



# UM11045

## LPC802 User manual

Rev. 1.5 — 22 March 2021

User manual

### Document information

Info	Content
Keywords	LPC802, LPC802 UM, LPC802 user manual
Abstract	LPC802 User manual



## Revision history

Rev	Date	Description
1.5	20210322	LPC802 User manual.
Modifications		<ul style="list-style-type: none"> <li>Updated <a href="#">Table 108 "PIO0_11 register (PIO0_11, address 0x4004 401C) bit description"</a></li> <li>Updated <a href="#">Table 97 "Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description"</a></li> <li>Updated <a href="#">Table 89 "Pin assign register 0 (PINASSIGN0, address 0x4000 C000) bit description"</a> and <a href="#">Table 90 "Pin assign register 1 (PINASSIGN1, address 0x4000 C004) bit description"</a></li> <li>Update <a href="#">Figure 45 "ADC clocking"</a></li> <li>Updated <a href="#">Table 94 "Pin assign register 5 (PINASSIGN5, address 0x4000 C014) bit description"</a></li> <li>Updated <a href="#">Table 235 "Module Configuration register (MODCFG, offset 0xF0) bit description"</a></li> <li>Updated <a href="#">Section 4.6 "IAP commands"</a></li> <li>Added LPC802UK part number in <a href="#">Table 84 "LPC802 device identification numbers"</a></li> </ul>
1.4	20180427	LPC802 User manual.
		<ul style="list-style-type: none"> <li>Added LPC802UK part.</li> <li>Updated <a href="#">Table 82 "Power configuration register (PDRUNCFG, address 0x4004 8238) bit description"</a>: Reset value of BOD_PD is 0.</li> </ul>
1.3	20180323	LPC802 User manual.

## Revision history ...continued

Rev	Date	Description
Modifications:		<ul style="list-style-type: none"> <li>Updated Table 12 "USART ISP Write to RAM command", Table 13 "USART ISP Read Memory command", Table 15 "USART ISP Copy command", Table 16 "USART ISP Go command", Table 23 "USART ISP Compare command", and Table 25 "USART ISP Read CRC checksum command": added on-chip RAM.</li> <li>Removed IBARO register from Chapter 5 "LPC802 Nested Vectored Interrupt Controller (NVIC)".</li> <li>Updated Table 175 "SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI) bit description", Table 176 "SPI Interrupt Enable clear register (INTENCLR, addresses 0x4005 8010 (SPI) bit description", and Table 182 "SPI Interrupt Status register (INTSTAT, addresses 0x4005 8028 (SPI) bit description".</li> <li>Added a remark to Table 97 "Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description" and Section 9.4.5 "Analog mode".</li> <li>Updated Section 12.7.7 "Deep power-down mode": added text: Five general-purpose registers are available to store information during deep power-down mode.</li> <li>Updated Section 21.3.1 "Perform a single ADC conversion using a software trigger": Changed the result of the conversion: 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 <math>(4095 \times (VIN - VREFN)) / (VREFP - VREFN)</math>.</li> <li>Updated Section 21.6.3 "A/D Conversion Sequence B Control Register" remark to: Set the BURST and SEQB_ENA bits at the same time. Fixed title of Table 250 "A/D Conversion Sequence B Control Register (SEQB_CTRL, address 0x4001 C00C) bit description".</li> <li>Updated Table 247 "Register overview: ADC (base address 0x4001 C000)": changed access for SEQA_GDAT and SEQB_GDAT to RO.</li> <li>Updated Table 249 "A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description" Removed TSAMP.</li> <li>Updated Table 237 "Global interrupt flag register (IRQ_FLAG, address 0x4000 40F8) bit description": Bits 31:2 are reserved.</li> <li>Updated Section 17.2 "Features" The watchdog reload or watchdog feed sequence can optionally be protected so that it can only be performed after the "warning interrupt" time is reached.</li> <li>Updated Section 17.5 "General description" "General description". Added a bullet: Set the Watchdog timer update mode (WDPROTECT) in the MOD register after a delay of three WDCLK clock cycles.</li> <li>Updated Section 17.5.3 "Using the WWDT lock features" and Section 17.5.3.2 "Changing the WWDT reload value".</li> <li>Updated Table 220 "Watchdog mode register (MOD, 0x4000 0000) bit description". New text for enumerated values of bit 4, WDPROTECT.</li> <li>Updated Section 17.6.2 "Watchdog Timer Constant register". New text: If the WDPROTECT bit in WDMOD = 1, an attempt to perform the watchdog reload or watchdog feed sequence before the watchdog timer is below the values of WDWARNINT and WDWINDOW will cause a watchdog feed error and set the WDTOF flag.</li> <li>Updated Section 19.6.6 "Idle channel register".</li> <li>Updated Table 230 "Register overview: MRT (base address 0x4000 4000)": changed access of Timer0 and Timer1.</li> </ul>

## Contact information

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

## Revision history ...continued

Rev	Date	Description
1.2	20180213	LPC802 User manual.
Modifications:	<ul style="list-style-type: none"><li>• Added Level Shifter functionality to Section 1.2 “Features”.</li><li>• Added Section 8.4.3 “Level Shifter function”.</li><li>• Updated Table 88 “Register overview: Switch matrix (base address 0x4000 C000)”</li><li>• Updated Table 95 “Pin assign register 6 (PINASSIGN6, address 0x4000 C018) bit description”. Added LVLSHFT.</li><li>• Added Table 96 “Pin assign register 7 (PINASSIGN7, address 0x4000 C01C) bit description”.</li><li>• Added Table 235 “Module Configuration register (MODCFG, offset 0xF0) bit description”.</li></ul>	
1.1	20180125	LPC802 User manual.
Modifications:	<ul style="list-style-type: none"><li>• Updated Figure 1 “LPC802 block diagram”.</li><li>• Updated Table 38 “Connection of interrupt sources to the NVIC”.</li><li>• Updated Figure 6 “LPC802 clock generation”.</li><li>• Updated Table 156 “Peripheral configuration in reduced power modes”: Flash is standby in deep-sleep mode.</li></ul>	
1	20171201	Initial revision. LPC802 User manual.

### 1.1 Introduction

---

The LPC802 are an Arm® Cortex® -M0+ based, low-cost 32-bit MCU family operating at CPU frequencies of up to 15 MHz. The LPC802 support up to 16 KB of flash memory and 2 KB of SRAM.

The peripheral complement of the LPC802 includes one I2C-bus interface, up to two USARTs, one SPI interface, one multi-rate timer, self-wake-up timer, one general purpose 32-bit counter/timer, one 12-bit ADC, one analog comparator, function-configurable I/O ports through a switch matrix, and up to 17 general-purpose I/O pins.

**Remark:** For additional documentation, see [Section 26.2 “References”](#).

### 1.2 Features

---

- System:
  - Arm Cortex-M0+ processor (revision r0p1), running at frequencies of up to 15 MHz with single-cycle multiplier and fast single-cycle I/O port.
  - Arm Cortex-M0+ built-in Nested Vectored Interrupt Controller (NVIC).
  - System tick timer.
  - AHB multilayer matrix.
  - Serial Wire Debug (SWD) with four break points and two watch points. JTAG boundary scan (BSDL) supported.
- Memory:
  - 16 KB on-chip flash programming memory.
  - Code Read Protection (CRP).
  - Up to 2 KB SRAM.
- Dual I/O power (LPC802M011JDH20):
  - Independent supplies on each package side permitting level-shifting signals from one off-chip voltage domain to another and/or interfacing directly to off-chip peripherals operating at different supply levels.
  - The switch matrix provides level shifter functionality to allow up to two selected signals to be routed from user-selected pins in one voltage domain to selected pins in the alternate domain. This feature can also be used on a single supply device if voltage level shifting is not required.
- ROM API support:
  - Bootloader.
  - Supports Flash In-Application Programming (IAP).
  - Supports In-System Programming (ISP) through USART.
  - On-chip ROM APIs for integer divide.
  - Free Running Oscillator (FRO) API.

- Digital peripherals:
  - High-speed GPIO interface connected to the Arm Cortex-M0+ I/O bus with up to 17 General-Purpose I/O (GPIO) pins with configurable pull-up/pull-down resistors, programmable open-drain mode, and input inverter. GPIO direction control supports independent set/clear/toggle of individual bits.
  - High-current source output driver (20 mA) on three pins.
  - Switch matrix for flexible configuration of each I/O pin function.
  - CRC engine.
- Timers:
  - One 32-bit general purpose counter/timer, with four match outputs and three capture inputs. Supports PWM mode, and external count.
  - Two 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.
  - 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 480 Ksamples/s. The ADC supports two independent conversion sequences.
  - Comparator with four input pins and external or internal reference voltage.
- Serial peripherals:
  - Two USART interfaces with pin functions assigned through the switch matrix and one fractional baud rate generators.
  - One SPI controller with pin functions assigned through the switch matrix.
  - One I2C-bus interface. Supports Standard mode and Fast mode.
- Clock generation:
  - Free Running Oscillator (FRO). This oscillator provides a selectable 9 MHz, 12 MHz and 15 MHz outputs that can be used as a system clock. The FRO is trimmed to  $\pm 1$  % accuracy over the entire voltage and temperature range of 0 C to 70 C.
  - 1 MHz low power oscillator can be used as a clock source.
  - Clock output function with divider that can reflect all internal clock sources.
- Power control:
  - Reduced power modes: sleep mode, deep-sleep mode, power-down mode, and deep power-down mode.
  - Wake-up from deep-sleep and power-down modes on activity on USART, SPI, and I2C peripherals.
  - Wake-up from deep power-down mode on multiple pins
  - Timer-controlled self wake-up from sleep, deep-sleep, and power-down modes.
  - Power-On Reset (POR).
  - Brownout detect (BOD).
- Unique device serial number for identification.

- Single or dual power supplies (1.71 V to 3.6 V).
- Operating temperature range -40 °C to +105 °C.
- Available in TSSOP16, TSSOP20, and HVQFN33 packages

## 1.3 Ordering options

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
LPC802M001JDH16	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
LPC802M001JDH20	TSSOP20	plastic thin shrink small outline package; 20 leads; body width 4.4 mm	SOT360-1
LPC802M011JDH20	TSSOP20	plastic thin shrink small outline package; 20 leads; body width 4.4 mm	SOT360-1
LPC802M001JHI33	HVQFN33	HVQFN: plastic thermal enhanced very thin quad flat package; no leads; 33 terminals; body 5 × 5 × 0.85 mm	n/a
LPC802UK	WLCSP16	wafer level chip-size package; 16 (4x4) bumps; 1.86 x 1.86 x 0.3 mm	SOT1393-2

Table 2. Ordering options

Type number	Flash/KB	SRAM/ KB	USART	I <sup>2</sup> C	SPI	GPIO	Dual I/O power supply	Package
LPC802M001JDH16	16	2	2	1	1	13	-	TSSOP16
LPC802M001JDH20	16	2	2	1	1	17	-	TSSOP20
LPC802M011JDH20	16	2	2	1	1	16	yes	TSSOP20
LPC802M001JHI33	16	2	2	1	1	25	-	HVQFN33
LPC802UK	16	2	2	1	1	13	-	WLCSP16



## 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 15 MHz. Integrated in the core are the NVIC and Serial Wire Debug with four breakpoints and two watch points. The Arm Cortex-M0+ core supports a single-cycle I/O enabled port (IOP) for fast GPIO access at address 0xA000 0000. The Arm Cortex M0+ core version is r0p1.

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

## 1.5 Block diagram

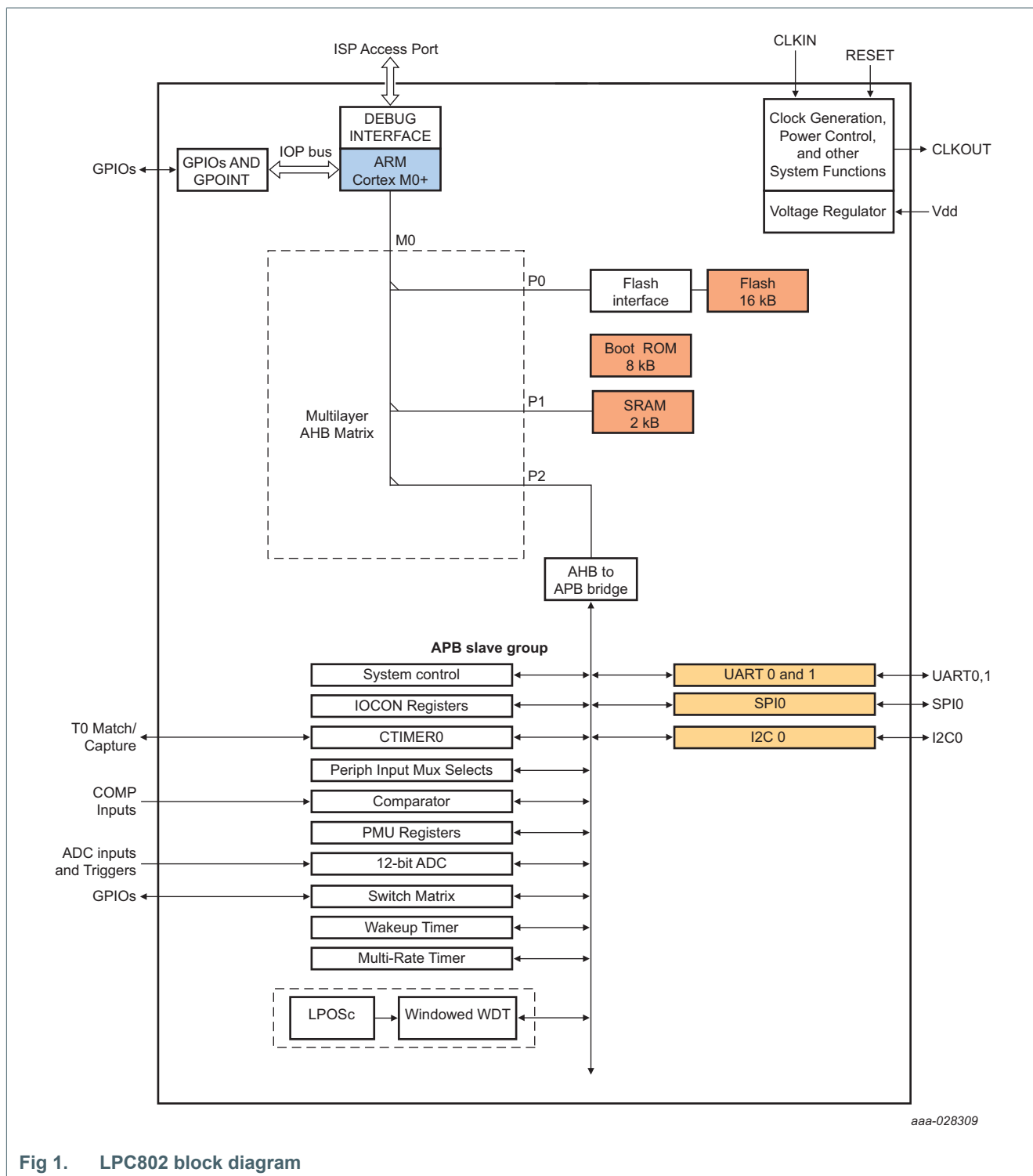


Fig 1. LPC802 block diagram

### 2.1 How to read this chapter

---

The memory mapping is identical for all LPC802 parts.

### 2.2 General description

---

The LPC802 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

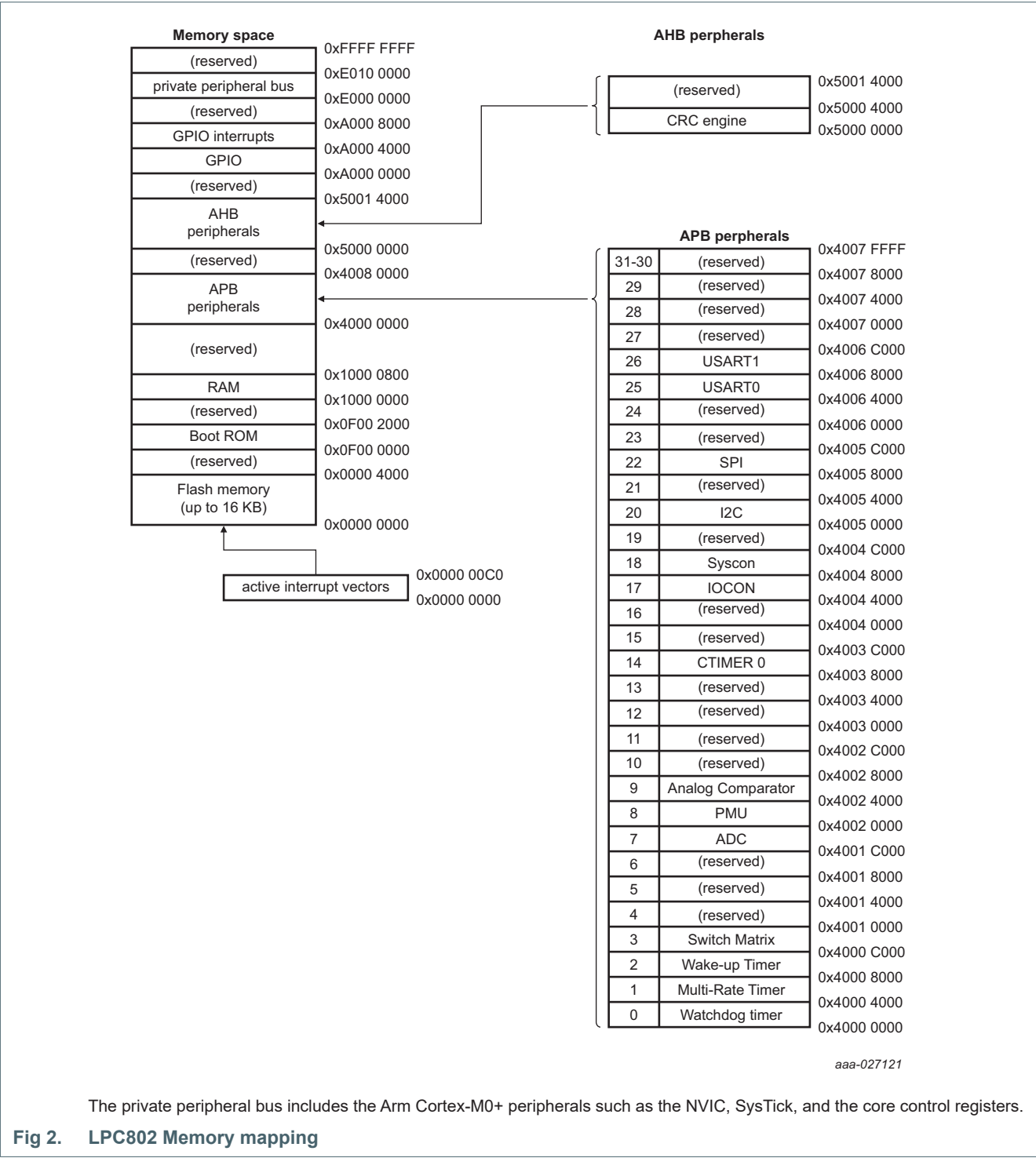


Fig 2. LPC802 Memory mapping

### 3.1 How to read this chapter

The bootloader is identical for all parts.

