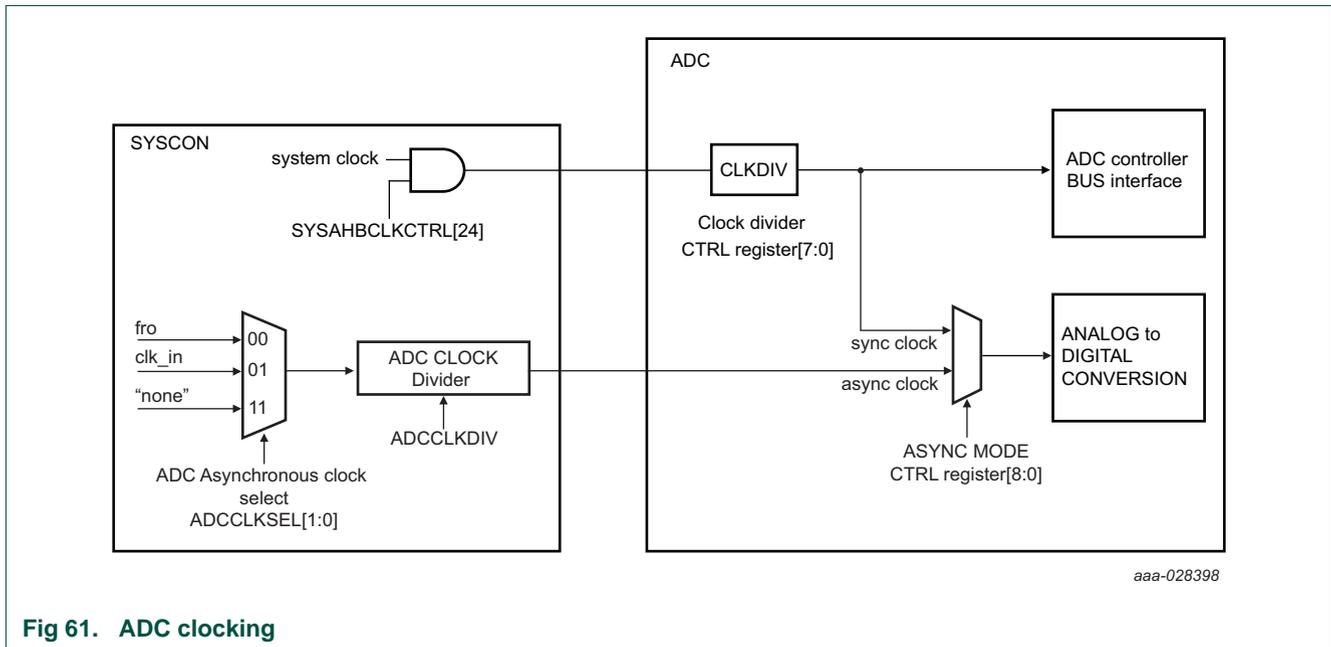
Fig 61. ADC clocking

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 452.

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.

Related registers:

- Table 455 "A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description"
- Table 456 "A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description"

Table 450. ADC hardware trigger inputs

Input #	Source	Description
0	-	No hardware trigger.
1	PININT0_IRQ	GPIO_INT interrupt 0.
2	PININT1_IRQ	GPIO_INT interrupt 1.
3	SCT0_OUT3	SCT output 3.
4	SCT0_OUT4	SCT output 4.
5	T0_MAT3	CTIMER match 3.
6	CMP0_OUT_ADC	Analog comparator output.
7	GPIO_INT_BMAT	GPIO_INT bmatch.
8	ARM_TXEV	ARM core TXEV event.

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 451. 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 452. ADC pin description

Function	Direction	Type	Connect to	Use register	Reference	Description
ADC_0	AI	external to pin	PIO0_7	PINENABLE0	Table 196	Analog input channel 0.
ADC_1	AI	external to pin	PIO0_6	PINENABLE0	Table 196	Analog input channel 1.
ADC_2	AI	external to pin	PIO0_14	PINENABLE0	Table 196	Analog input channel 2.
ADC_3	AI	external to pin	PIO0_23	PINENABLE0	Table 196	Analog input channel 3.
ADC_4	AI	external to pin	PIO0_22	PINENABLE0	Table 196	Analog input channel 4.
ADC_5	AI	external to pin	PIO0_21	PINENABLE0	Table 196	Analog input channel 5.
ADC_6	AI	external to pin	PIO0_20	PINENABLE0	Table 196	Analog input channel 6.
ADC_7	AI	external to pin	PIO0_19	PINENABLE0	Table 196	Analog input channel 7.
ADC_8	AI	external to pin	PIO0_18	PINENABLE0	Table 196	Analog input channel 8.
ADC_9	AI	external to pin	PIO0_17	PINENABLE0	Table 196	Analog input channel 9.
ADC_10	AI	external to pin	PIO0_13	PINENABLE0	Table 196	Analog input channel 10.
ADC_11	AI	external to pin	PIO0_4	PINENABLE0	Table 196	Analog input channel 11.

UM11029

User manual

All information provided in this document is subject to legal disclaimers.

Rev. 1.6 — 8 December 2017

© NXP Semiconductors N.V. 2017. All rights reserved.