### 3.2 Features

- 8 KB on-chip boot ROM
- Contains the bootloader with In-System Programming (ISP) facility over the peripheral communication (UART) and the following API:
  - In-Application Programming (IAP) of flash memory.
  - Integer divide routines.
  - FRO frequency select API.

### 3.3 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 configuration
USART ISP (single supply device)	PIO0_4 is UART0 TX. PIO0_0 is UART0 RX.
USART ISP (dual supply device)	PIO0_9 is UART0 TX. PIO0_8 is UART0 RX.

### 3.4 General description

#### 3.4.1 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 from the USART bus.

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 the USART interface. Otherwise, the bootloader checks if there is valid user code in flash. If the valid user code is not found, the bootloader enters the ISP mode.

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

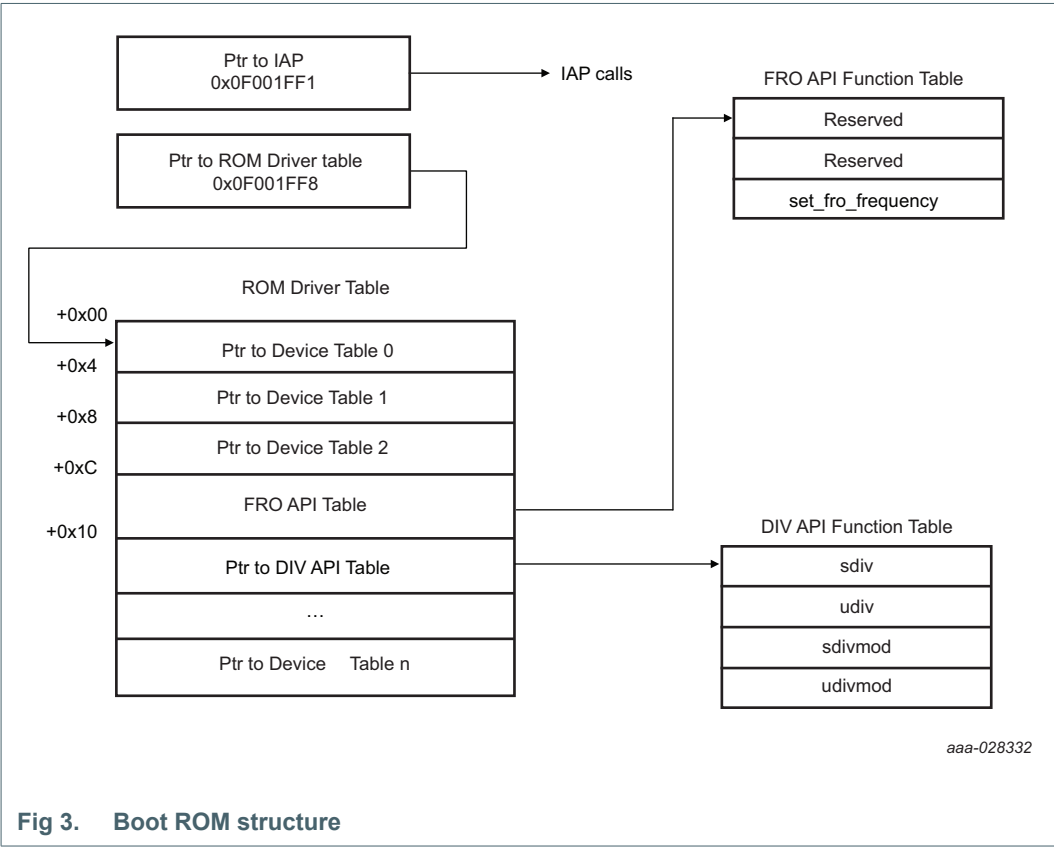
See [Chapter 4 “LPC802 ISP and IAP”](#) for more details.

3.4.2 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.
- On-chip ROM APIs for integer divide.
- FRO frequency select API.

The structure of the boot ROM APIs is shown in [Figure 3](#).



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 PWRD_API_T *pPWRD; /*!< Power API function table base address */
    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**

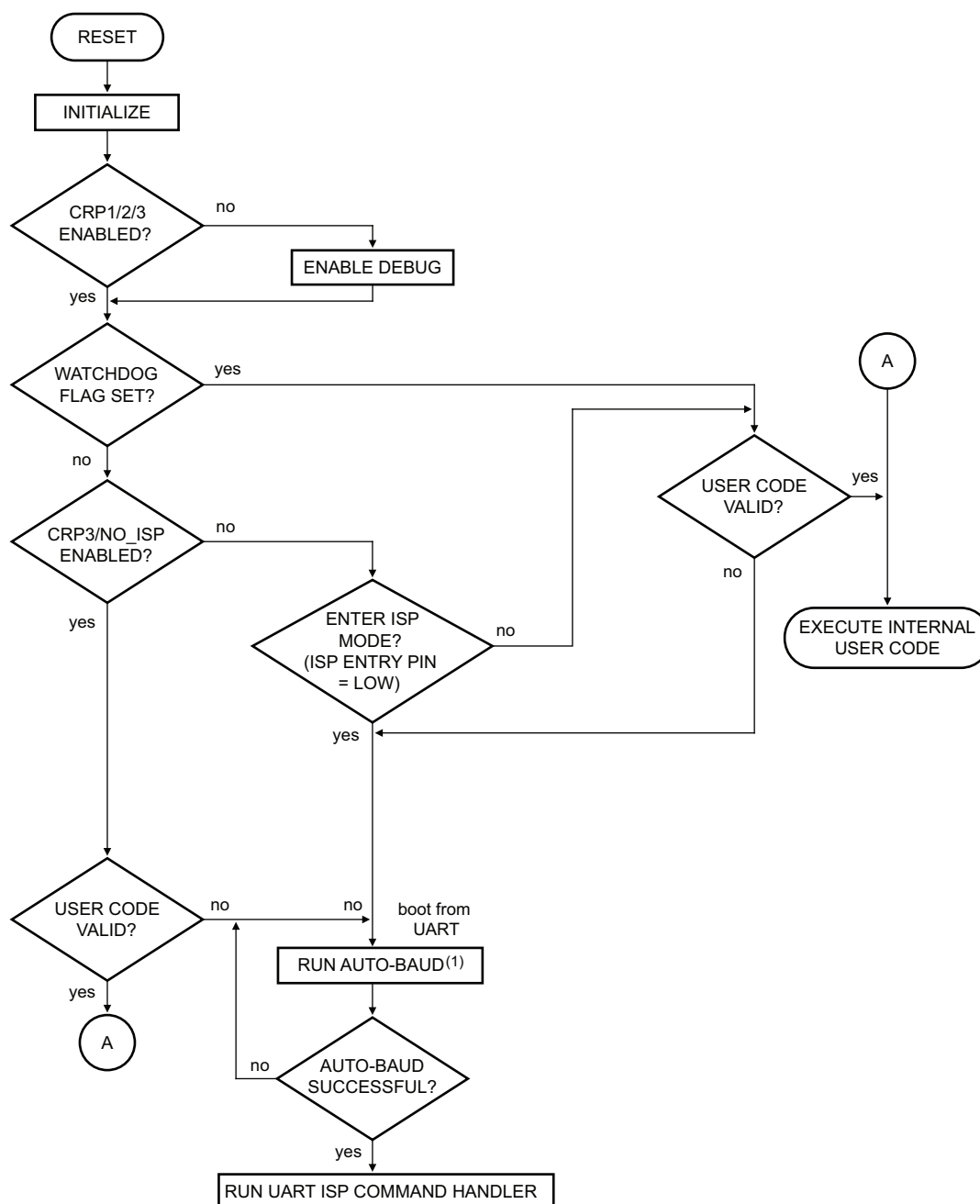
API	Description	Reference
Flash IAP	Flash In-Application programming	<a href="#">Table 27</a>
Integer divider API	32-bit integer divide routines	<a href="#">Table 270</a>
FRO API	Free Running Oscillator configuration Section	<a href="#">Section 7.4</a>

## 3.5 Functional description

### 3.5.1 Memory map after any reset

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

## 3.5.2 Boot process



(1) The boot-code is implementing auto-baud in software.

**Fig 4. Boot process flowchart**



### 4.1 How to read this chapter

---

All LPC802 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. In-Application Programming allows application software to do the same kinds of operations.

### 4.2 Features

---

- In-System Programming: In-System programming (ISP) is programming or reprogramming the on-chip flash memory, using the boot loader software and USART 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.

### 4.3 General description

---

#### 4.3.1 Boot loader

For the boot loader operation and ISP pin, see [Chapter 3 “LPC802 Boot Process”](#).

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

#### 4.3.2 Memory map after any reset

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

#### 4.3.3 Flash content protection mechanism

The LPC802 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.

#### 4.3.4 Criteria for Valid User Code

The reserved CPU exception vector location 7 (offset 0x0000 001C in the vector table) should contain the 2's complement of the checksum 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 enter USART ISP mode automatically. When the table entries 0 through 7 are 0, even the checksum is 0, the user code is still invalid.

#### 4.3.5 Flash partitions

Some IAP and ISP commands operate on virtual sectors and specify virtual sector numbers. In addition, a page erase command is available. The size of a virtual sector is 1 KB and the size of a page is 64 Byte. One virtual sector contains 16 pages. Page 254 and page 255 in sector 15 are not available for user code because of the boot block.

Table 5. LPC802 flash configuration

Virtual sector number	Virtual sector size [KB]	Page number	Address range	16 KB flash
0	1	0 - 15	0x0000 0000 - 0x0000 03FF	yes
1	1	16 - 31	0x0000 0400 - 0x0000 07FF	yes
2	1	32 - 47	0x0000 0800 - 0x0000 0BFF	yes
3	1	48 - 63	0x0000 0C00 - 0x0000 0FFF	yes
4	1	64 - 79	0x0000 1000 - 0x0000 13FF	yes
5	1	80 - 95	0x0000 1400 - 0x0000 17FF	yes
6	1	96 - 111	0x0000 1800 - 0x0000 1BFF	yes
7	1	112 - 127	0x0000 1C00 - 0x0000 1FFF	yes
8	1	128 - 143	0x0000 2000 - 0x0000 23FF	yes
9	1	144 - 159	0x0000 2400 - 0x0000 27FF	yes
10	1	160 - 175	0x0000 2800 - 0x0000 2BFF	yes
11	1	176 - 191	0x0000 2C00 - 0x0000 2FFF	yes
12	1	192 - 207	0x0000 3000 - 0x0000 33FF	yes
13	1	208 - 223	0x0000 3400 - 0x0000 37FF	yes
14	1	224 - 239	0x0000 3800 - 0x0000 3BFF	yes
15	1	240 - 255	0x0000 3C00 - 0x0000 3FFF	yes

### 4.3.6 Code Read Protection (CRP)

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

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

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

**Table 6. 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	<p>Access to chip via the SWD pins is disabled. This mode allows partial flash update using the following USART ISP commands and restrictions:</p> <ul style="list-style-type: none"> <li>• Write to RAM command cannot access RAM below 0x1000 0380.</li> <li>• Copy RAM to flash command cannot write to Sector 0.</li> <li>• Erase command can erase Sector 0 only when all sectors are selected for erase.</li> <li>• Compare command is disabled.</li> <li>• Read Memory command is disabled.</li> </ul> <p>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.</p>
CRP2	0x8765 4321	<p>Access to chip via the SWD pins is disabled. The following ISP commands are disabled:</p> <ul style="list-style-type: none"> <li>• Read Memory</li> <li>• Write to RAM</li> <li>• Go</li> <li>• Copy RAM to flash</li> <li>• Compare</li> </ul> <p>When CRP2 is enabled the ISP erase command only allows erasure of all user sectors.</p>
CRP3	0x4321 8765	<p>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.</p> <p>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.</p> <p><b>Caution: If CRP3 is selected, no future factory testing can be performed on the device.</b></p>

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

#### 4.3.6.1 ISP entry protection

In addition to the three CRP modes, the user can prevent the sampling of the pin for entering ISP mode and thereby release the pin for other applications. This is called the NO\_ISP mode. The NO\_ISP mode can be entered by programming the pattern 0x4E69 7370 at location 0x0000 02FC.

The NO\_ISP mode is identical to the CRP3 mode except for SWD access, which is allowed in NO\_ISP mode but disabled in CRP3 mode. The NO\_ISP mode does not offer any code protection.

#### 4.3.6.2 ISP entry configuration and detection

The LPC802 USART ISP mode allows programming and reprogramming of the internal FLASH via a set of commands on the USART bus.

### 4.3.7 ISP interrupt and SRAM use

#### 4.3.7.1 Interrupts during IAP

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

#### 4.3.7.2 RAM used by ISP command handlers

The stack of UART ISP commands is located at address 0x1000 03A8. The maximum stack usage is 640 bytes (0x280) and grows downwards.

The ISP flash programming commands use the top 4 bytes of on-chip RAM.

#### 4.3.7.3 RAM used by IAP command handlers

The IAP flash programming commands use the top 4 bytes of on-chip RAM.

## 4.4 USART ISP communication protocol

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

### 4.4.1 USART ISP initialization

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

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

The auto-baud routine looks at the received characters to verify synchronization. If synchronization is verified then "OK<CR><LF>" string is sent to the host. The host should respond by sending the crystal frequency (in kHz) at which the part is running. The response is required for backward compatibility of the boot loader code and is ignored. "OK<CR><LF>" string is sent to the host after receiving the crystal frequency. If synchronization is not verified then the auto-baud routine waits again for a synchronization character. In USART ISP mode, the part is clocked by the FRO 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 4.5 "USART ISP commands"](#).

### 4.4.2 USART ISP command format

"Command Parameter\_0 Parameter\_1 ... Parameter\_n<CR><LF>" "Data" (Data only for Write commands). By default, all commands (all characters) sent from the host are echoed. The echo command can be issued to disable this.

### 4.4.3 USART ISP response format

"Return\_Code<CR><LF>Response\_0<CR><LF>Response\_1<CR><LF> ... Response\_n<CR><LF>" "Data" (Data only for Read commands).

### 4.4.4 USART ISP data format

The data stream is in plain binary format.

## 4.5 USART ISP commands

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

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

**Table 7. USART ISP command summary**

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

[Table 8](#) lists the supported USART ISP commands for each CRP level.

**Table 8. ISP commands allowed for different CRP levels**

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

Table 8. ISP commands allowed for different CRP levels

ISP command	CRP1	CRP2	CRP3 (no entry in ISP mode allowed)
Erase page(s)	yes; page 0 can only be erased when all pages are erased (not recommended, use Erase Sector).	yes; all pages only	n/a
Blank check sectors	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

### 4.5.1 Unlock

Table 9. USART ISP Unlock command

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

### 4.5.2 Set Baud Rate

Table 10. USART ISP Set Baud Rate command

Command	B
Input	Baud Rate: 9600   19200   38400   57600   115200   230400   460800 Stop bit: 1   2 <b>Remark:</b> In asynchronous USART mode, the USART clock U_PCLK = FRGCLKDIV/ (1+(MULT/DIV)), thus, the U_PCLK is 12MHz. Then, the maximum baud rate = U_PCLK/16 x (BRGVAL+1), where BRGVAL is the value of the BRG register.
Return Code	CMD_SUCCESS   INVALID_BAUD_RATE   INVALID_STOP_BIT   PARAM_ERROR
Description	This command is used to change the baud rate. The new baud rate is effective after the command handler sends the CMD_SUCCESS return code.
Example	"B 57600 1<CR><LF>" sets the serial port to baud rate 57600 bps and 1 stop bit.

### 4.5.3 Echo

Table 11. 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.

### 4.5.4 Write to RAM

The host should send the plain binary code after receiving the CMD\_SUCCESS return code. This ISP command handler responds with "OK<CR><LF>" when the transfer has finished.

Table 12. USART ISP Write to RAM command

Command	W
Input	<b>Start Address:</b> On-chip RAM address where data bytes are to be written. This address should be a word boundary. <b>Number of Bytes:</b> 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 on-chip RAM. This command is blocked when code read protection levels 2 or 3 are enabled. Writing to addresses below 0x1000 03A8 is disabled for CRP1.
Example	"W 268437504 4<CR><LF>" writes 4 bytes of data to address 0x1000 0800.

### 4.5.5 Read Memory

Reads the plain binary code of the data stream, followed by the CMD\_SUCCESS return code.

Table 13. USART ISP Read Memory command

Command	R
Input	<b>Start Address:</b> Address from where data bytes are to be read. This address should be a word boundary. <b>Number of Bytes:</b> 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 on-chip RAM or flash memory. This command is blocked when code read protection is enabled.
Example	"R 268437504 4<CR><LF>" reads 4 bytes of data from address 0x1000 0800.



### 4.5.6 Prepare sectors for write operation

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

Table 14. USART ISP Prepare sectors for write operation command

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

### 4.5.7 Copy RAM to flash

When writing to the flash, the following 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 4.3.3](#).
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 15. USART ISP Copy command

Command	C
Input	<b>Flash Address(DST):</b> Destination flash address where data bytes are to be written. The destination address should be a 64 byte boundary. <b>RAM Address(SRC):</b> Source on-chip RAM address from where data bytes are to be read. <b>Number of Bytes:</b> 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 <a href="#">Section 4.3.3</a> for the number of bytes that can be written.
Example	"C 0 268437504 512<CR><LF>" copies 512 bytes from the RAM address 0x1000 0800 to the flash address 0.

### 4.5.8 Go

Table 16. USART ISP Go command

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

### 4.5.9 Erase sectors

**Remark:** The blank (erased) state of a page flash memory is a logic 0.

Table 17. USART ISP Erase sector command

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

### 4.5.10 Erase pages

**Remark:** The blank (erased) state of a page flash memory is a logic 0. In sector 15, page 254 and page 255 cannot be erased since boot block resides in these pages. Page 254 and page 255 cannot pass as a parameter.

Table 18. USART ISP Erase page command

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

### 4.5.11 Blank check sectors

**Remark:** The blank (erased) state of a page flash memory is a logic 0.

Table 19. USART ISP Blank check sector command

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

### 4.5.12 Read Part Identification number

Table 20. USART ISP Read Part Identification command

Command	J
Input	None.
Return Code	CMD_SUCCESS followed by part identification number (see <a href="#">Table 21</a> ).
Description	This command is used to read the part identification number.

Table 21. LPC802 device identification numbers

Part number	Part ID
LPC802M001JDH16	0x00008023
LPC802M001JDH20	0x00008021
LPC802M011JDH20	0x00008022
LPC802M001JHI33	0x00008024

### 4.5.13 Read Boot code version number

Table 22. USART ISP Read Boot Code version number command

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

### 4.5.14 Compare

Table 23. USART ISP Compare command

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

### 4.5.15 ReadUID

Table 24. USART ReadUID command

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

### 4.5.16 Read CRC checksum

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

The checksum is calculated as follows:

CRC-32 polynomial:  $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Seed Value: 0xFFFF FFFF

Table 25. USART ISP Read CRC checksum command

Command	S
Input	<b>Address:</b> The data are read from this address for CRC checksum calculation. This address must be on a word boundary. <b>Number of Bytes:</b> 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 on-chip RAM or flash memory. This command is blocked when code read protection is enabled.
Example	"S 33587200 4<CR><LF>" reads the CRC checksum for 4 bytes of data from address 0x0200 8000. If checksum value is 0xCBF43926, then the host will receive: "3421780262 <CR><LF>"

## 4.5.17 ISP/IAP Error codes

Table 26. ISP/IAP Error codes

Return Code	Error code	Description
0x0	CMD_SUCCESS	Command is executed successfully. Sent by ISP handler only when command given by the host has been completely and successfully executed.
0x1	INVALID_COMMAND	Invalid command.
0x2	SRC_ADDR_ERROR	Source address is not on word boundary.
0x3	DST_ADDR_ERROR	Destination address is not on a correct boundary.
0x4	SRC_ADDR_NOT_MAPPED	Source address is not mapped in the memory map. Count value is taken into consideration where applicable.
0x5	DST_ADDR_NOT_MAPPED	Destination address is not mapped in the memory map. Count value is taken into consideration where applicable.
0x6	COUNT_ERROR	Byte count is not multiple of 4 or is not a permitted value.
0x7	INVALID_SECTOR/INVALID_PAGE	Sector/page number is invalid or end sector number is greater than start sector number.
0x8	SECTOR_NOT_BLANK	Sector is not blank.
0x9	SECTOR_NOT_PREPARED_FOR_WRITE_OPERATION	Command to prepare sector for write operation was not executed.
0xA	COMPARE_ERROR	Source and destination data not equal.
0xB	BUSY	Flash programming hardware interface is busy.
0xC	PARAM_ERROR	Insufficient number of parameters or invalid parameter.
0xD	ADDR_ERROR	Address is not on word boundary.
0xE	ADDR_NOT_MAPPED	Address is not mapped in the memory map. Count value is taken into consideration where applicable.
0xF	CMD_LOCKED	Command is locked.
0x10	INVALID_CODE	Unlock code is invalid.
0x11	INVALID_BAUD_RATE	Invalid baud rate setting.
0x12	INVALID_STOP_BIT	Invalid stop bit setting.
0x13	CODE_READ_PROTECTION_ENABLED	Code read protection enabled.
0x14	-	Reserved.
0x15	USER_CODE_CHECKSUM	User code checksum is invalid.
0x16	-	Reserved.
0x17	EFRO_NO_POWER	FRO not turned on in the PDRUNCFG register.
0x18	FLASH_NO_POWER	Flash not turned on in the PDRUNCFG register.
0x19	-	Reserved.
0x20	FLASH_ERASE_PROGRAM	Flash erase/program not successful.
0x21	INVALID_PAGE	Page is invalid.
0x1A	-	Reserved.
0x1B	FLASH_NO_CLOCK	Flash clock disabled in the AHBCLKCTRL register.
0x1C	REINVOKE_ISP_CONFIG	Reinvoke ISP not successful.
0x1D	NO_VALID_IMAGE	Invalid image

## 4.6 IAP commands

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

The number of parameters and results vary according to the IAP command. The maximum number of parameters is 5, passed to the "Copy RAM to FLASH" command. The maximum number of results is 5, returned by the "ReadUID" command. The command handler sends the status code INVALID\_COMMAND when an undefined command is received. The IAP routine resides at location 0x0F001FF0 and it is thumb code, therefore called as 0x0F001FF1 by the Cortex-M0+ to ensure Thumb operation.

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

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

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

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

or

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

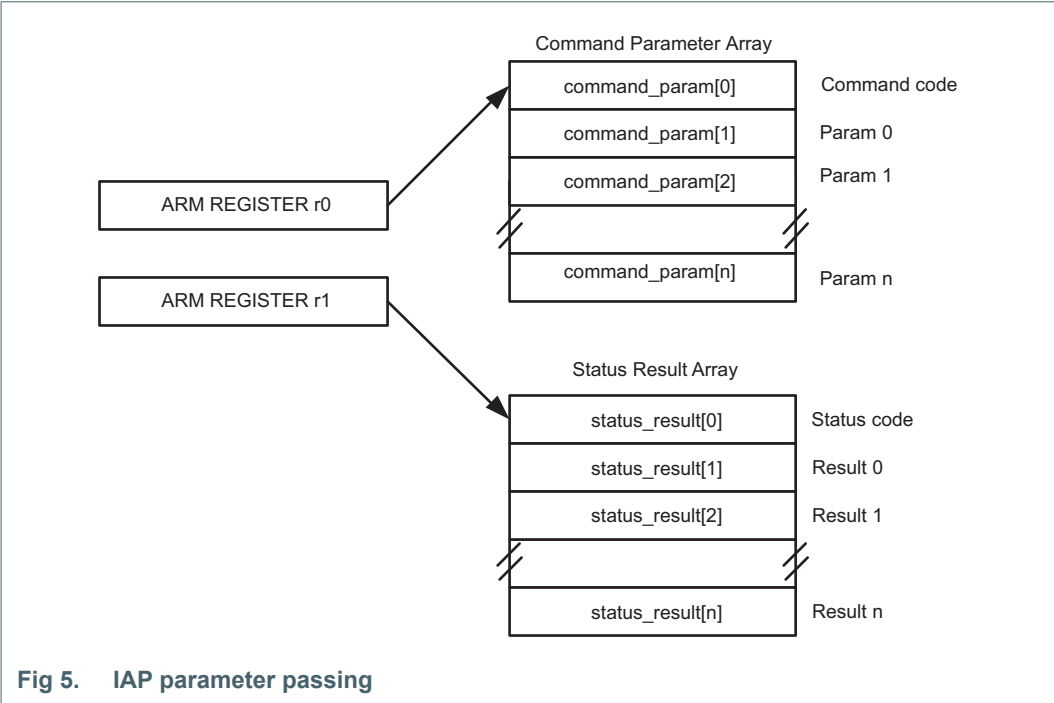
Up to 4 parameters can be passed in the r0, r1, r2 and r3 registers respectively (see the *Arm Thumb Procedure Call Standard SWS ESPC 0002 A-05*). Additional parameters are passed on the stack. Up to 4 parameters can be returned in the r0, r1, r2 and r3 registers respectively. Additional parameters are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

The flash memory is not accessible during a write or erase operation.



Table 27. IAP Command Summary

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



### 4.6.1 Prepare sector(s) for write operation

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

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

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

### 4.6.2 Copy RAM to flash

See [Section 4.5.7](#) for limitations on the write-to-flash process.

Table 29. IAP Copy RAM to flash command

Command	Copy RAM to flash
Input	<b>Command code: 51 (decimal)</b> <b>Param0(DST):</b> Destination flash address where data bytes are to be written. This address should be a 64 byte boundary. <b>Param1(SRC):</b> Source RAM address from which data bytes are to be read. This address should be a word boundary. <b>Param2:</b> Number of bytes to be written. Should be 64   128   256   512   1024. <b>Param3:</b> NULL.
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 <a href="#">Section 4.3.3</a> for the number of bytes that can be written.  <b>Remark:</b> 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. Param3 is overwritten by the fixed value of 12 MHz, which is the FRO reference clock used by the flash controller.

### 4.6.3 Erase Sector(s)

**Remark:** The blank (erased) state of a page flash memory is a logic 0.

**Table 30. IAP Erase Sector(s) command**

Command	Erase Sector(s)
Input	<b>Command code: 52 (decimal)</b> <b>Param0:</b> Start Sector Number <b>Param1:</b> End Sector Number (should be greater than or equal to start sector number). <b>Param2:</b> NULL.
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. <b>Remark:</b> 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. Param2 is overwritten by the fixed value of 12 MHz, which is the FRO reference clock used by the flash controller.

### 4.6.4 Blank check sector(s)

**Remark:** The blank (erased) state of a page flash memory is a logic 0.

**Table 31. IAP Blank check sector(s) command**

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

### 4.6.5 Read Part Identification number

**Table 32. IAP Read Part Identification command**

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

### 4.6.6 Read Boot code version number

Table 33. IAP Read Boot Code version number command

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

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

Table 34. IAP Compare command

Command	Compare
Input	<b>Command code: 56 (decimal)</b> <b>Param0(DST):</b> Starting flash or RAM address of data bytes to be compared; should be a word boundary. <b>Param1(SRC):</b> Starting flash or RAM address of data bytes to be compared; should be a word boundary. <b>Param2:</b> 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	<b>Result0:</b> Offset of the first mismatch if the status code is COMPARE_ERROR.
Description	This command is used to compare the memory contents at two locations.

### 4.6.8 Reinvoke ISP

Table 35. Reinvoke ISP

Command	Compare
Input	<b>Command code: 57 (decimal)</b> <b>Param0(mode):</b> ISP interface selection 1 - USART ISP
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.

### 4.6.9 ReadUID

Table 36. IAP ReadUID command

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

### 4.6.10 Erase page

**Remark:** The blank (erased) state of a page flash memory is a logic 0. In sector 15, page 254 and page 255 cannot be erased since boot block resides in these pages. Page 254 and page 255 cannot pass as a parameter.

Table 37. IAP Erase page command

Command	Erase page
Input	<b>Command code: 59 (decimal)</b> <b>Param0:</b> Start page number. <b>Param1:</b> End page number (should be greater than or equal to start page) <b>Param2:</b> NULL
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.  <b>Remark:</b> 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. In sector 15, page 254 and page 255 cannot be erased since boot block resides in these pages. Param2 is overwritten by the fixed value of 12 MHz, which is the FRO reference clock used by the flash controller.

### 4.6.11 IAP Error Codes

See [Table 26 "ISP/IAP Error codes"](#).

### 5.1 How to read this chapter

The NVIC is identical on all LPC802 parts.

### 5.2 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.

### 5.3 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.

#### 5.3.1 Interrupt sources

[Table 38](#) 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 38. Connection of interrupt sources to the NVIC**

Interrupt number	Name	Description	Flags
0	SPI0_IRQ	SPI0 interrupt	See <a href="#">Table 175 “SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI) bit description”</a> .
2:1		Reserved	-
3	UART0_IRQ	USART0 interrupt	See <a href="#">Table 161 “USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1)) bit description”</a>
4	UART1_IRQ	USART1 interrupt	Same as UART0_IRQ
7:5		Reserved	-

Table 38. Connection of interrupt sources to the NVIC

Interrupt number	Name	Description	Flags
8	I2C0_IRQ	I2C0 interrupt	See <a href="#">Table 191 “Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0) bit description”</a> .
9			
10	MRT_IRQ	Multi-rate timer interrupt	Global MRT interrupt. GFLAG0 GFLAG1 GFLAG2 GFLAG3
11	CMP_IRQ	Analog comparator interrupt	COMPEDGE - rising, falling, or both edges can set the bit.
12	WDT_IRQ	Windowed watchdog timer interrupt	WARNINT - watchdog warning interrupt
13	BOD_IRQ	BOD interrupts	BODINTVAL - BOD interrupt level
14	-	Reserved	-
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	-
22:20	-	Reserved	-
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

Table 38. Connection of interrupt sources to the NVIC

Interrupt number	Name	Description	Flags
29	PININT5_IRQ	Pin interrupt 5 or pattern match engine slice 5 interrupt.	PSTAT - pin interrupt status
30	PININT6_IRQ	Pin interrupt 6 or pattern match engine slice 6 interrupt.	PSTAT - pin interrupt status
31	PININT7_IRQ	Pin interrupt 7 or pattern match engine slice 7 interrupt.	PSTAT - pin interrupt status

### 5.3.2 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 38](#) as NMI using the NMISRC register in the SYSCON block ([Table 76](#)). 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.

### 5.3.3 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](#)).



## 5.4 Register description

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

**Table 39. 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	<a href="#">Table 40</a>
-	-	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	<a href="#">Table 41</a>
-	-	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	<a href="#">Table 42</a>
-	-	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	<a href="#">Table 43</a>
-	-	0x284	Reserved.	0	-
-	-	0x300	Reserved.	0	-
-	-	0x304	Reserved.	0	-
IPR0	RW	0x400	Interrupt Priority Registers 0. This register allows assigning a priority to each interrupt. This register contains the 2-bit priority fields for interrupts 0 to 3.	0	<a href="#">Table 44</a>
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	<a href="#">Table 45</a>
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	<a href="#">Table 46</a>
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	<a href="#">Table 47</a>
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	<a href="#">Table 48</a>
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	<a href="#">Table 49</a>
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	<a href="#">Table 50</a>
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	<a href="#">Table 51</a>

### 5.4.1 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 5.4.2](#)).

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

Bit	Symbol	Description	Reset value
0	ISE_SPI0	Interrupt enable.	0
2:1	-	Reserved	-
3	ISE_UART0	Interrupt enable.	0
4	ISE_UART1	Interrupt enable.	0
7:5	-	Reserved	-
8	ISE_I2C0	Interrupt enable.	0
9	-	Reserved	-
10	ISE_MRT	Interrupt enable.	0
11	ISE_CMP	Interrupt enable for both comparator.	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
22:20	-	Reserved	-
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	Interrupt enable.	0
30	ISE_PININT6	Interrupt enable.	0
31	ISE_PININT7	Interrupt enable.	0

### 5.4.2 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 5.4.1](#)).

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 41. Interrupt clear enable register 0 (ICER0, address 0xE000 E180)**

Bit	Symbol	Description	Reset value
0	ICE_SPI0	Interrupt disable.	0
2:1	-	Reserved	-
3	ICE_UART0	Interrupt disable.	0
4	ICE_UART1	Interrupt disable.	0
7:5	-	Reserved	-
8	ICE_I2C0	Interrupt disable.	0
9	-	Reserved	-
10	ICE_MRT	Interrupt disable.	0
11	-	Reserved	-
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
22:20	-	Reserved	-
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
29	ICE_PININT5	Interrupt disable.	0
30	ICE_PININT6	Interrupt disable.	0
31	ICE_PININT7	Interrupt disable.	0

### 5.4.3 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 5.4.4](#)).

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

Bit	Symbol	Description	Reset value
0	ISP_SPI0	Interrupt pending set.	0
2:1	-	Reserved	0
3	ISP_UART0	Interrupt pending set.	0
4	ISP_UART1	Interrupt pending set.	0
7:5	-	Reserved	0
8	ISP_I2C0	Interrupt pending set.	0
9	-	Reserved	0
10	ISP_MRT	Interrupt pending set.	0
11	ISP_CMP	Interrupt pending set for both comparator.	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
22:20	-	Reserved	0
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	Interrupt pending set.	0
30	ISP_PININT6	Interrupt pending set.	0
31	ISP_PININT7	Interrupt pending set.	0

#### 5.4.4 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 5.4.3](#)).

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

Bit	Symbol	Function	Reset value
0	ICP_SPI0	Interrupt pending clear.	0
2:1	-	Reserved	-
3	ICP_UART0	Interrupt pending clear.	0
4	ICP_UART1	Interrupt pending clear.	0
7:5	-	Reserved	-
8	ICP_I2C0	Interrupt pending clear.	0
9	-	Reserved	-
10	ICP_MRT	Interrupt pending clear.	0
11	ICP_CMP	Interrupt pending clear.	0
12	ICP_WDT	Interrupt pending clear.	0
13	ICP_BOD	Interrupt pending clear.	0
14	ICP_FLASH	Interrupt pending clear.	0
15	ICP_WKT	Interrupt pending clear.	0
16	ICP_ADC_SEQA	Interrupt pending clear.	0
17	ICP_ADC_SEQB	Interrupt pending clear.	0
18	ICP_ADC_THCMP	Interrupt pending clear.	0
19	ICP_ADC_OVR	Interrupt pending clear.	0
22:0	-	Reserved	-
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	Interrupt pending clear.	0
30	ICP_PININT6	Interrupt pending clear.	0
31	ICP_PININT7	Interrupt pending clear.	0

### 5.4.5 Interrupt Priority Register 0

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

Table 44. 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	-	Reserved.
21:16	-	These bits ignore writes, and read as 0.
29:22	-	Reserved.
31:30	IP_UART0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

### 5.4.6 Interrupt Priority Register 1

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

**Table 45. 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	-	Reserved.
21:16	-	These bits ignore writes, and read as 0.
23:22	-	Reserved.
29:24	-	These bits ignore writes, and read as 0.
31:30	IP_I2C1	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

### 5.4.7 Interrupt Priority Register 2

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

**Table 46. Interrupt Priority Register 2 (IPR2, address 0xE000 E408) bit description**

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

### 5.4.8 Interrupt Priority Register 3

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

**Table 47. 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.

### 5.4.9 Interrupt Priority Register 4

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

**Table 48. 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.

### 5.4.10 Interrupt Priority Register 5

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

**Table 49. 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	-	Reserved
13:8	-	These bits ignore writes, and read as 0.
15:14	-	Reserved
21:16	-	These bits ignore writes, and read as 0.
23:22	-	Reserved
29:24	-	Reserved.
31:30	IP_CT32B0	Interrupt Priority. 0 = highest priority. 3 = lowest priority.