457 of 530

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
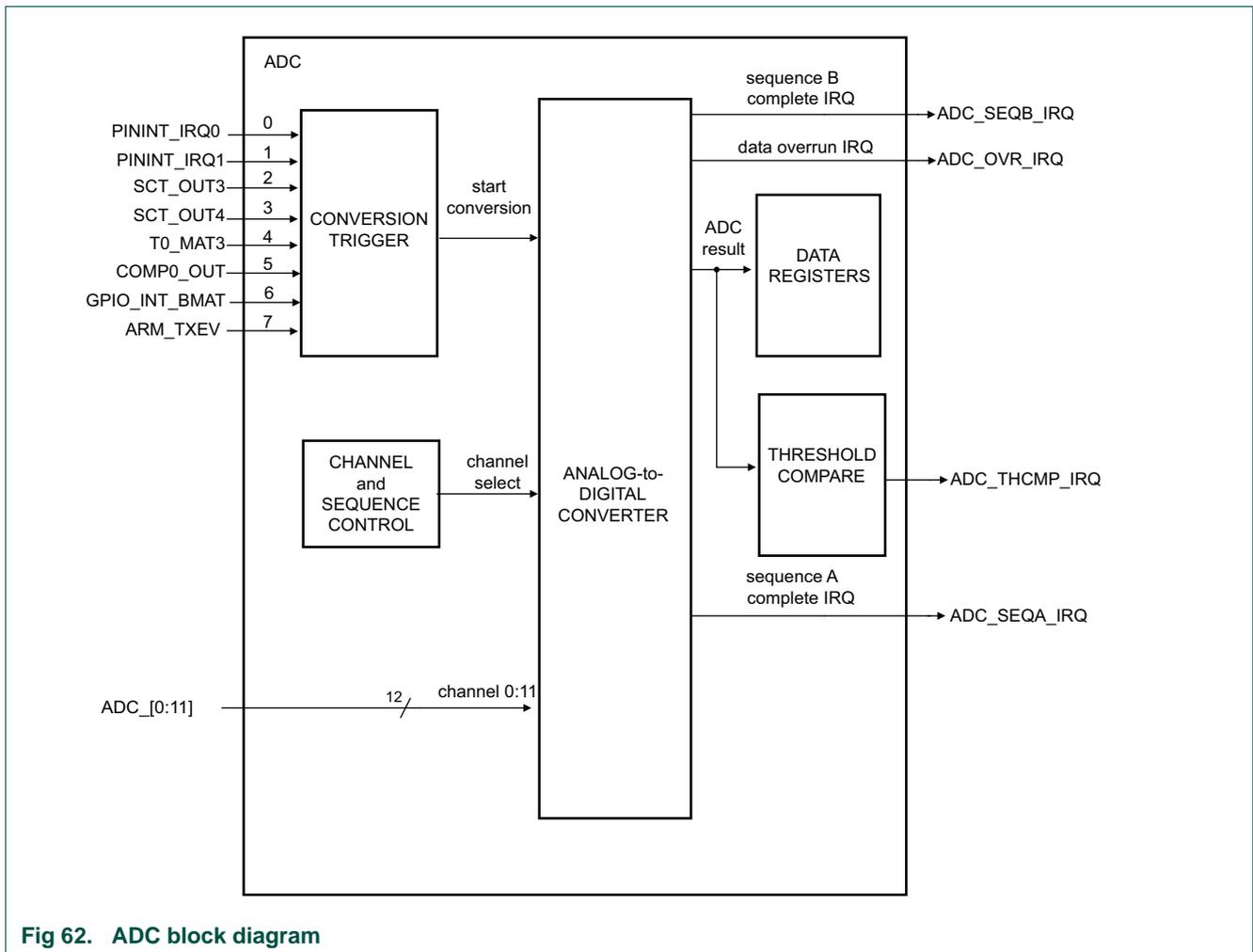
Fig 62. ADC block diagram

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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **458 of 530**

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 30 MHz. The ADC clock drives the successive approximation process.

A fully accurate conversion requires 25 of these ADC clocks.

27.6 Register description

The reset value reflects the data stored in used bits only. It does not include reserved bits content.

Table 453. 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 454
-	-	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 455
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 456
SEQA_GDAT	R/W	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 457
SEQB_GDAT	R/W	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 458
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 459
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 459
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 459
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 459
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 459
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 459
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 459
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 459
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 459
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 459

Table 453. Register overview : ADC (base address 0x4001 C000)

Name	Access	Address offset	Description	Reset value	Reference
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 459
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 459
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 460
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 461
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 462
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 463
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 464
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 465
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 466
TRM	R/W	0x06C	ADC trim register.	0x0000 0F00	Table 467

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 454. 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 30 MHz for 1.2 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 455. 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.	0x00
			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.	
14:12	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.	0x0
			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.	
17:15	-		Reserved.	-
18	TRIGPOL		Select the polarity of the selected input trigger for this conversion sequence.	0
			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	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 455. 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.	
25:20	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	N/A
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
29	LOWPRIO		Set priority for sequence A.	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.	

User manual **Rev. 1.6 — 8 December 2017** **463 of 530**

Table 455. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
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 456. 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.	0x00
			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.	
14:12	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.	0x0
			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.	
17:15	-		Reserved.	-
18	TRIGPOL		Select the polarity of the selected input trigger for this conversion sequence.	0
			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	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.	
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:	0
			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	Enable synchronization. The hardware trigger bypass is not enabled.	
		1	Bypass synchronization. The hardware trigger bypass is enabled.	

UM11029

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **465 of 530**

Table 456. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
25:20	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	N/A
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
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.	

Table 456. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
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.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **467 of 530**

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 457. 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 457. 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 458. 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.	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **469 of 530**

Table 458. 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 459. 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 459. 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.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **472 of 530**

Table 460. 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 461. 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 462. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **473 of 530**

Table 463. 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 464. 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 464. 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 465. 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 465. 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	

UM11029

User manual **Rev. 1.6 — 8 December 2017** **477 of 530**

Table 465. 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 466. 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 466. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **479 of 530**

Table 466. 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 467. 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 321). 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.

28.1 Basic configuration

The DAC is configured using the following registers:

1. Power: Use the PDRUNCFG register to power the DAC.

2. Peripheral clock: Use the SYSAHBCLKCTRL register to enable the clock to the DAC register interface. See Section 8.6.22.

3. Pins: Enable the DAC fixed-pin function in the relevant SWM PINENABLE register, and also set the DACENABLE bit in the relevant IOCON register.

4. DMA: The DAC can be connected to the DMA controller (see Section 28.5.2). For DMA connections.

28.2 Features

- 10-bit digital to analog converter.
- Resistor string architecture.
- Buffered output.
- Power-down mode.
- Selectable speed vs. power.
- Maximum update rate of 1 MHz.

28.3 Architecture
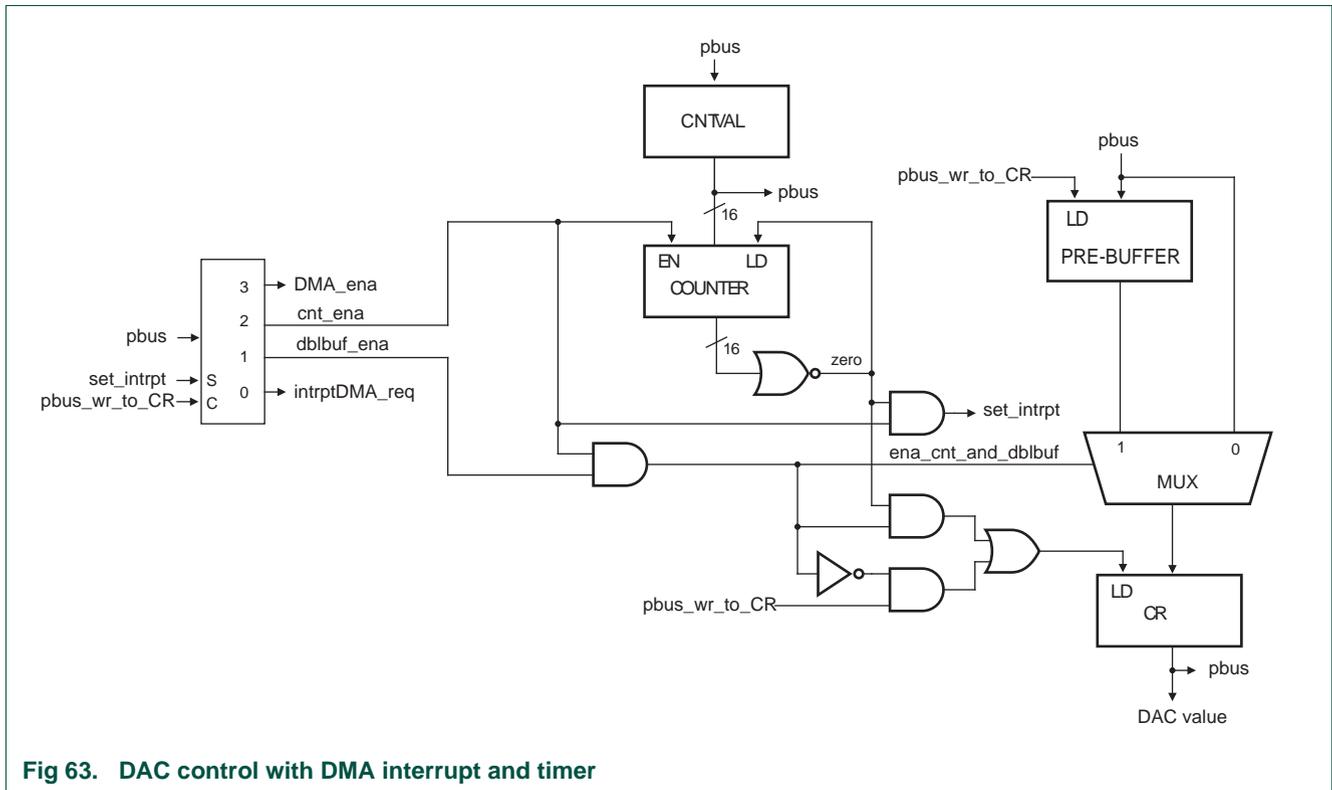
Fig 63. DAC control with DMA interrupt and timer

28.4 Pin description

Table 468 gives a brief summary of each of DAC related pins.

Table 468. D/A Pin Description

Pin	Type	Description
DAC_OUT	Output	**Analog Output.** After the selected settling time after the CR is written with a new value, the voltage on this pin (with respect to V_{SSA}) is VALUE $\times$ ((V_{REFP})/1024). Note that DAC_OUT is disabled when the CPU is in Deep-sleep, Power-down, or Deep Power-down modes.
V_{REFP}	Reference	**Voltage Reference.** This pin provides a voltage reference level for the ADC and DAC. Note: V_{REFP} should be tied to VDD(3V3) if the ADC and DAC are not used.
V_{DDA}, V_{SSA}	Power	**Analog Power and Ground.** These should typically be the same voltages as V_{DD} and V_{SS}, but should be isolated to minimize noise and error. **Note: V_{DDA} should be tied to VDD(3V3) and V_{SSA} should be tied to V_{SS} if the ADC and DAC are not used.**

28.5 Register description

Table 469. Register overview: DAC (base address 0x4001 4000(DAC0), 0x4001 8000 (DAC1))

Name	Access	Address offset	Description	Reset value[1]	Table
CR	R/W	0x000	D/A Converter Register. This register contains the digital value to be converted to analog and a power control bit.	0	470
CTRL	R/W	0x004	DAC Control register. This register controls DMA and timer operation.	0	471
CNTVAL	R/W	0x008	DAC Counter Value register. This register contains the reload value for the DAC DMA/Interrupt timer.	0	472

[1] Reset value reflects the data stored in used bits only. It does not include reserved bits content.

28.5.1 D/A Converter Register

This read/write register includes the digital value to be converted to analog, and a bit that trades off performance vs. power. Bits 5:0 are reserved for future, higher-resolution D/A converters.