### 5.4.11 Interrupt Priority Register 6

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

**Table 50. 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.

### 5.4.12 Interrupt Priority Register 7

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

**Table 51. Interrupt Priority Register 7 (IPR7, address 0xE000 E41C) bit description**

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



### 6.1 How to read this chapter

---

The system configuration block is identical for all LPC802 parts.

### 6.2 Features

---

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

### 6.3 Basic configuration

---

Configure the SYSCON block as follows:

- The SYSCON uses the CLKIN, CLKOUT, and  $\overline{\text{RESET}}$  pins. Configure the pin functions through the switch matrix. See [Section 6.4](#).
- No clock configuration is needed. The clock to the SYSCON block is always enabled. By default, the SYSCON block is clocked by the FRO.

#### 6.3.1 Set up the FRO

The FRO provides a selectable fro\_oscout of 18 MHz, 24 MHz, and 30 MHz outputs, which are divided to 9 MHz, 12 MHz, 15 MHz outputs to use as a system clock. Also, these frequencies can be divided down to provide frequencies (fro\_div) of 4.5 MHz, 6 MHz, and 7.5 MHz for the 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 6.6.28 “Power configuration register”](#)

2. Select the fro\_oscout (30 MHz/24 MHz/18 MHz) using the set\_fro\_frequency API call: [Chapter 7 “LPC802 FRO API ROM routine”](#) and [Figure 9 “LPC802 FRO subsystem”](#).

### 6.3.2 Configure the main clock and system clock

The clock source for the registers and memories is derived from main clock.

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. You have the following options:

- FRO: 12 MHz internal oscillator (default).
- External clock input: CLKIN from external pin.
- Low power oscillator.
- FRO DIV: 6 MHz (default).

[Section 6.6.3 “Main clock source select register”](#)

2. Update the main clock source.

[Section 6.6.4 “Main clock source update enable register”](#)

3. Select the divider value for the system clock. A divider value of 0 disables the system clock.

[Section 6.6.5 “System clock divider register”](#)

4. Select the memories and peripherals that are operating in your application and therefore must have an active clock. The core is always clocked.

[Section 6.6.9 “System clock control 0 register”](#)

## 6.4 Pin description

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

See [Section 8.3.1 “Connect an internal signal to a package pin”](#) to assign the CLKOUT function to a pin.

See [Section 8.3.2](#) to enable the clock input and the external reset input.

**Table 52. SYSCON pin description**

Function	Direction	Pin	Description	SWM register	Reference
CLKOUT	O	any	CLKOUT clock output.	PINASSIGN5	<a href="#">Table 94</a>
CLKIN	I	PIO0_1/ACMP_I2/CLKIN/ADC_0	External clock input to the main clock. Disable the ACMP_I2/ADC_0 function in the PINENABLE register.	PINENABLE0	<a href="#">Table 97</a>
RESET	I	RESET/PIO0_5	External reset input	PINENABLE0	<a href="#">Table 97</a>

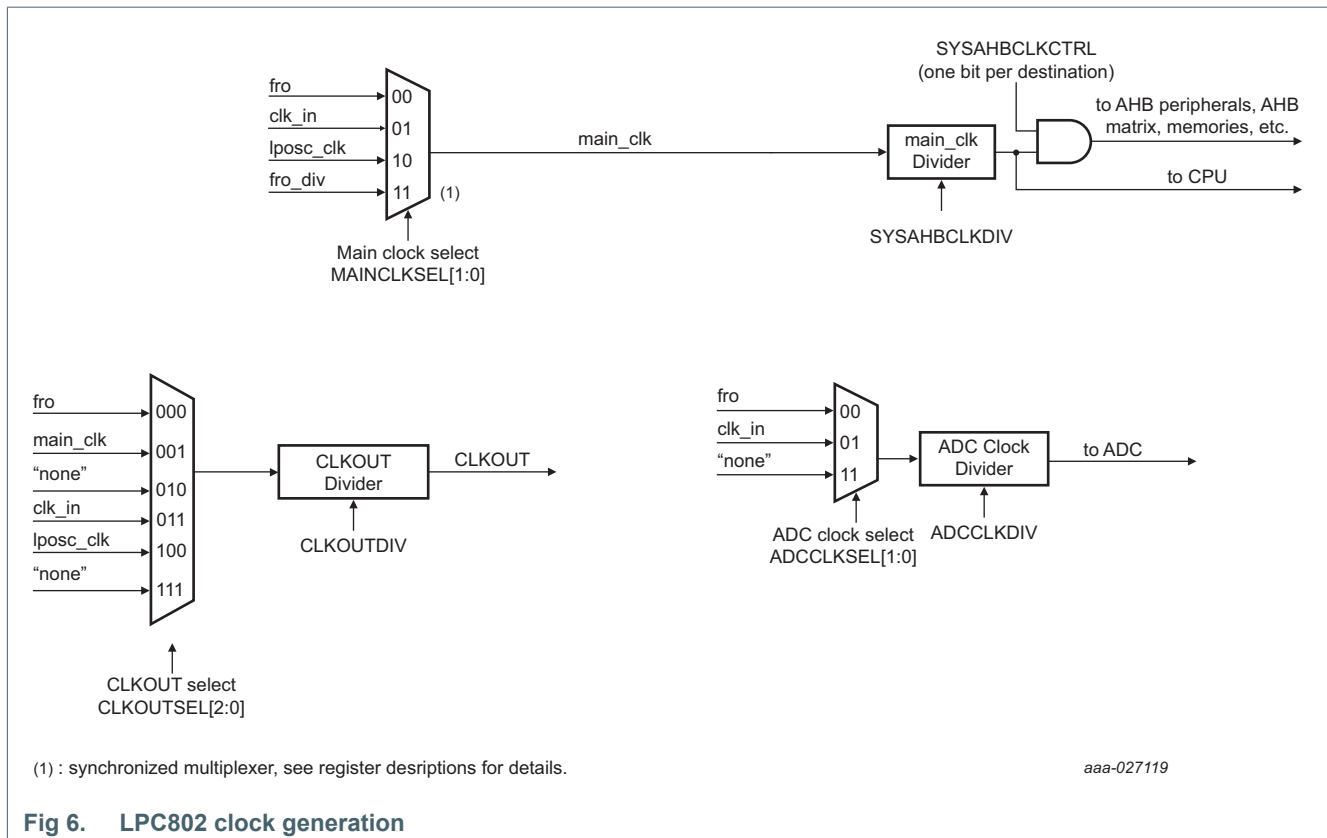
## 6.5 General description

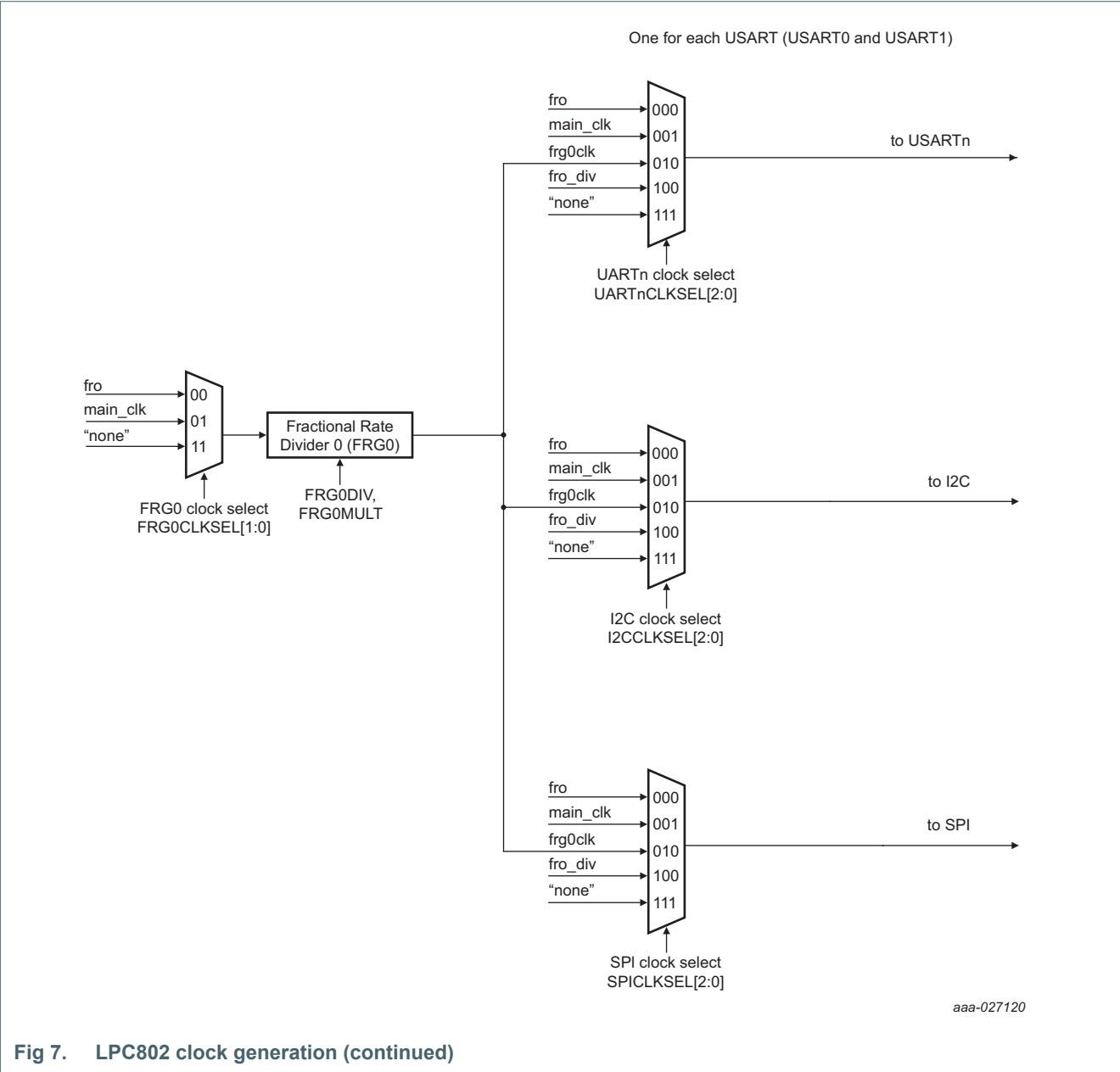
### 6.5.1 Clock generation

The system control block generates all clocks for the chip. Except for the USART clock, SPI clock, I<sup>2</sup>C clock, and ADC asynchronous, the clocks to the core and peripherals run at the same frequency. The maximum system clock frequency is 15 MHz. See [Figure 6 “LPC802 clock generation”](#).

Table 53. Clocking diagram signal name descriptions

Name	Description
clk_in	The internal clock that comes from the main CLK_IN pin function. That function must be connected to the pin by selecting it in the SWM block.
frg_clk	The output of the Fractional Rate Generator. The FRG and its source selection are shown in <a href="#">Figure 7</a> .
fro_div	Divided output of the currently selected on-chip FRO oscillator. See <a href="#">Figure 9 “LPC802 FRO subsystem”</a> .
fro	The output of the currently selected on-chip FRO oscillator. See <a href="#">Figure 9 “LPC802 FRO subsystem”</a> .
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 <a href="#">Figure 6</a> .
“none”	A tied-off source that should be selected to save power when the output of the related multiplexer is not used.
lposc_clk	The output of the 1 MHz low power oscillator. It must also be enabled in the PDRUNCFG0 register. See <a href="#">Section 6.6.28</a> .





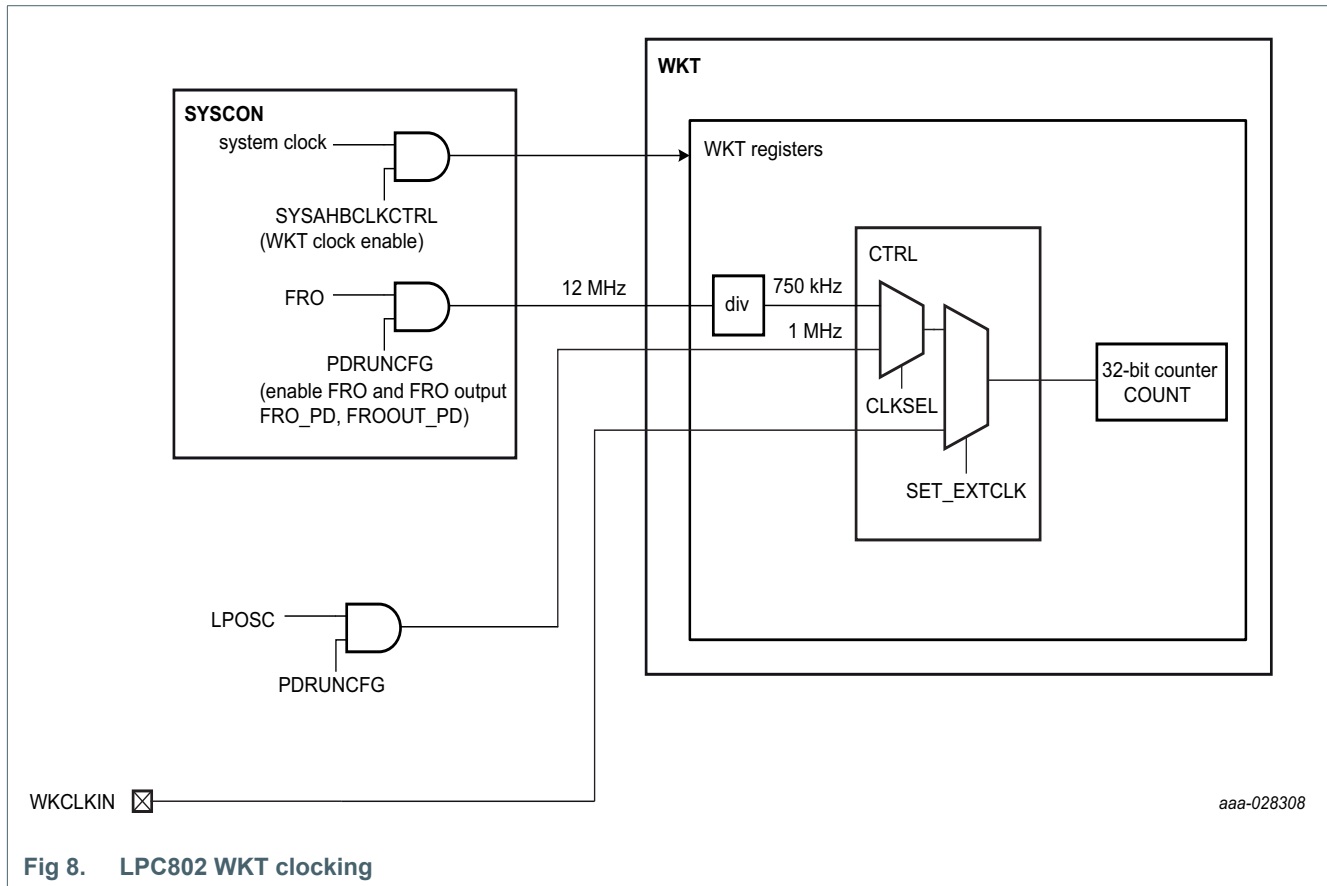


Fig 8. LPC802 WKT clocking

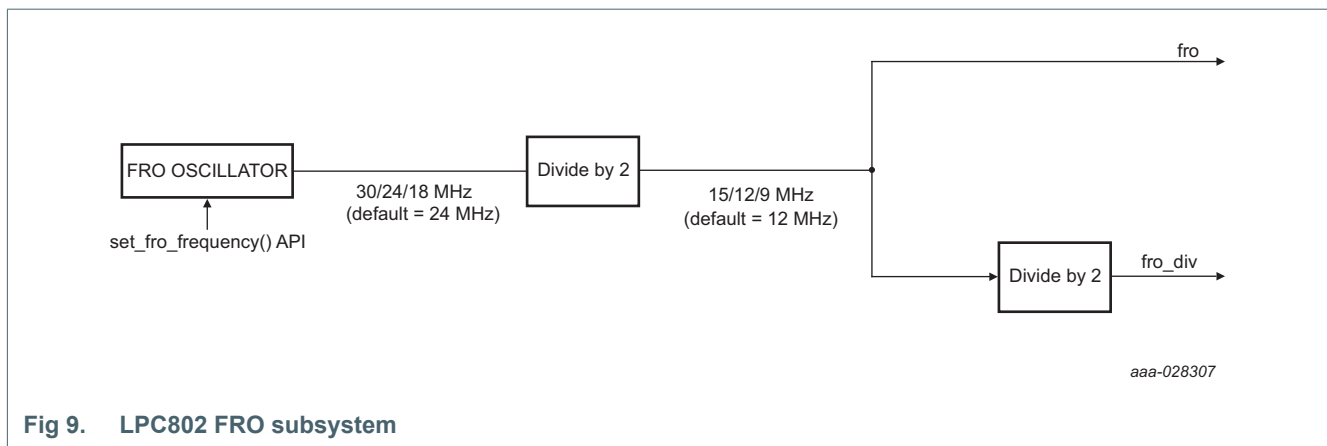


Fig 9. LPC802 FRO subsystem

## 6.5.2 Power control of analog components

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

[Section 6.6.26 “Deep-sleep mode configuration register”](#)