Table 470. D/A Converter Register (CR - address 0x4001 4000 (DAC0), 0x4001 8000 (DAC1)) bit description

Bit	Symbol	Value	Description	Reset Value
5:0	-		Reserved. Read value is undefined, only zero should be written.	NA
15:6	VALUE		After the selected settling time after this field is written with a new VALUE, the voltage on the DAC_OUT pin (with respect to V_{SSA}) is VALUE $\times$ ((V_{REFP})/1024).	0
16	BIAS		Settling time The settling times noted in the description of the BIAS bit are valid for a capacitance load on the DAC_OUT pin not exceeding 100 pF. A load impedance value greater than that value will cause settling time longer than the specified time. One or more graphs of load impedance vs. settling time will be included in the final data sheet.	0
		0	The settling time of the DAC is 1 μs max, and the maximum current is 700 mA. This allows a maximum update rate of 1 MHz.	
		1	The settling time of the DAC is 2.5 μs and the maximum current is 350 μA. This allows a maximum update rate of 400 kHz.	
31:17	-		Reserved. Read value is undefined, only zero should be written.	NA

28.5.2 D/A Converter Control register

This read/write register enables the DMA operation and controls the DMA timer.

Table 471. D/A Control register (CTRL - address 0x4001 4004 (DAC0), 0x4001 8004 (DAC1)) bit description

Bit	Symbol	Value	Description	Reset Value
0	INT_DMA_REQ		DMA interrupt request	0
		0	Clear on any write to the CR register.	
		1	Set by hardware when the timer times out.	

Table 471. D/A Control register (CTRL - address 0x4001 4004 (DAC0), 0x4001 8004 (DAC1)) bit description

Bit	Symbol	Value	Description	Reset Value
1	DBLBUF_ENA		Double buffering	0
		0	Disable	
		1	Enable. When this bit and the CNT_ENA bit are both set, the double-buffering feature in the CR register will be enabled. Writes to the CR register are written to a pre-buffer and then transferred to the CR on the next time-out of the counter.	
2	CNT_ENA		Time-out counter operation	0
		0	Disable	
		1	Enable	
3	DMA_ENA		DMA access	0
		0	Disable	
		1	Enable. DMA Burst Request Input 7 is enabled for the DAC.	
31:4	-		Reserved. Read value is undefined, only zero should be written.	NA

28.5.3 D/A Converter Counter Value register

This read/write register contains the reload value for the Interrupt/DMA counter.

Table 472. D/A Converter Counter Value register (CNTVAL - address 0x4001 4008 (DAC0), 0x4001 8008 (DAC1) bit description

Bit	Symbol	Description	Reset Value
15:0	VALUE	16-bit reload value for the DAC interrupt/DMA timer.	0
31:16	-	Reserved	-

28.6 Operation

28.6.1 DMA counter

When the counter enable bit CNT_ENA in DACCTRL is set, a 16-bit counter will begin counting down, at the rate selected by PCLK, from the value programmed into the DACCNTVAL register. The counter is decremented Each time the counter reaches zero, the counter will be reloaded by the value of DACCNTVAL and the DMA request bit INT_DMA_REQ will be set in hardware.

Note that the contents of the DACCTRL and DACCNTVAL registers are read and write accessible, but the timer itself is not accessible for either read or write.

If the DMA_ENA bit is set in the DACCTRL register, the DAC DMA request will be routed to the GPDMA. When the DMA_ENA bit is cleared, the default state after a reset, DAC DMA requests are blocked.

28.6.2 Double buffering

Double-buffering is enabled only if both, the CNT_ENA and the DBLBUF_ENA bits are set in DACCTRL. In this case, any write to the CR register will only load the pre-buffer, which shares its register address with the CR register. The CR itself will be loaded from the pre-buffer whenever the counter reaches zero and the DMA request is set. At the same time the counter is reloaded with the COUNTVAL register value.

Reading the CR register will only return the contents of the CR register itself, not the contents of the pre-buffer register.

If either the CNT_ENA or the DBLBUF_ENA bits are 0, any writes to the CR address will go directly to the CR register.

29.1 How to read this chapter

The analog comparator is available on all LPC84x parts.

29.2 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

29.3 Basic configuration

Configure the analog comparator using the following registers:

- In the SYSAHBCLKCTRL register, set bit 19 (Table 147) to enable the clock to the register interface.
- You can enable or disable the power to the analog comparator through the PDRUNCFG register (Table 173).
- Clear the analog comparator peripheral reset using the PRESETCTRL register (Table 149).
- The analog comparator interrupt is connected to interrupt #11 in the NVIC.
- Configure the analog comparator pin functions through the switch matrix. See Section 29.4.

29.3.1 Connect the comparator output to the SCTimer/PWM

The comparator output function (ACMP_O) can be used to start or stop the SCTimer/PWM or, more generally, to create an SCT event without assigning a pin through the switch matrix. To create an SCT event internally connected to the comparator output, select the comparator output as one of the SCT inputs through the INPUT MUX (see Table 288 "SCT input mux registers 0 to 3 (SCT0_INMUX[0:3], address 0x4002 C020 (SCT0_INMUX0) to 0x4002 C02C (SCT0_INMUX3)) bit description".

29.3.2 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 455 "A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description" and Table 456 "A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description".

29.4 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 10.3.1 "Connect an internal signal to a package pin" to assign the analog comparator output to any pin on the LPC84x package.

Table 473. Analog comparator pin description

Function	Type	Pin	Description	SWM register	Reference
ACMP_I1	I	PIO0_0	Comparator input 1	PINENABLE0	Table 196
ACMP_I2	I	PIO0_1	Comparator input 2	PINENABLE0	Table 196
ACMP_I3	I	PIO0_14	Comparator input 3	PINENABLE0	Table 196
ACMP_I4	I	PIO0_23	Comparator input 4	PINENABLE0	Table 196
ACMP_I5	I	PIO0_30	Comparator input 5	PINENABLE0	Table 196
ACMP_O	O	any	Comparator output	PINASSIGN11	Table 192
VDDCMP	I	PIO0_6	External reference voltage source for 32-stage Voltage Ladder.	PINENABLE0	Table 196

29.5 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 64 and Table 475).