### 6.5.3 Configuration of reduced power-modes

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

[Section 6.6.28 “Power configuration register”](#)

[Section 6.6.25 “Start logic 1 interrupt wake-up enable register”](#)

### 6.5.4 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 6.6.23 “Pin interrupt select registers”](#)).

## 6.6 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 54](#) as reserved should not be written to.

**Table 54. Register overview: System configuration (base address 0x4004 8000)**

Name	Access	Offset	Description	Reset value	Section
SYSMEMREMAP	R/W	0x000	System memory remap	0x2	<a href="#">6.6.1</a>
-	-	0x004	Reserved	-	-
-	-	0x008	Reserved	-	-
-	-	0x00C	Reserved	-	-
-	-	0x010	Reserved	-	-
-	-	0x014	Reserved	-	-
-	-	0x018	Reserved	-	-
-	-	0x01C	Reserved	-	-
-	-	0x020	Reserved	-	-
-	-	0x024	Reserved	-	-
-	-	0x02C	Reserved	-	-
-	-	0x030	Reserved	-	-
-	-	0x034	Reserved	-	-
SYSRSTSTAT	R/W	0x038	System reset status register	0	<a href="#">6.6.2</a>
-	-	0x040	Reserved	-	-
-	-	0x044	Reserved	-	-
-	-	0x048	Reserved	-	-
-	-	0x04C	Reserved	-	-
MAINCLKSEL	R/W	0x050	Main clock source select	0	<a href="#">6.6.3</a>

Table 54. Register overview: System configuration (base address 0x4004 8000) ...continued

Name	Access	Offset	Description	Reset value	Section
MAINCLKUEN	R/W	0x054	Main clock source update enable	0	<a href="#">6.6.4</a>
SYSAHBCLKDIV	R/W	0x058	System clock divider	1	<a href="#">6.6.5</a>
-	-	0x05C	Reserved	-	-
-	-	0x060	Reserved	-	-
ADCCLKSEL	R/W	0x064	ADC clock source select	0	<a href="#">6.6.6</a>
ADCCLKDIV	R/W	0x068	ADC clock divider	0	<a href="#">6.6.7</a>
-	-	0x06C	Reserved	-	-
-	-	0x070	Reserved	-	-
-	-	0x074	Reserved	-	-
-	-	0x078	Reserved	-	-
LPOSCCLKEN	R/W	0x07C	WDT and Wake Timer clock enable control	0x1	<a href="#">6.6.8</a>
SYSAHBCLKCTRL0	R/W	0x080	System clock control 0	0x17	<a href="#">6.6.9</a>
-	-	0x084	Reserved	-	-
PRESETCTRL0	R/W	0x088	Peripheral reset control 0	0xFFFFFFFF	<a href="#">6.6.10</a>
PRESETCTRL1	R/W	0x08C	Peripheral reset control 1	0x1F	<a href="#">6.6.11</a>
UART0CLKSEL	R/W	0x090	Function clock source select for UART0	0x7	<a href="#">6.6.12</a>
UART1CLKSEL	R/W	0x094	Function clock source select for UART1	0x7	<a href="#">6.6.12</a>
-	-	0x098	Reserved	-	-
-	-	0x09C	Reserved	-	-
-	-	0x0A0	Reserved	-	-
I2C0CLKSEL	R/W	0x0A4	Function clock source select for I <sup>2</sup> C0	0x7	<a href="#">6.6.12</a>
-	-	0x0A8	Reserved	-	-
-	-	0x0AC	Reserved	-	-
-	-	0x0B0	Reserved	-	-
SPI0CLKSEL	R/W	0x0B4	Function clock source select for SPI0	0x7	<a href="#">6.6.12</a>
-	-	0x0B8	Reserved	-	-
-	-	0x0BC	Reserved	-	-
-	-	0x0C0	Reserved	-	-
-	-	0x0C4	Reserved	-	-
-	-	0x0C8	Reserved	-	-
-	-	0x0CC	Reserved	-	-
FRG0DIV	R/W	0x0D0	Fractional generator divider value	0x0	<a href="#">6.6.13</a>
FRG0MULT	R/W	0x0D4	Fractional generator multiplier value	0x0	<a href="#">6.6.14</a>
FRG0CLKSEL	R/W	0x0D8	FRG0 clock source select	0	<a href="#">6.6.15</a>
-	-	0x0DC	Reserved	-	-
-	-	0x0E0	Reserved	-	-
-	-	0x0E4	Reserved	-	-
-	-	0x0E8	Reserved	-	-
-	-	0x0EC	Reserved	-	-
CLKOUTSEL	R/W	0x0F0	CLKOUT clock source select	0	<a href="#">6.6.16</a>
CLKOUTDIV	R/W	0x0F4	CLKOUT clock divider	0	<a href="#">6.6.17</a>

Table 54. Register overview: System configuration (base address 0x4004 8000) ...continued

Name	Access	Offset	Description	Reset value	Section
-	-	0x0F8	Reserved	-	-
-	-	0x0FC	Reserved	-	-
PIOPORCAP0	R	0x100	POR captured PIO0 status 0	User dependent	<a href="#">6.6.18</a>
-	-	0x104 - 0x130	Reserved	-	-
BODCTRL	R/W	0x150	Brown-Out Detect	0x10	<a href="#">6.6.19</a>
SYSTCKCAL	R/W	0x154	System tick counter calibration	0	<a href="#">6.6.20</a>
-	-	0x158 - 0x16C	Reserved	-	-
IRQLATENCY	R/W	0x170	IRQ delay. Allows trade-off between interrupt latency and determinism.	0x0000 0010	<a href="#">6.6.21</a>
NMISRC	R/W	0x174	NMI Source Control	0	<a href="#">6.6.22</a>
PINTSEL0	R/W	0x178	GPIO Pin Interrupt Select register 0	0	<a href="#">6.6.23</a>
PINTSEL1	R/W	0x17C	GPIO Pin Interrupt Select register 1	0	<a href="#">6.6.23</a>
PINTSEL2	R/W	0x180	GPIO Pin Interrupt Select register 2	0	<a href="#">6.6.23</a>
PINTSEL3	R/W	0x184	GPIO Pin Interrupt Select register 3	0	<a href="#">6.6.23</a>
PINTSEL4	R/W	0x188	GPIO Pin Interrupt Select register 4	0	<a href="#">6.6.23</a>
PINTSEL5	R/W	0x18C	GPIO Pin Interrupt Select register 5	0	<a href="#">6.6.23</a>
PINTSEL6	R/W	0x190	GPIO Pin Interrupt Select register 6	0	<a href="#">6.6.23</a>
PINTSEL7	R/W	0x194	GPIO Pin Interrupt Select register 7	0	<a href="#">6.6.23</a>
-	-	0x198 - 0x200	Reserved	-	-
STARTERP0	R/W	0x204	Start logic 0 pin wake-up enable register	0	<a href="#">6.6.24</a>
-	-	0x208 - 0x210	Reserved	-	-
STARTERP1	R/W	0x214	Start logic 1 interrupt wake-up enable register	0	<a href="#">6.6.25</a>
-	-	0x218 - 0x22C	Reserved	-	-
PDSLEEP_CFG	R/W	0x230	Power-down states in deep-sleep mode	0xFFFF	<a href="#">6.6.26</a>
PDAWAKE_CFG	R/W	0x234	Power-down states for wake-up from deep-sleep	0x8050	<a href="#">6.6.27</a>
PDRUN_CFG	R/W	0x238	Power configuration register	0x8050	<a href="#">6.6.28</a>
-	-	0x23C - 0x3F4	Reserved	-	-
DEVICE_ID	R	0x3F8	Device ID	part dependent	<a href="#">6.6.29</a>

### 6.6.1 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 55. 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	-

### 6.6.2 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 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 56](#) applies to the POR reset.

**Table 56. System reset status register (SYSRSTSTAT, address 0x4004 8038) bit description**

Bit	Symbol	Value	Description	Reset value
0	POR_BOD		POR and BOD reset status	0
		0	No POR and BOD detected	
		1	POR and BOD detected. Writing a one clears this reset.	
1	EXTRST		Status of the external 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	-	-	Reserved	-
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	-

### 6.6.3 Main clock source select register

The MAINCLKSEL register selects the main\_clock\_pre\_pll, which can be the FRO, external clock, low power oscillator, or FRO\_DIV.

Bit 0 of the MAINCLKUEN register ([Section 6.6.4](#)) must be toggled from 0 to 1 for the update to take effect.

Table 57. 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	0
		0x0	FRO	
		0x1	External clock	
		0x2	Low power oscillator	
		0x3	FRO_DIV	
31:2	-	-	Reserved	-

#### 6.6.4 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 58. 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	-

#### 6.6.5 System clock divider register

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

Table 59. 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	-

#### 6.6.6 ADC clock source select register

The ADCCLKSEL register selects the ADC clock, which can be the FRO or external\_clk.

**Table 60. 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	External clock	
		0x2	None	
		0x3	None	
31:3	-		Reserved	-

### 6.6.7 ADC clock divider register

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

**Table 61. 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	-

### 6.6.8 WDT and Wake Timer clock enable control register

**Table 62. WDT and Wake Timer clock enable control register (LPOSCCLKEN, address 0x4004807C) bit description**

Bit	Symbol	Value	Description	Reset value
0	WDT		Enables clock for WDT	1
		0	Disable	
		1	Enable	
1	WKT		Enables clock for Wake Timer	0
		0	Disable	
		1	Enable	
31:2	-		Reserved	0

### 6.6.9 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 63. 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		Enables clock for SRAM0.	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			Reserved	0
9	WKT		Enables clock for self-wake-up timer.	0
		0	Disable	
		1	Enable	
10	MRT		Enables clock for multi-rate timer.	0
		0	Disable	
		1	Enable	
11	SPI0		Enables clock for SPI0.	0
		0	Disable	
		1	Enable	
12			Reserved	
13	CRC		Enables clock for CRC.	0
		0	Disable	
		1	Enable	
14	UART0		Enables clock for USART0.	0
		0	Disable	
		1	Enable	

**Table 63. System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description ...continued**

Bit	Symbol	Value	Description	Reset value
15	UART1		Enables clock for USART1.	0
		0	Disable	
		1	Enable	
16			Reserved	0
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	
23:20			Reserved	0
24	ADC		Enables clock to ADC.	0
		0	Disable	
		1	Enable	
25	CTIMER0	-	Enables clock for CTIMER0	0
		0	Disable	
		1	Enable	
27:26			Reserved	0
28	GPIO_INT		Enable clock for GPIO pin interrupt registers.	0
		0	Disable	
		1	Enable	
31:29			Reserved	0

### 6.6.10 Peripheral reset control 0 register

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

Table 64. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
3:0	-		Reserved	4'b1111
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		I2C0 reset control	1
		0	Assert the I2C0 reset.	
		1	Clear the I2C0 reset.	
6	GPIO0_RST_N		GPIO0 reset control	1
		0	Assert the GPIO0 reset	
		1	Clear the GPIO0 reset.	
7	SWM_RST_N		SWM reset control	1
		0	Assert the SWM reset.	
		1	Clear the SWM reset.	
8			Reserved	1
9	WKT_RST_N		Self-wake-up timer (WKT) reset control	1
		0	Assert the WKT reset.	
		1	Clear the WKT reset.	
10	MRT_RST_N		Multi-rate timer (MRT) reset control	1
		0	Assert the MRT reset.	
		1	Clear the MRT reset.	
11	SPI0_RST_N		11SPI0_RST_N	1
		0	Assert the SPI0 reset.	
		1	Clear the SPI0 reset.	
12			Reserved	1
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.	
17:16	-	-	Reserved	

Table 64. Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description

Bit	Symbol	Value	Description	Reset value
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.	
23:20			Reserved	1
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.	
27:26	-	-	Reserved	1
28	GPIOINT_RST_N		GPIOINT reset control	1
		0	Assert the GPIOINT reset.	
		1	Clear the GPIOINT reset.	
31:29			Reserved	1

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

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

### 6.6.12 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 66](#).

- UART0 clock source select register (UART0CLKSEL, address 0x4004 8090).
- UART1 clock source select register (UART1CLKSEL, address 0x4004 8094).
- I<sup>2</sup>C0 clock source select register (I2C0CLKSEL, address 0x4004 80A4).
- SPI0 clock source select register (SPI0CLKSEL, address 0x4004 80B4).

**Table 66. 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	None	
		0x4	FRO_DIV	
		0x5	Reserved	
		0x6	Reserved	
		0x7	None	
31:3	-	-	Reserved	-

### 6.6.13 Fractional generator 0 divider value register

The UART, I<sup>2</sup>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:

$$\text{frg0clk} = \text{frg0\_src\_clk} / (1 + \text{MULT}/\text{DIV}).$$

FRG0\_SRC\_CLK is input clock of fractional generator 0, which can be the FRO or main clock.

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

- The DIV value programmed in this register is the denominator of the divider used by the fractional rate generator to create the fractional component of FRG0CLK.
- The MULT value of the fractional divider is programmed in the FRG0MULT register. See [Table 68](#).

**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 13.3.1 “Configure the USART clock and baud rate”](#).

[Section 13.7.1 “Clocking and baud rates”](#).

**Table 67. Fractional generator 0 divider value register (FRG0DIV, address 0x4004 80D0) 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	-

#### 6.6.14 Fractional generator 0 multiplier value register

The UART, I<sup>2</sup>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:

$$\text{frg0clk} = \text{frg0\_src\_clk} / (1 + \text{MULT} / \text{DIV}).$$

FRG0\_SRC\_CLK is input clock of fractional generator 0, which can be the FRO or main clock.

The fractional portion  $(1 + \text{MULT} / \text{DIV})$  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 67](#).
- 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 13.3.1 “Configure the USART clock and baud rate”](#).

[Section 13.7.1 “Clocking and baud rates”](#).

**Table 68. Fractional generator 0 multiplier value register (FRG0MULT, address 0x4004 80D4) 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	-

### 6.6.15 FRG0 clock source select register

The FRG0CLKSEL register selects the frg0\_src clock, which can be the FRO or main clock.

**Table 69. 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	None	
		0x3	None	
31:2	-	-	Reserved	-

### 6.6.16 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 70. 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	None	
		0x3	External clock	
		0x4	Low power oscillator	
		0x5	None	
		0x6	None	
		0x7	None	
31:3	-	-	Reserved	0

### 6.6.17 CLKOUT clock divider register

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

**Table 71. 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	-

### 6.6.18 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 72. POR captured PIO status register 0 (PIOPORCAP0, address 0x4004 8100) bit description**

Bit	Symbol	Description	Reset value
31:0	PIOSTAT	State of PIO0_0 to PIO0_5 and PIO0_7 to PIO0_17 at power-on reset	Implementation dependent

### 6.6.19 BOD control register

The BOD control register selects three separate threshold values for sending a BOD interrupt to the NVIC and one threshold level for forced reset. Reset and interrupt threshold values listed in [Table 73](#) 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 LPC802 data sheet for the BOD reset and interrupt levels.

**Table 73. BOD control register (BODCTRL, address 0x4004 8150) bit description**

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

### 6.6.20 System tick counter calibration register

This register determines the value of the SYST\_CALIB register.

**Table 74. 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	-

### 6.6.21 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 75. 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	-

### 6.6.22 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 38](#). For a description of the NMI functionality, see [Section 5.3.2 “Non-Maskable Interrupt \(NMI\)”](#).

**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 76. 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 <a href="#">Table 38</a> 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.

### 6.6.23 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 5 and 7 to 17 for pins PIO0\_0 to PIO0\_5 and PIO0\_7 to PIO0\_17 to the INTPIN bits. 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 38](#)).

To use the selected pins for pin interrupts or the pattern match engine, see [Section 11.5.2 “Pattern match engine”](#).