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.

Input 7 of the multiplexer connects the DACOUT0.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **491 of 530**
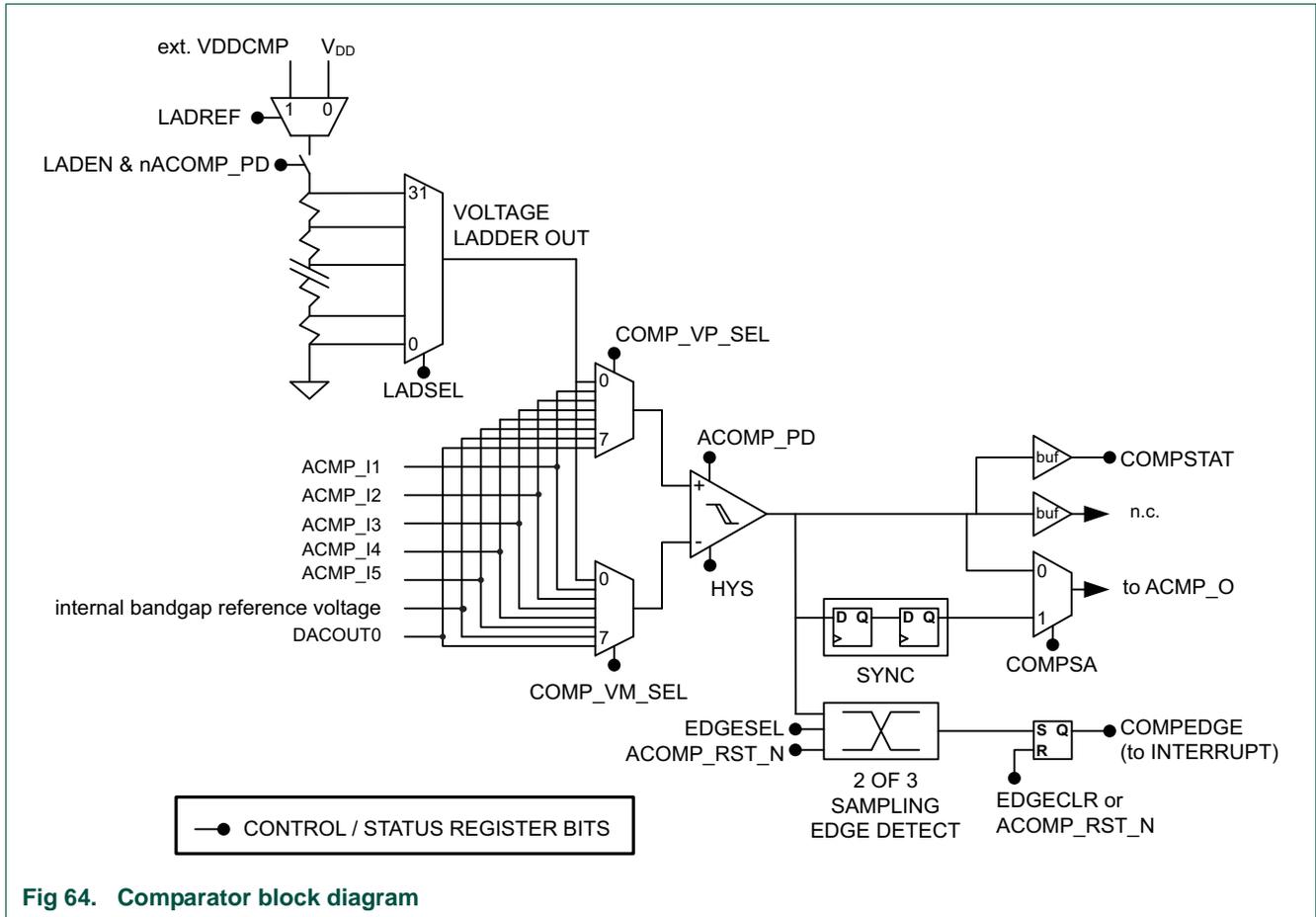
Fig 64. Comparator block diagram

29.5.1 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.

29.5.2 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.

29.5.3 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.

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **492 of 530**

29.5.4 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 147 "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.

The comparator output can be routed to the SCT via the switch matrix allowing to capture the time of a voltage crossing or to count crossings in either or both directions. See Section 29.3.1 "Connect the comparator output to the SCTimer/PWM".

29.6 Register description

Table 474. 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 475
LAD	R/W	0x004	Voltage ladder register	0	Table 476

29.6.1 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 475 are reserved and should be written as 0.

Table 475. 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 475. 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	DACOUT0.	
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	DACOUT0.	
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	-

29.6.2 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 476. 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **495 of 530**

30.1 How to read this chapter

The CRC engine is available on all LPC84x parts.

30.2 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)

30.3 Basic configuration

Enable the clock to the CRC engine in the SYSAHBCLKCTRL register (Table 147, bit 13).

30.4 Pin description

The CRC engine has no configurable pins.

30.5 General description

The Cyclic Redundancy Check (CRC) generator with programmable polynomial settings supports several CRC standards commonly used.
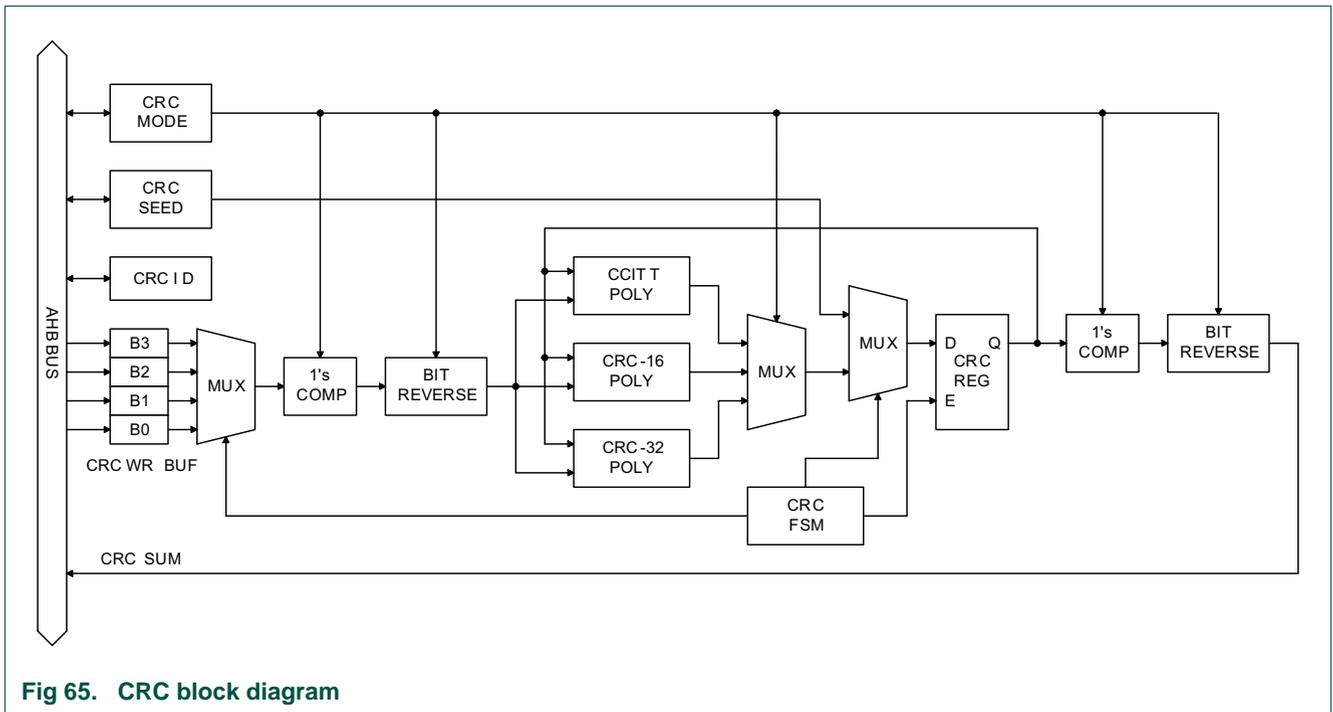
Fig 65. CRC block diagram

30.6 Register description

Table 477. Register overview: CRC engine (base address 0x5000 0000)

Name	Access	Address offset	Description	Reset value	Reference
MODE	R/W	0x000	CRC mode register	0x0000 0000	Table 478
SEED	R/W	0x004	CRC seed register	0x0000 FFFF	Table 479
SUM	RO	0x008	CRC checksum register	0x0000 FFFF	Table 480
WR_DATA	WO	0x008	CRC data register	-	Table 481

30.6.1 CRC mode register

Table 478. CRC mode register (MODE, address 0x5000 0000) 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

UM11029

All information provided in this document is subject to legal disclaimers.

© NXP Semiconductors N.V. 2017. All rights reserved.

User manual **Rev. 1.6 — 8 December 2017** **498 of 530**

30.6.2 CRC seed register

Table 479. 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

30.6.3 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 480. 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

30.6.4 CRC data register

This register is a Write-only register containing the data block for which the CRC sum will be calculated.

Table 481. CRC data register (WR_DATA, address 0x5000 0008) 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.	-

30.7 Functional description

The following sections describe the register settings for each supported CRC standard:

30.7.1 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

30.7.2 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

30.7.3 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

UM11029

User manual **Rev. 1.6 — 8 December 2017** **500 of 530**

31.1 How to read this chapter

The ROM-based 32-bit integer division routines are available on all parts.

31.2 Features

- Performance-optimized signed/unsigned integer division.
- Performance-optimized signed/unsigned integer division with remainder.
- ROM-based routines to reduce code size.
- Support for integers up to 32 bit.
- ROM calls can easily be added to EABI-compliant functions to overload "/" and "%" operators in C.