**Table 77. 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_5 and PIO0_7 to PIO0_17 correspond to numbers 0 to 5 and 7 to 17.	0
31:6	-	Reserved	-

### 6.6.24 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 38 “Connection of interrupt sources to the NVIC”](#).

**Table 78. Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description**

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

**Table 78. Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description ...continued**

Bit	Symbol	Value	Description	Reset value
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	-

### 6.6.25 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 38 “Connection of interrupt sources to the NVIC”](#).

**Table 79. 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	
2:1	-		Reserved	-
3	USART0		USART0 interrupt wake-up. Configure USART in synchronous slave mode.	0
		0	Disabled	
		1	Enabled	
4	USART1		USART1 interrupt wake-up. Configure USART in synchronous slave mode.	0
		0	Disabled	
		1	Enabled	
7:5	-		Reserved	-
8	I2C0		I2C0 interrupt wake-up.	0
		0	Disabled	
		1	Enabled	
11:9	-		Reserved	-

**Table 79. Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description ...continued**

Bit	Symbol	Value	Description	Reset value
12	WWDT		WWDT interrupt wake-up	0
		0	Disabled	
		1	Enabled	
13	BOD		BOD interrupt wake-up	0
		0	Disabled	
		1	Enabled	
14	-		Reserved	-
15	WKT		Self-wake-up timer interrupt wake-up	0
		0	Disabled	
		1	Enabled	
31:20	-		Reserved.	-

### 6.6.26 Deep-sleep mode configuration register

The bits in this register (BOD\_PD and LPOSC\_OD) can be programmed to control aspects of deep-sleep and power-down modes. The bits are loaded into corresponding bits of the PDRUNCFG register when deep-sleep mode or power-down mode is entered.

**Remark:** Hardware forces the analog blocks to be powered down in deep-sleep and power-down modes. An exception are the BOD and low power oscillator, which can be configured to remain running through this register. The LPOSC\_PD value written to the PDSLEEPCFG register is overwritten if the LOCK bit in the WWDT MOD register (see [Table 220](#)) is set. See [Section 17.5.3](#) for details.

**Table 80. Deep-sleep configuration register (PDSLEEPCFG, address 0x4004 8230) bit description**

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0b111
3	BOD_PD		BOD power-down control for deep-sleep and power-down mode	1
		0	Powered	
		1	Powered down	
5:4	-		Reserved.	11
6	LPOSC_PD		Low power oscillator power-down control for deep-sleep and power-down mode. Changing this bit to powered-down has no effect when the LOCK bit in the WWDT MOD register is set. In this case, the low power oscillator is always running.	1
		0	Powered	
		1	Powered down	
15:7	-		Reserved	0b11111111
31:16	-	-	Reserved	0

### 6.6.27 Wake-up configuration register

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

**Table 81. 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			Reserved	
6	LPOSC_PD		Low power oscillator wake-up configuration. Changing this bit to powered-down has no effect when the LOCK bit in the WWDT MOD register is set. In this case, the low power oscillator is always running.	1
		0	Powered	
		1	Powered down	
14:7	-		Reserved. Always write these bits as 0x00	0x00
15	ACMP		Analog comparator wake-up configuration	1
		0	Powered	
		1	Powered down	
31:16	-	-	Reserved	0

### 6.6.28 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.



Table 82. Power configuration register (PDRUNCFG, address 0x4004 8238) bit description

Bit	Symbol	Value	Description	Reset value
0	FROOUT_PD		FRO oscillator output power	0
		0	Powered	
		1	Powered down	
1	FRO_PD		FRO oscillator power down	0
		0	Powered	
		1	Powered down	
2	FLASH_PD		Flash power down	0
		0	Powered	
		1	Powered down	
3	BOD_PD		BOD power down	0
		0	Powered	
		1	Powered down	
4	ADC_PD		ADC wake-up configuration	1
		0	Powered	
		1	Powered down	
5			Reserved	1
6	LPOSC_PD		Low power oscillator power down. Changing this bit to powered-down has no effect when the LOCK bit in the WWDT MOD register is set. In this case, the low power oscillator is always running.	1
		0	Powered	
		1	Powered down	
14:7	-		Reserved. Always write these bits as 0x00	0x00
15	ACMP		Analog comparator power down	1
		0	Powered	
		1	Powered down	
31:16	-	-	Reserved	0

### 6.6.29 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 21](#)).

Table 83. Device ID register (DEVICE\_ID, address 0x4004 83F8) bit description

Bit	Symbol	Description	Reset value
31:0	DEVICEID	Used to read the part identification number.	part-dependent

Table 84. LPC802 device identification numbers

Part number	Part ID
LPC802M001JDH16	0x00008023
LPC802M001JDH20	0x00008021

Table 84. LPC802 device identification numbers

Part number	Part ID
LPC802M011JDH20	0x00008022
LPC802M001JHI33	0x00008024
LPC802UK	0x00008021

## 6.7 Functional description

### 6.7.1 Reset

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

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

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.

### 6.7.2 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 73](#)).

The interrupt signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC (see [Table 39](#)) 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 79](#)) 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.

### 7.1 How to read this chapter

The ROM-based set fro\_oscout API call is available on all parts.

### 7.2 Features

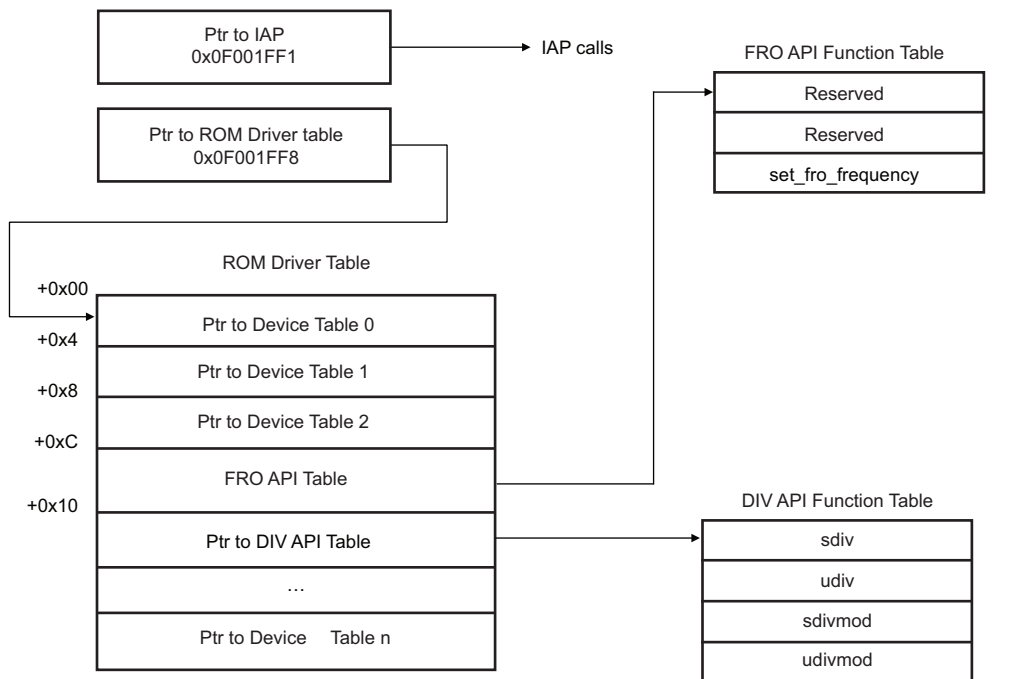
- Select desired on-chip fro\_oscout.

### 7.3 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 select desired fro\_oscout (30 MHz/ 24 MHz/18 MHz) and this is divided by two to provide FRO frequencies of 15 MHz, 12 MHz, and 9 MHz. This is performed by executing a function, which is pointed by a pointer within the ROM Driver Table. [Figure 10](#) shows the pointer structure used to call the set FRO frequency API.

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



aaa-028332

Fig 10. ROM pointer structure

## 7.4 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 85. FRO API call**

Function prototype	API description	Reference
<code>void set_fro_frequency(uint32_t iFreq);</code>	Setup the fro_oscout to either 30 MHz, 24 MHz or 18 MHz.	<a href="#">7.4.1</a>

### 7.4.1 set\_fro\_frequency

This routine sets up the fro\_oscout (30 MHz/24 MHz/18 MHz) to provide frequencies of 15 MHz, 12 MHz, and 9 MHz. The requested frequency is set up and the appropriate factory trim value is used.

See [Figure 9 “LPC802 FRO subsystem”](#) for more details to select FRO frequency.

[Table 86](#) shows the set\_fro\_frequency routine.

**Table 86. set\_fro\_frequency routine**

Routine	sidiv
Prototype	<code>void set_fro_frequency(uint32_t iFreq);</code>
Input parameter	<b>Param0</b> — 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.

#### 7.4.1.1 Param0: frequency

The frequency is the required fro\_oscout, 30 MHz, 24 MHz or 18 MHz. The fro\_oscout is internally divided by two to provide frequencies of 15 MHz, 12 MHz, and 9 MHz.

### 8.1 How to read this chapter

---

The switch matrix is identical for all LPC802 parts.

### 8.2 Features

---

- Flexible assignment of digital peripheral functions to pins
- Enable/disable of analog functions
- Provides level shifter functionality to allow up to two selected signals to be routed from user-selected pins in one voltage domain to selected pins in the alternate domain. This feature can also be used on a single supply device if voltage level shifting is not required.

### 8.3 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.

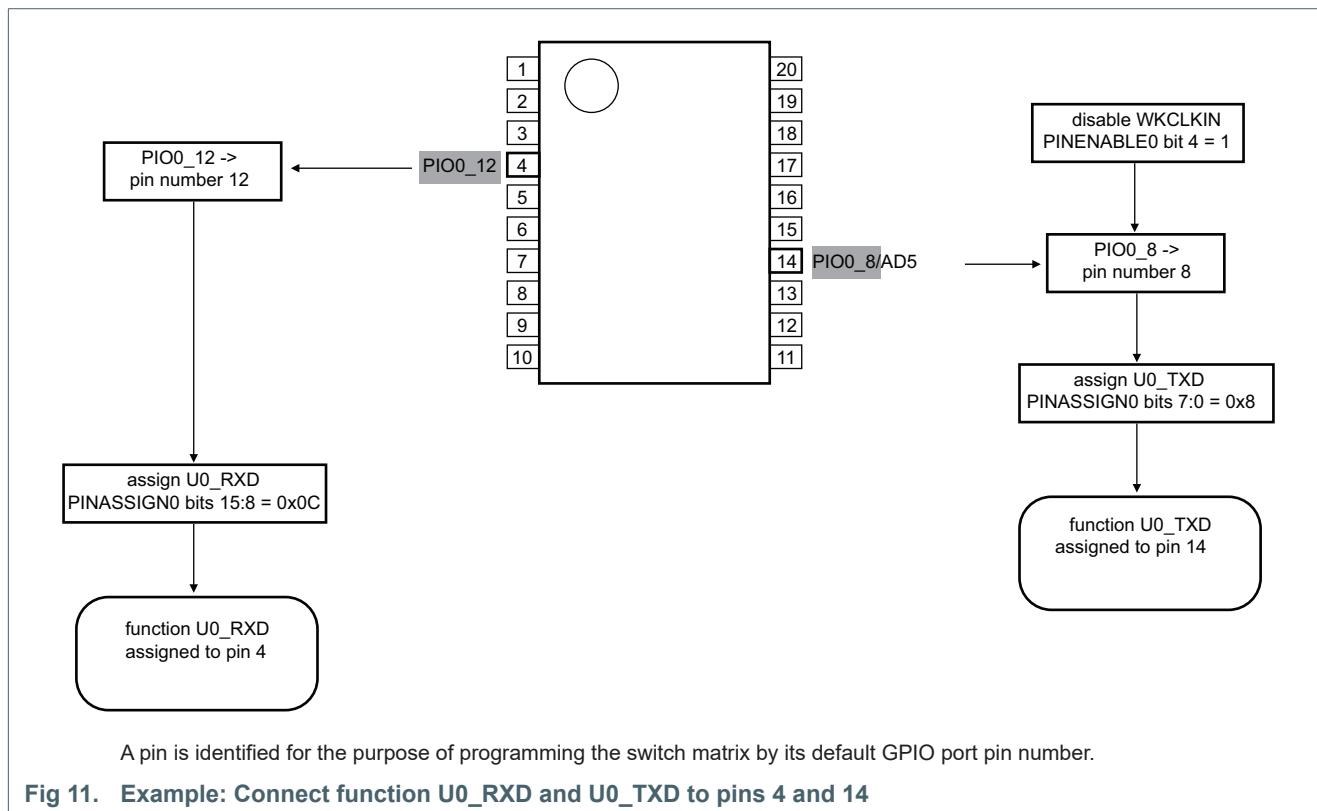
Before activating a peripheral or enabling its interrupt, use the switch matrix to connect the peripheral to external pins.

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.

### 8.3.1 Connect an internal signal to a package pin



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. 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 87](#) or in the data sheet.
2. Use the LPC802 data sheet to decide which pin x on the LPC802 package to connect the pin function to.
3. Use the pin description table to find the default GPIO function PIO0\_n 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.
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.

### 8.3.2 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 LPC802 package, disable any special function available on this pin in the PINENABLE0 register and do not assign any movable function to it.

By default, all pins except pins PIO0\_2, PIO0\_3, and PIO0\_5 are assigned to GPIO.

- 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 register. All other possible functions on this pins are now disabled.

### 8.3.3 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.



## 8.4 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, and 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 analog comparator inputs can only be assigned to one particular external pin with the appropriate electrical characteristics. These functions are called fixed-pin functions. If a fixed-pin function is not used, it can be replaced by any other movable function.

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

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

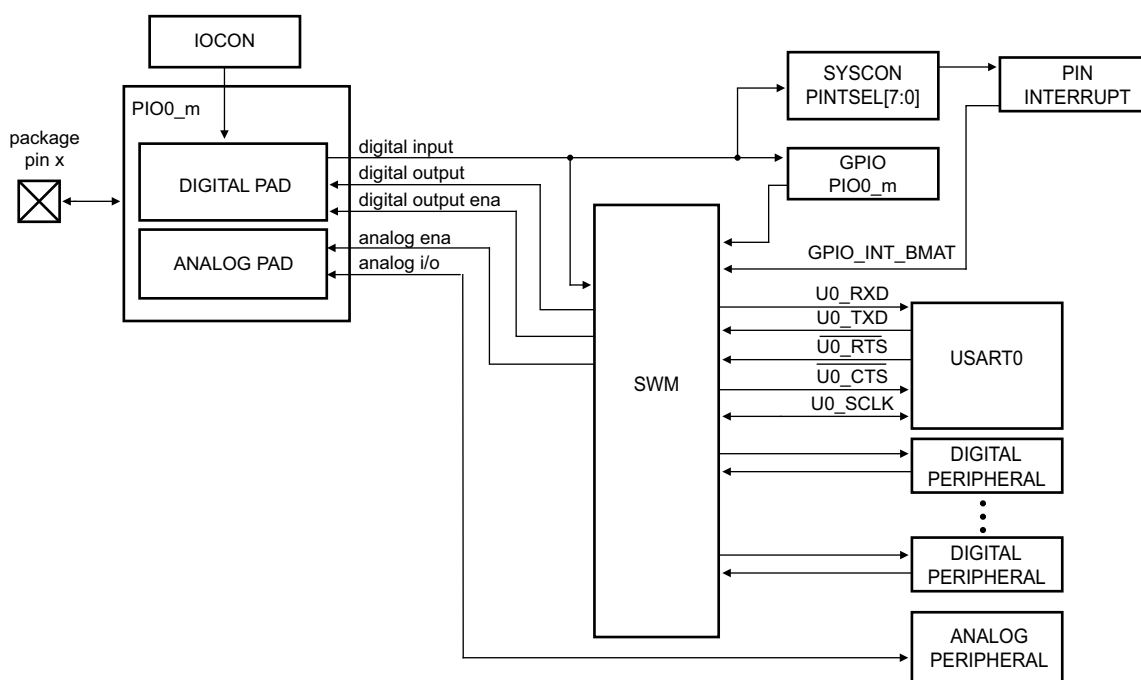


Fig 12. Functional diagram of the switch matrix

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

- It is allowed to connect one input signal on a pin to multiple internal inputs by programming the same pin number in more than one PINASSIGN register.

Example:

You can enable the CLKIN input in the PINENABLE0 register on pin PIO0\_1 and also assign other block inputs to pin PIO0\_1 through the PINASSIGN registers to feed the CLKIN into the other block.

- 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.

- It is not allowed to 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.

### 8.4.1 Movable functions

**Table 87. Movable functions (assign to pins PIO0\_0 to PIO0\_5 and PIO0\_7 to PIO0\_17 through switch matrix)**

Function name	Type	Description	SWM Pin assign register	Reference
U0_TXD	O	Transmitter output for USART0.	PINASSIGN0	<a href="#">Table 89</a>
U0_RXD	I	Receiver input for USART0.	PINASSIGN0	<a href="#">Table 89</a>
U0_RTS	O	Request To Send output for USART0.	PINASSIGN0	<a href="#">Table 89</a>
U0_CTS	I	Clear To Send input for USART0.	PINASSIGN0	<a href="#">Table 89</a>
U0_SCLK	I/O	Serial clock input/output for USART0 in synchronous mode.	PINASSIGN1	<a href="#">Table 90</a>
U1_TXD	O	Transmitter output for USART1.	PINASSIGN1	<a href="#">Table 90</a>
U1_RXD	I	Receiver input for USART1.	PINASSIGN1	<a href="#">Table 90</a>
U1_SCLK	I/O	Serial clock input/output for USART1 in synchronous mode.	PINASSIGN1	<a href="#">Table 91</a>
SPI0_SCK	I/O	Serial clock for SPI0.	PINASSIGN2	<a href="#">Table 91</a>
SPI0_MOSI	I/O	Master Out Slave In for SPI0.	PINASSIGN2	<a href="#">Table 91</a>
SPI0_MISO	I/O	Master In Slave Out for SPI0.	PINASSIGN2	<a href="#">Table 91</a>
SPI0_SSEL0	I/O	Slave select 0 for SPI0.	PINASSIGN2	<a href="#">Table 91</a>
SPI0_SSEL1	I/O	Slave select 1 for SPI0.	PINASSIGN3	<a href="#">Table 92</a>
TO_CAP0	I	Time Capture channel 0.	PINASSIGN3	<a href="#">Table 92</a>
TO_CAP1	I	Time Capture channel 1.	PINASSIGN3	<a href="#">Table 92</a>
TO_CAP2	I	Time Capture channel 2.	PINASSIGN3	<a href="#">Table 92</a>
TO_MAT0	O	Timer Match channel 0.	PINASSIGN4	<a href="#">Table 93</a>
TO_MAT1	O	Timer Match channel 1.	PINASSIGN4	<a href="#">Table 93</a>
TO_MAT2	O	Timer Match channel 2.	PINASSIGN4	<a href="#">Table 93</a>

Table 87. Movable functions (assign to pins PIO0\_0 to PIO0\_5 and PIO0\_7 to PIO0\_17 through switch matrix)

Function name	Type	Description	SWM Pin assign register	Reference
TO_MAT3	O	Timer Match channel 3.	PINASSIGN4	<a href="#">Table 93</a>
I2C0_SDA	I/O	I <sup>2</sup> C0-bus data input/output.	PINASSIGN5	<a href="#">Table 94</a>
I2C0_SCL	I/O	I <sup>2</sup> C0-bus clock input/output.	PINASSIGN5	<a href="#">Table 94</a>
ACMP_O	O	Analog comparator output.	PINASSIGN5	<a href="#">Table 94</a>
CLKOUT	O	Clock output.	PINASSIGN5	<a href="#">Table 94</a>
GPIO_INT_BMAT	O	Output of the pattern match engine.	PINASSIGN6	<a href="#">Table 95</a>
LVLSHFT_IN0	I	Level shift input 0.	PINASSIGN6	<a href="#">Table 95</a>
LVLSHFT_IN1	I	Level shift input 1.	PINASSIGN6	<a href="#">Table 95</a>
LVLSHFT_OUT0	O	Level shift output 0.	PINASSIGN6	<a href="#">Table 95</a>
LVLSHFT_OUT1	O	Level shift output 1.	PINASSIGN7	<a href="#">Table 96</a>

### 8.4.2 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\_5 and PIO0\_7 to PIO0\_17, for assignment through the pin-assignment registers, are numbered 0 to 5 and 7 to 17.

There are two types of functions which must be assigned to port pins in different ways:

#### 1. Movable functions (PINASSIGN0 to 7):

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, 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):

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 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, except RESETN. The SWCLK and SWDIO fixed pin functions are enabled by the boot ROM if the CRP setting allows it.

- If a fixed-pin analog function is selected, its assigned pin cannot be used for any other function.

### 8.4.3 Level Shifter function

The LPC802 provides a dual voltage I/O feature. The pins on one side of the package are supplied by VDDIO and the pins on the other side are supplied by V<sub>DD</sub>. Each of these two supplies can be connected to different voltages within the allowed V<sub>DD</sub> range. This feature allows the device to level-shift signals from one off-chip voltage domain to another. The switch matrix provides level shifter functionality to allow up to two selected signals to be routed from user-selected pins in one voltage domain to selected pins in the alternate domain. This feature can also be used on a single supply device if voltage level shifting is not required. See [Table 95 “Pin assign register 6 \(PINASSIGN6, address 0x4000 C018\) bit description”](#) and [Table 96 “Pin assign register 7 \(PINASSIGN7, address 0x4000 C01C\) bit description”](#) for details.

## 8.5 Register description

Table 88. 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	<a href="#">Table 89</a>
PINASSIGN1	R/W	0x004	Pin assign register 1. Assign movable functions U0_SCLK, U1_TXD, U1_RXD, U1_SCLK.	0xFFFF FFFF	<a href="#">Table 90</a>
PINASSIGN2	R/W	0x008	Pin assign register 2. Assign movable functions SPI0_SCK, SPI0_MOSI, SPI0_MISO, SPI0_SSEL0.	0xFFFF FFFF	<a href="#">Table 91</a>
PINASSIGN3	R/W	0x00C	Pin assign register 3. Assign movable function SPI0_SSEL1, T0_CAP0, T0_CAP1, T0_CAP2.	0xFFFF FFFF	<a href="#">Table 92</a>
PINASSIGN4	R/W	0x010	Pin assign register 4. Assign movable functions T0_MAT0, T0_MAT1, T0_MAT2, T0_MAT3.	0xFFFF FFFF	<a href="#">Table 93</a>
PINASSIGN5	R/W	0x014	Pin assign register 5. Assign movable functions I2C0_SDA, I2C0_SCL, COMP0_OUT, CLKOUT.	0xFFFF FFFF	<a href="#">Table 94</a>
PINASSIGN6	R/W	0x018	Pin assign register 6. Assign movable functions GPIO_INT_BMAT, LVLSHFT_IN0, LVLSHFT_IN1, LVLSHFT_OUT0.	0xFFFF FFFF	<a href="#">Table 95</a>
PINASSIGN7	R/W	0x01C	Pin assign register 7. Assign movable functions LVLSHFT_OUT1.	0xFFFF FFFF	<a href="#">Table 96</a>
PINENABLE0	R/W	0x1C0	Pin enable register 0. Enables fixed-pin functions ACMP_I1, ACMP_I2, ACMP_I3, ACMP_I4, SWCLK, SWDIO, RESET, CLKIN, WKCLKIN, VDDCMP, ADC_0 - ADC_11.	0x3FFF8F	<a href="#">Table 97</a>

### 8.5.1 Pin assign register 0

Table 89. 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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	0xFF

### 8.5.2 Pin assign register 1

Table 90. 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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	0xFF
31:24	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	0xFF

### 8.5.3 Pin assign register 2

Table 91. Pin assign register 2 (PINASSIGN2, address 0x4000 C008) bit description

Bit	Symbol	Description	Reset value
7:0	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	0xFF

Table 91. Pin assign register 2 (PINASSIGN2, address 0x4000 C008) bit description

Bit	Symbol	Description	Reset value
15:8	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
23:16	SPI0_MISO_IO	SPI0_MISO function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
31:24	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF

### 8.5.4 Pin assign register 3

Table 92. Pin assign register 3 (PINASSIGN3, address 0x4000 C00C) bit description

Bit	Symbol	Description	Reset value
7:0	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17 (= 0x11).	0xFF

### 8.5.5 Pin assign register 4

Table 93. Pin assign register 4 (PINASSIGN4, address 0x4000 C010) bit description

Bit	Symbol	Description	Reset value
7:0	T0_MAT0	T0_MAT0 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
15:8	T0_MAT1	T0_MAT1 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
23:16	T0_MAT2	T0_MAT2 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
31:24	T0_MAT3	T0_MAT3 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF

### 8.5.6 Pin assign register 5

Table 94. Pin assign register 5 (PINASSIGN5, address 0x4000 C014) bit description

Bit	Symbol	Description	Reset value
7:0	I2C0_SDA_IO	I2C0_SDA function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
15:8	I2C0_SCL_IO	I2C0_SCL function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
23:16	ACMP_O	ACMP_O function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
31:24	CLKOUT_O	CLKOUT function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF

### 8.5.7 Pin assign register 6

Table 95. Pin assign register 6 (PINASSIGN6, address 0x4000 C018) bit description

Bit	Symbol	Description	Reset value
7:0	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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_17(= 0x11).	0xFF
15:8	LVLSHFT_IN0	LVLSHFT_IN0 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_30 (= 0x1E).	0xFF
23:16	LVLSHFT_IN1	LVLSHFT_IN1 function assignment. The value is the pin number to be assigned to this function. The following pins are available: PIO0_0 (= 0) to PIO0_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_30 (= 0x1E).	0xFF
31:24	LVLSHFT_OUT0	LVLSHFT_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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_30 (= 0x1E).	0xFF

### 8.5.8 Pin assign register 7

Table 96. Pin assign register 7 (PINASSIGN7, address 0x4000 C01C) bit description

Bit	Symbol	Description	Reset value
7:0	LVLSHFT_OUT1	LVLSHFT_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_5 (= 0x5) and from PIO0_7 (= 0x7) to PIO0_30 (= 0x1E).	0xFF
31:8	-	Reserved.	-



## 8.5.9 PINENABLE 0

Table 97. 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.	
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_16.	
		1	ACMP_I4 disabled.	
4	SWCLK		SWCLK function select.	0
		0	SWCLK enabled on pin PIO0_3.	
		1	SWCLK disabled.	
5	SWDIO		SWDIO function select.	0
		0	SWDIO enabled on pin PIO0_2.	
		1	SWDIO disabled.	
6	RESETN		RESETN function select.	0
		0	RESETN enabled on pin PIO0_5.	
		1	RESETN disabled.	
7	CLKIN		CLKIN function select.	1
		0	CLKIN enabled on pin PIO0_1.	
		1	CLKIN disabled.	
8	WKCLKIN		WKCLKIN function select.	1
		0	WKCLKIN enabled on pin PIO0_11.	
		1	WKCLKIN disabled.	
9	VDDCMP		VDDCMP function select.	1
		0	VDDCMP enabled on pin PIO0_7.	
		1	VDDCMP disabled.	
10	ADC_0		ADC_0 function select.	1
		0	ADC_0 enabled on pin PIO0_1.	
		1	ADC_0 disabled.	
11	ADC_1		ADC_1 function select.	1
		0	ADC_1 enabled on pin PIO0_7.	
		1	ADC_1 disabled.	
12	ADC_2		ADC_2 function select.	1
		0	ADC2 enabled on pin PIO0_14.	
		1	ADC2 disabled.	



Table 97. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description

Bit	Symbol	Value	Description	Reset value
13	ADC_3		ADC_3 function select.	1
		0	ADC_3 enabled on pin PIO0_16.	
		1	ADC_3 disabled.	
14	ADC_4		ADC_4 function select.	1
		0	ADC_4 enabled on pin PIO0_9.	
		1	ADC_4 disabled.	
15	ADC_5		ADC_5 function select.	1
		0	ADC_5 enabled on pin PIO0_8.	
		1	ADC_5 disabled.	
16	ADC_6		ADC_6 function select.	1
		0	ADC_6 enabled on pin PIO0_11.	
		1	ADC_6 disabled.	
17	ADC_7		ADC_7 function select.	1
		0	ADC_7 enabled on pin PIO0_10.	
		1	ADC_7 disabled.	
18	ADC_8		ADC_8 function select.	1
		0	ADC_8 enabled on pin PIO0_15.	
		1	ADC_8 disabled.	
19	ADC_9		ADC_9 function select.	1
		0	ADC_9 enabled on pin PIO0_17.	
		1	ADC_9 disabled.	
20	ADC_10		ADC_10 function select.	1
		0	ADC_10 enabled on pin PIO0_13.	
		1	ADC_10 disabled.	
21	ADC_11		ADC_11 function select.	1
		0	ADC_11 enabled on pin PIO0_4.	
		1	ADC_11 disabled.	
31:22			Reserved	

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

### 9.1 How to read this chapter

The IOCON block is identical for all LPC802 parts. Registers for pins that are not available on a specific package are reserved.

**Table 98. Pinout summary**

Package	Pins/configuration registers available
HVQFN33	PIO0_0 to PIO0_5, PIO0_7 to PIO0_17.
TSSOP16	PIO0_0 to PIO0_5, PIO0_7 to PIO0_9, PIO0_11 to PIO0_13, PIO0_17.
TSSOP20 (DS)	PIO0_0 to PIO0_5, PIO0_7 to PIO0_16.
TSSOP20 (SS)	PIO0_0 to PIO0_5, PIO0_7 to PIO0_17.

### 9.2 Features

The following electrical properties are configurable for each pin:

- Pull-up/pull-down resistor.
- Open-drain mode.
- Hysteresis.
- Analog mode (for a subset of pins, see the LPC802 data sheet).

The I2C supports data rates up to 400 kbit/s on standard digital pins.

### 9.3 Basic configuration

Enable the clock to the IOCON in the SYSAHBCLKCTRL register ([Table 63](#), bit 18). Once the pins are configured, you can disable the IOCON clock to conserve power.

## 9.4 General description

### 9.4.1 Pin configuration

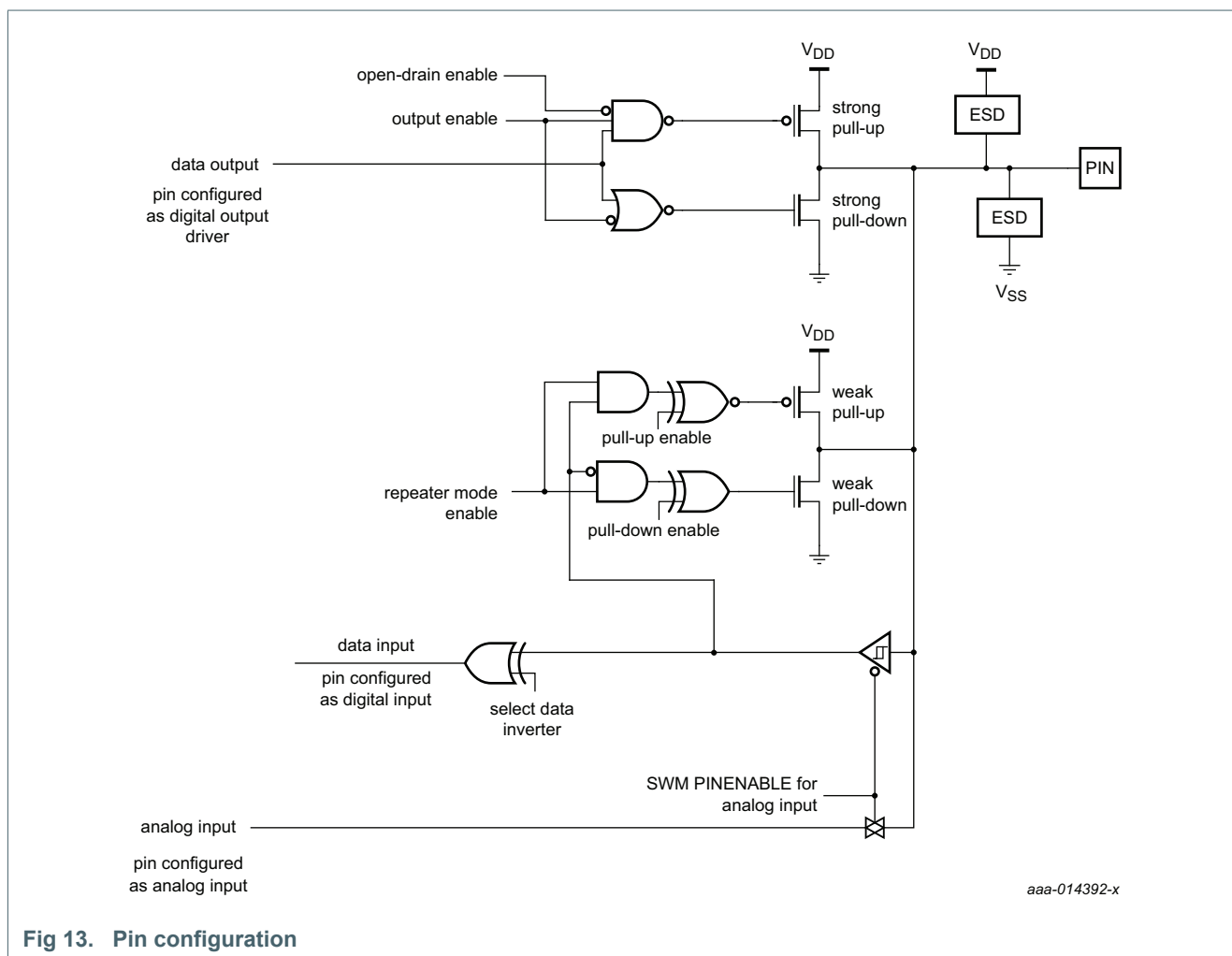


Fig 13. Pin configuration

### 9.4.2 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 87 “Movable functions \(assign to pins PIO0\\_0 to PIO0\\_5 and PIO0\\_7 to PIO0\\_17 through switch matrix\)”](#)

### 9.4.3 Pin mode

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

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

#### 9.4.4 Open-drain mode

An open-drain mode can be enabled for all digital I/O 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  $V_{DD} = 0$ .

#### 9.4.5 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 must be disabled via the IOCON register.

## 9.5 Register description

Each port pin  $PIO0\_m$  has one IOCON register assigned to control the pin's function and electrical characteristics.