31.3 General description

The API calls to the ROM are performed by executing functions which are pointed by a pointer within the ROM Driver Table. Figure 66 shows the pointer structure used to call the Integer divider API.
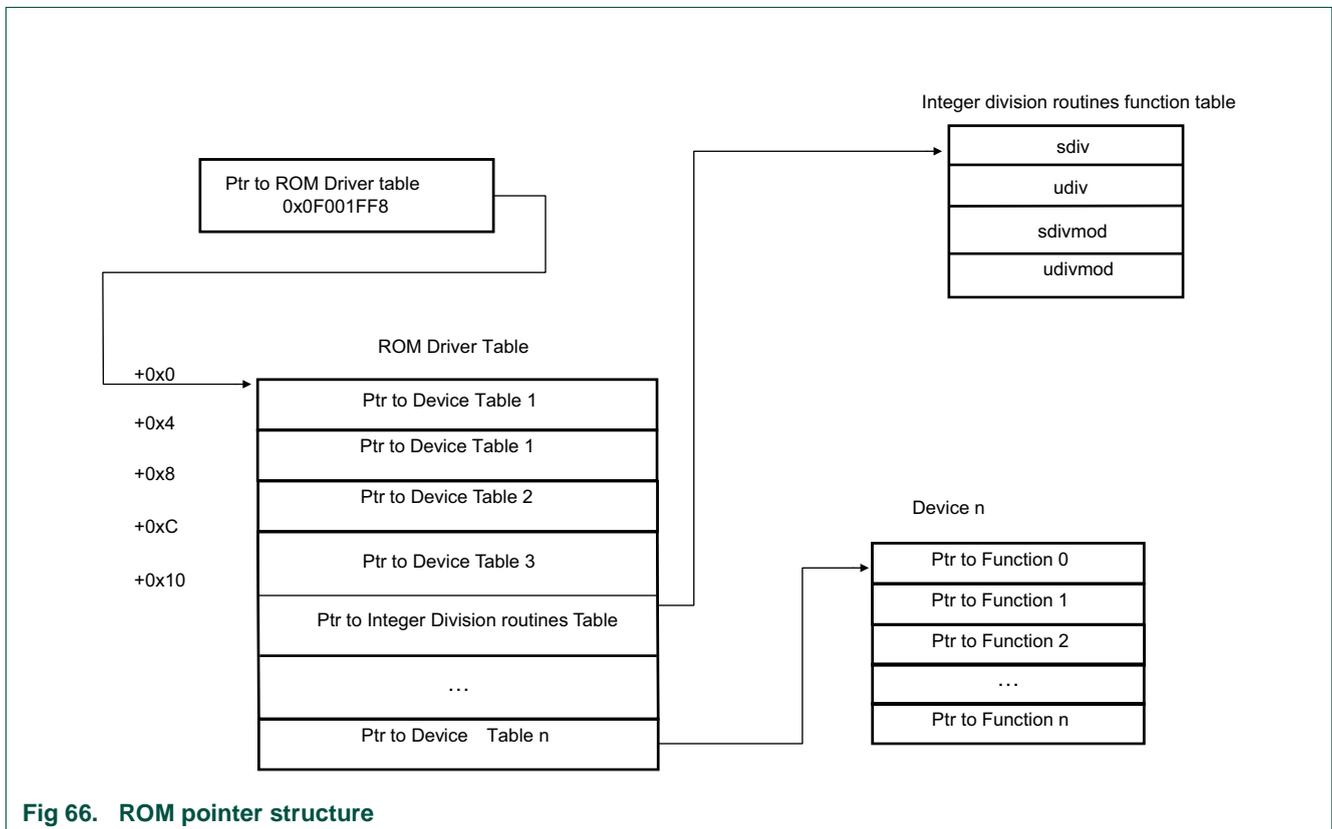
Fig 66.   ROM pointer structure

31.4 API description

The integer division routines perform arithmetic integer division operations and can be called in the application code through simple API calls.

Table 482. Divide API calls

API call	Description	Reference
`int(*sdiv)(int numerator, int denominator);`	Signed integer division	Table 483
`unsigned(*udiv) (int numerator, int denominator);`	Unsigned integer division	Table 484
`sdiv_t (*sdivmod)(int numerator, int denominator);`	Signed integer division with remainder	Table 485
`udiv_t (*udivmod)(unsigned numerator, unsigned denominator);`	Unsigned integer division with remainder	Table 486

The following function prototypes are used:

```
typedef struct {
     int quot;        /*!< Quotient */
     int rem;         /*!< Remainder */
} IDIV_RETURN_T;

typedef struct {
     unsigned quot;/*!< Quotient */
     unsigned rem;/*!< Reminder */
} UIDIV_RETURN_T;

typedef struct {
     int (*sidiv)(int numerator, int denominator); /*!< Signed integer division */
     unsigned (*uidiv)(unsigned numerator, unsigned denominator); /*!< Unsigned
     integer division */
     IDIV_RETURN_T (*sidivmod)(int numerator, int denominator); /*!< Signed integer
     division with remainder */
     UIDIV_RETURN_T (*uidivmod)(unsigned numerator, unsigned denominator);/*!<
     Unsigned integer division
          with remainder */
} ROM_DIV_API_T;


ROM_DIV_API_T const *pROMDiv = LPC_ROM_API->divApiBase;
```

The ROM API table shown in Section 3.5.2 "ROM-based APIs" must be included in the code.

31.4.1 DIV signed integer division

Table 483. sidiv

Routine	sidiv
Prototype	`int(*sidiv)(int32_t numerator, int32_t denominator);`

Table 483. sidiv

Routine	sidiv
Input parameter	numerator: Numerator signed integer. denominator: Denominator signed integer.
Return	Signed division result without remainder.
Description	Signed integer division

31.4.2 DIV unsigned integer division

Table 484. uidiv

Routine	uidiv
Prototype	`int(*uidiv)(int32_t numerator, int32_t denominator);`
Input parameter	numerator: Numerator signed integer. denominator: Denominator signed integer.
Return	Unsigned division result without remainder.
Description	Unsigned integer division

31.4.3 DIV signed integer division with remainder

Table 485. sidivmod

Routine	sidivmod
Prototype	`IDIV_RETURN_T (*sidivmod) (int32_t numerator, int32_t denominator);`
Input parameter	numerator: Numerator signed integer. denominator: Denominator signed integer.
Return	Signed division result remainder.
Description	Signed integer division with remainder

31.4.4 DIV unsigned integer division with remainder

Table 486. uidivmod

Routine	uidivmod
Prototype	`UIDIV_RETURN_T(*uidiv)(uint32_t numerator, uint32_t denominator);`
Input parameter	numerator: Numerator unsigned integer. denominator: Denominator unsigned integer.
Return	Unsigned division result with remainder.
Description	Unsigned integer division

31.5 Functional description

31.5.1 Signed division

The example C-code listing below shows how to perform a signed integer division via the ROM API.

```
/* Divide (-99) by (+6) */
int32_t result;
result = pROMDiv->sidiv(-99, 6);
/* result now contains (-16) */
```

31.5.2 Unsigned division with remainder

The example C-code listing below shows how to perform an unsigned integer division with remainder via the ROM API.

```
/* Modulus Divide (+99) by (+4) */
uidiv_return result;
result = pROMDiv-> uidivmod (+99, 4);
/* result.div contains (+24) */
/* result.mod contains (+3) */
```

32.1 How to read this chapter

The debug functionality is identical for all LPC84x parts.

32.2 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.
- Micro Trace Buffer (MTB) supported.

32.3 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.

Support for boundary scan and Micro Trace Buffer is available. In order to use the micro-trace buffer for debugging, enable the MTB clock in the SYSAHBCLKCTRL register (Table 147).

Only RAM0 can be used as trace buffer by MTB, that means the maximum trace buffer size is 4 KB.

32.4 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 10.3.2 to enable the analog comparator inputs and the reference voltage input.

Table 487. 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 196
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 LPC84x. This pin is pulled up internally.	PINENABLE0	Table 196

The boundary scan mode and the pins needed are selected by hardware (see Section 32.5.3). There is no access to the boundary scan pins through the switch matrix.

Remark: If SWD is disabled in the FAIM configuration, these pins will be configured as GPIO pins on POR.

Table 488. 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.

32.5 Functional description

32.5.1 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.

32.5.2 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 67.   Connecting the SWD pins to a standard SWD connector
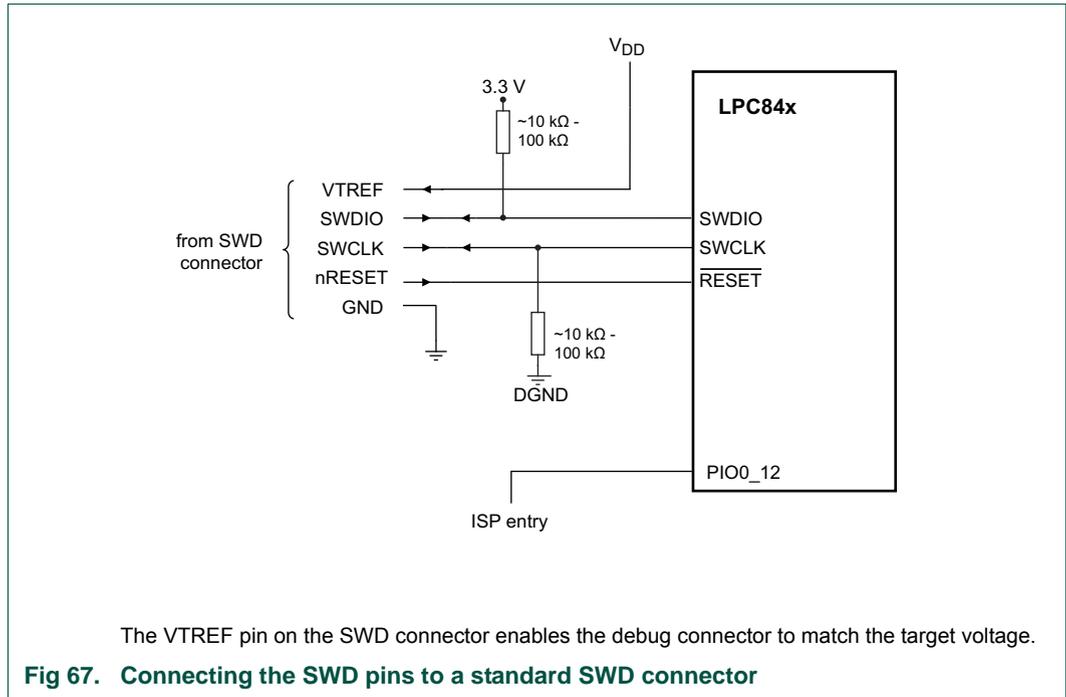
32.5.3 Boundary scan

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.

32.5.4 Micro Trace Buffer (MTB)

The MTB registers are located at memory address 0x5000 C000 and are described in [Ref. 4](#). The EXTTRACE register in the SYSCON block (see [Section 8.6.35](#)) starts and stops tracing in conjunction with the TSTARTEN and TSTOPEN bits in the MTB MASTER register. The trace is stored in the local SRAM starting at address 0x1000 0000. The trace memory location is configured in the MTB POSITION register.

Remark: The MTB BASE register is not implemented. Reading the BASE register returns 0x0 independently of the SRAM memory area configured for trace.

UM11029

User manual **Rev. 1.6 — 8 December 2017** **508 of 530**

33.1 Abbreviations

Table 489. 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
GPIO	General Purpose Input/Output
JTAG	Joint Test Action Group
PLL	Phase-Locked Loop
RC	Resistor-Capacitor
SPI	Serial Peripheral Interface
SSI	Serial Synchronous Interface
SSP	Synchronous Serial Port
TAP	Test Access Port
UART	Universal Asynchronous Receiver/Transmitter
USART	Universal Synchronous Asynchronous Receiver/Transmitter

33.2 References

[1] **LPC84x —** LPC84x Data sheet

[2] **ES_LPC84x —** LPC84x Errata sheet

[3] **DDI0484B_cortex_m0p_r0p0_trm —** ARM Cortex-M0+ Technical Reference Manual

[4] **DDI0486A —** ARM technical reference manual

[5] **AN11538 —** AN11538 application note and code bundle (SCT cookbook)

[6] ARMv6-M Architecture Reference Manual

33.3 Legal information

33.3.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

33.3.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or

malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

33.3.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

33.4 Tables

UM11029

User manual **Rev. 1.6 — 8 December 2017** **512 of 530**

UM11029

User manual   **Rev. 1.6 — 8 December 2017**   **517 of 530**

33.5 Figures

33.6 Contents

Chapter 6: LPC84x Flash signature generator

Chapter 7: LPC84x Nested Vectored Interrupt Controller (NVIC)

Chapter 8: LPC84x System configuration (SYSCON)

Chapter 9: LPC84x FRO API ROM routine

Chapter 10: LPC84x Switch matrix (SWM)

Chapter 11: LPC84x I/O Configuration (IOCON)

Chapter 12: LPC84x General Purpose I/O (GPIO)

Chapter 13: LPC84x Pin interrupts/pattern match engine

Chapter 26: LPC84x Capacitive Touch

Chapter 27: LPC84x 12-bit Analog-to-Digital Converter (ADC)

Chapter 28: LPC84x Digital-to-Analog Converter (DAC)