**Remark:** See [Table 100](#) for the IOCON register map ordered by pin name.

**Table 99. 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	0x0000 00B0	<a href="#">Table 101</a>
PIO0_13	R/W	0x004	I/O configuration for pin PIO0_13/ADC_10	0x0000 00B0	<a href="#">Table 102</a>
PIO0_12	R/W	0x008	I/O configuration for pin PIO0_12	0x0000 00B0	<a href="#">Table 103</a>
PIO0_5	R/W	0x00C	I/O configuration for pin PIO0_5/RESET	0x0000 00B0	<a href="#">Table 104</a>
PIO0_4	R/W	0x010	I/O configuration for pin PIO0_4/ADC_11/TRSTN	0x0000 00B0	<a href="#">Table 105</a>
PIO0_3	R/W	0x014	I/O configuration for pin PIO0_3/SWCLK	0x0000 00B0	<a href="#">Table 106</a>
PIO0_2	R/W	0x018	I/O configuration for pin PIO0_2/SWDIO	0x0000 00B0	<a href="#">Table 107</a>
PIO0_11	R/W	0x01C	I/O configuration for pin PIO0_11/ADC_6/WKCLKIN	0x0000 00B0	<a href="#">Table 108</a>
PIO0_10	R/W	0x020	I/O configuration for pin PIO0_10/ADC_7	0x0000 00B0	<a href="#">Table 109</a>
PIO0_16	R/W	0x024	I/O configuration for pin PIO0_16/ADC_3/ACMP_14	0x0000 00B0	<a href="#">Table 110</a>
PIO0_15	R/W	0x028	I/O configuration for pin PIO0_15/ADC_8	0x0000 00B0	<a href="#">Table 111</a>
PIO0_1	R/W	0x02C	I/O configuration for pin PIO0_1/ADC_0/ACMP_I2/CLKIN	0x0000 00B0	<a href="#">Table 112</a>
-	-	0x030	Reserved	0x0000 00B0	-
PIO0_9	R/W	0x034	I/O configuration for pin PIO0_9/ADC_4	0x0000 00B0	<a href="#">Table 113</a>
PIO0_8	R/W	0x038	I/O configuration for pin PIO0_8/ADC_5	0x0000 0090	<a href="#">Table 114</a>

Table 99. 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/VDDCMP	0x0000 0090	<a href="#">Table 115</a>
-	-	0x040	Reserved.		-
PIO0_0	R/W	0x044	I/O configuration for pin PIO0_0/ACMP_I1	0x0000 00B0	<a href="#">Table 116</a>
PIO0_14	R/W	0x048	I/O configuration for pin PIO0_14/ACMP_I3/ADC_2	0x0000 00B0	<a href="#">Table 117</a>

Table 100. I/O configuration registers ordered by pin name

Name	Address offset	True open-drain	Analog <sup>[1]</sup>	High-drive output	Reference
PIO0_0	0x044	no	yes	no	<a href="#">Table 116</a>
PIO0_1	0x02C	no	yes	no	<a href="#">Table 112</a>
PIO0_2	0x018	no	no	yes	<a href="#">Table 107</a>
PIO0_3	0x014	no	no	yes	<a href="#">Table 106</a>
PIO0_4	0x010	no	yes	no	<a href="#">Table 105</a>
PIO0_5	0x00C	no	no	no	<a href="#">Table 104</a>
PIO0_7	0x03C	no	yes	no	<a href="#">Table 115</a>
PIO0_8	0x038	no	yes	no	<a href="#">Table 114</a>
PIO0_9	0x034	no	yes	no	<a href="#">Table 113</a>
PIO0_10	0x020	no	yes	no	<a href="#">Table 109</a>
PIO0_11	0x01C	no	yes	no	<a href="#">Table 108</a>
PIO0_12	0x008	no	no	yes	<a href="#">Table 103</a>
PIO0_13	0x004	no	yes	no	<a href="#">Table 102</a>
PIO0_14	0x048	no	yes	no	<a href="#">Table 117</a>
PIO0_15	0x028	no	yes	no	<a href="#">Table 111</a>
PIO0_16	0x024	no	yes	no	<a href="#">Table 110</a>
PIO0_17	0x000	no	yes	no	<a href="#">Table 101</a>

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

### 9.5.1 PIO0\_17 register

Table 101. 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).	0b10
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	

Table 101. PIO0\_17 register (PIO0\_17, address 0x4004 4000) 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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.2 PIO0\_13 register

Table 102. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.3 PIO0\_12 register

Table 103. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.4 PIO0\_5 register

Table 104. 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).	0b10
		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

Table 104. PIO0\_5 register (PIO0\_5, address 0x4004 400C) bit description ...continued

Bit	Symbol	Value	Description	Reset value
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.5 PIO0\_4 register

Table 105. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.6 PIO0\_3 register

Table 106. 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).	0b10
		0x0	Inactive (no pull-down/pull-up resistor enabled).	
		0x1	Pull-down resistor enabled.	
		0x2	Pull-up resistor enabled.	
		0x3	Repeater mode.	



Table 106. 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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.7 PIO0\_2 register

Table 107. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.8 PIO0\_11 register

Table 108. PIO0\_11 register (PIO0\_11, address 0x4004 401C) 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).	0b10
		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.	-
10	OD	-	Open-drain mode	0
		0	Disable	
		1	Open-drain mode enabled.	
31:11	-	-	Reserved.	-

### 9.5.9 PIO0\_10 register

Table 109. PIO0\_10 register (PIO0\_10, address 0x4004 4020) 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).	0b10
		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.	-
10	OD	-	Open-drain mode	-
		0	Disable	
		1	Open-drain mode enabled.	
31:11	-	-	Reserved.	-

### 9.5.10 PIO0\_16 register

Table 110. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.11 PIO0\_15 register

Table 111. 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).	0b10
		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

Table 111. PIO0\_15 register (PIO0\_15, address 0x4004 4028) bit description ...continued

Bit	Symbol	Value	Description	Reset value
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.12 PIO0\_1 register

Table 112. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.13 PIO0\_9 register

Table 113. 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).	0b10
		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.	

Table 113. PIO0\_9 register (PIO0\_9, address 0x4004 4034) bit description ...continued

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.	001
10	OD		Open-drain mode.	0
		0	Disable.	
		1	Open-drain mode enabled. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.14 PIO0\_8 register

Table 114. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.15 PIO0\_7 register

Table 115. PIO0\_7 register (PIO0\_7, address 0x4004 403C) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-		Reserved.	0

Table 115. PIO0\_7 register (PIO0\_7, address 0x4004 403C) bit description ...continued

Bit	Symbol	Value	Description	Reset value
4:3	MODE		Selects function mode (on-chip pull-up/pull-down resistor control).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 9.5.16 PIO0\_0 register

Table 116. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

## 9.5.17 PIO0\_14 register

Table 117. 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).	0b10
		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. <b>Remark:</b> This is not a true open-drain mode.	
31:11	-	-	Reserved.	0

### 10.1 How to read this chapter

The GPIO registers control the pins on each port. Depending on the package type, not all pins are available, and the corresponding bits in the GPIO registers are reserved.

Table 118. GPIO pins available

Package	GPIO Port 0
HVQFN33	PIO0_0 to PIO0_5, PIO0_7 to PIO0_17
TSSOP20 (LPC802M001JDH20)	PIO0_0 to PIO0_5, PIO0_7 to PIO0_17
TSSOP20 (LPC802M011JDH20)	PIO0_0 to PIO0_5, PIO0_7 to PIO0_16
TSSOP16	PIO0_0 to PIO0_5, PIO0_7 to PIO0_9, PIO0_11 to PIO0_13, PIO0_17

### 10.2 Basic configuration

For the GPIO port registers, enable the clock to the GPIO port in the SYSAHBCLKCTRL register ([Table 63](#)).

### 10.3 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.

### 10.4 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 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.

### 10.5 Register description

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

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

**Remark:** The reset value of external in [Table 119](#) 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 119. Register overview: GPIO port (base address 0xA000 0000)

Name	Access	Address offset	Description	Reset value	Width	Reference
B0 to B17	R/W	0x0000 to 0x0011	Byte pin registers port 0 <a href="#">[1]</a>	external	byte (8 bit)	<a href="#">Table 120</a>
W0 to W17	R/W	0x1000 to 0x1044	Word pin registers port 0	external	word (32 bit)	<a href="#">Table 121</a>
DIR0	R/W	0x2000	Direction registers port 0	0	word (32 bit)	<a href="#">Table 122</a>
MASK0	R/W	0x2080	Mask register port 0	0	word (32 bit)	<a href="#">Table 123</a>
PIN0	R/W	0x2100	Port pin register port 0	external	word (32 bit)	<a href="#">Table 124</a>
MPIN0	R/W	0x2180	Masked port register port 0	external	word (32 bit)	<a href="#">Table 125</a>
SET0	R/W	0x2200	Write: Set register for port 0 Read: output bits for port 0	0	word (32 bit)	<a href="#">Table 126</a>
CLR0	WO	0x2280	Clear port 0	-	word (32 bit)	<a href="#">Table 127</a>
NOT0	WO	0x2300	Toggle port 0	-	word (32 bit)	<a href="#">Table 128</a>
DIRSET0	WO	0x2380	Set pin direction bits for port 0.	0	word (32 bit)	<a href="#">Table 129</a>
DIRCLR0	WO	0x2400	Clear pin direction bits for port 0.	-	word (32 bit)	<a href="#">Table 130</a>
DIRNOT0	WO	0x2480	Toggle pin direction bits for port 0.	-	word (32 bit)	<a href="#">Table 131</a>

[1] PIO0\_6 is not available for all registers. PIO0\_0 to PIO0\_5 and PIO0\_7 to PIO0\_17 are available for all registers. For LPC802M011JDH20, PIO0\_0 to PIO0\_5 and PIO0\_7 to PIO0\_16 are available for all registers.

### 10.5.1 GPIO port byte pin registers

Each GPIO pin has a byte register in this address range. Bytes read in this range will be 0 if the pin is LOW or 0x01 if the pin is HIGH, regardless of direction, masking, or alternate function, except that pins configured as analog I/O always read as zeros. Writes will set or clear the pins output bit based on bit 0 of the byte written. 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 120. GPIO port byte pin registers (B[0:17], addresses 0xA000 0000 (B0) to 0xA000 0011 (B17)) bit description

Bit	Symbol	Description	Reset value	Access
0	PBYTE	Read: state of the pin PIO0_n, at byte offset n, regardless of direction, masking, or alternate function, except that pins configured as analog I/O always read as 0. Write: loads the pin's output bit.	external	R/W
7:1		Reserved (0 on read, ignored on write)	0	-

### 10.5.2 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 121. GPIO port word pin registers (W[0:17], addresses 0xA000 1000 (W0) to 0xA000 1044 (W17)) bit description

Bit	Symbol	Description	Reset value	Access
17:0	PWORD	Word for pin PIO0_n is access at offset n. Read 0: pin PIO0_n is LOW. Write 0: clear output bit. Read 0xFFFF FFFF: pin PIO0_n is HIGH. Write any value 0x0000 0001 to 0xFFFF FFFF: set output bit. <b>Remark:</b> Only 0 or 0xFFFF FFFF can be read. Writing any value other than 0 will set the output bit.	external	R/W
31:18		Reserved	0	-

### 10.5.3 GPIO port direction registers

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

Table 122. GPIO direction port register (DIR0), address 0xA000 2000 bit description

Bit	Symbol	Description	Reset value	Access
17:0	DIRP	Selects pin direction for pin PIO0_n (bit 0 = PIO_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = input. 1 = output.	0	R/W
31:18		Reserved	0	-

### 10.5.4 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 123. GPIO mask port register (MASK0), address 0xA000 2080 bit description

Bit	Symbol	Description	Reset value	Access
17:0	MASKP	Controls which bits corresponding to PIO0_n are active in the MPORT register (bit 0 = PIO0_0, bit 1 = PIO_1, ..., bit 17 = PIO0_17). 0 = Read MPORT: pin state; write MPORT: load output bit. 1 = Read MPORT: 0; write MPORT: output bit not affected.	0	R/W
31:18		Reserved	0	-

### 10.5.5 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 124. GPIO port pin register (PIN0), address 0xA000 2100 bit description

Bit	Symbol	Description	Reset value	Access
17:0	PORT	Reads pin states or loads output bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = Read: pin is low; write: clear output bit. 1 = Read: pin is high; write: set output bit.	external	R/W
31:18		Reserved	0	-

### 10.5.6 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 125. GPIO masked port pin register (MPIN0), address 0xA000 2180 bit description

Bit	Symbol	Description	Reset value	Access
17:0	MPORTP	Masked port register (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 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.	external	R/W
31:18		Reserved	0	-

### 10.5.7 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 126. GPIO port set register (SET0), address 0xA000 2200 bit description

Bit	Symbol	Description	Reset value	Access
17:0	SETP	Read or set output bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = Read: output bit; write: no operation. 1 = Read: output bit; write: set output bit.	0	R/W
31:18		Reserved	0	-

### 10.5.8 GPIO port clear registers

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

Table 127. GPIO port clear register (CLR0), address 0xA000 2280 bit description

Bit	Symbol	Description	Reset value	Access
17:0	CLRP	Clear output bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = No operation. 1 = Clear output bit.	NA	WO
31:18		Reserved	0	-

### 10.5.9 GPIO port toggle registers

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

Table 128. GPIO port toggle register (NOT0), address 0xA000 2300 bit description

Bit	Symbol	Description	Reset value	Access
17:0	NOTP	Toggle output bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = no operation. 1 = Toggle output bit.	NA	WO
31:18		Reserved	0	-

### 10.5.10 GPIO port direction set registers

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

Table 129. GPIO port direction set register (DIRSET0), address 0xA000 2380 bit description

Bit	Symbol	Description	Reset value	Access
17:0	DIRSETP	Set direction bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = No operation. 1 = Set direction bit.	0	WO
31:18		Reserved	0	-

### 10.5.11 GPIO port direction clear registers

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

Table 130. GPIO port direction clear register (DIRCLR0), 0xA000 2400 bit description

Bit	Symbol	Description	Reset value	Access
17:0	DIRCLRP	Clear direction bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = No operation. 1 = Clear direction bit.	NA	WO
31:18		Reserved	0	-

### 10.5.12 GPIO port direction toggle registers

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

Table 131. GPIO port direction toggle register (DIRNOTP), address 0xA000 2480 bit description

Bit	Symbol	Description	Reset value	Access
17:0	DIRNOTP	Toggle direction bits (bit 0 = PIO0_0, bit 1 = PIO0_1, ..., bit 17 = PIO0_17). 0 = no operation. 1 = Toggle direction bit.	NA	WO
31:18		Reserved	0	-

## 10.6 Functional description

### 10.6.1 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 from bit 0 of a Byte Pin register (bits 7:1 are always zero).
- 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.

### 10.6.2 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 10.6.1](#) returns the state of pins, regardless of their direction or alternate functions.

### 10.6.3 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.

### 10.6.4 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.

### 10.6.5 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.

### 11.1 How to read this chapter

---

The pin interrupt generator and the pattern match engine are available on all LPC802 parts.

### 11.2 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.

### 11.3 Basic configuration

---

- Pin interrupts:
  - Select up to eight external interrupt pins from all GPIO port pins in the SYSCON block ([Table 77](#)). 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 63](#), 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 78](#)).
  - 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 77](#)). 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 63](#), 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 (PINASSIGN6, [Table 95](#)).

### 11.3.1 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 LPC802 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 6.6.23 “Pin interrupt select registers”](#) in the SYSCON block for the PINTSEL registers.

## 11.4 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 6.6.23 “Pin interrupt select registers”](#).

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

See [Section 8.3.1 “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 132. Pin interrupt/pattern match engine pin description**

Function	Direction	Pin	Description	SWM register	Reference
GPIO_INT_BMAT	O	any	GPIO pattern match output	PINASSIGN6	<a href="#">Table 95</a>

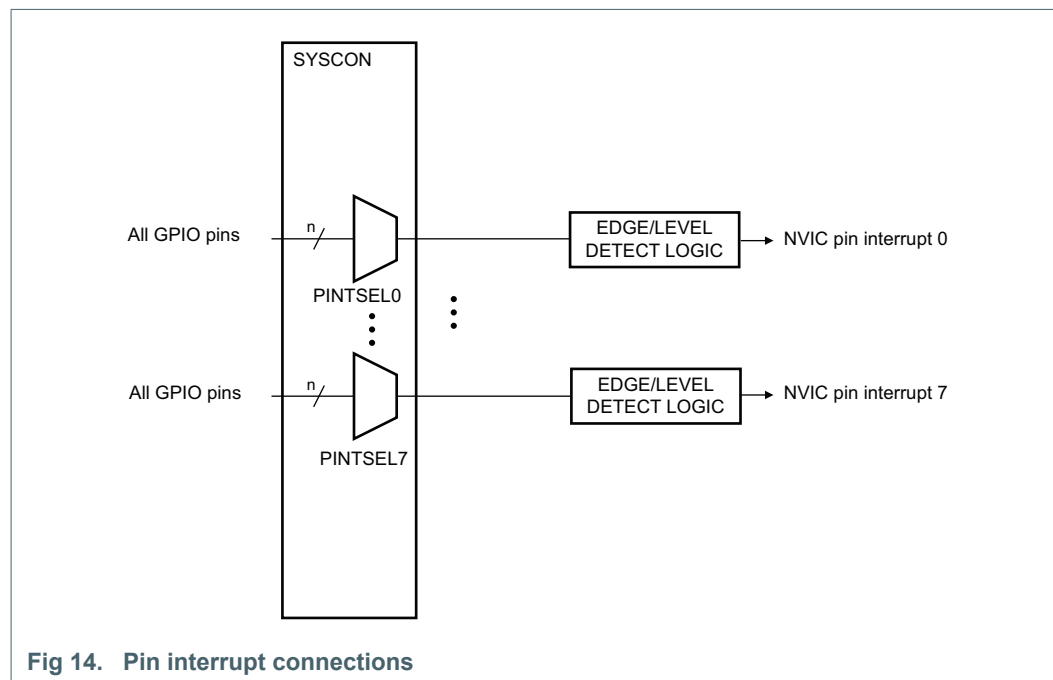
## 11.5 General description

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.



### 11.5.1 Pin interrupts

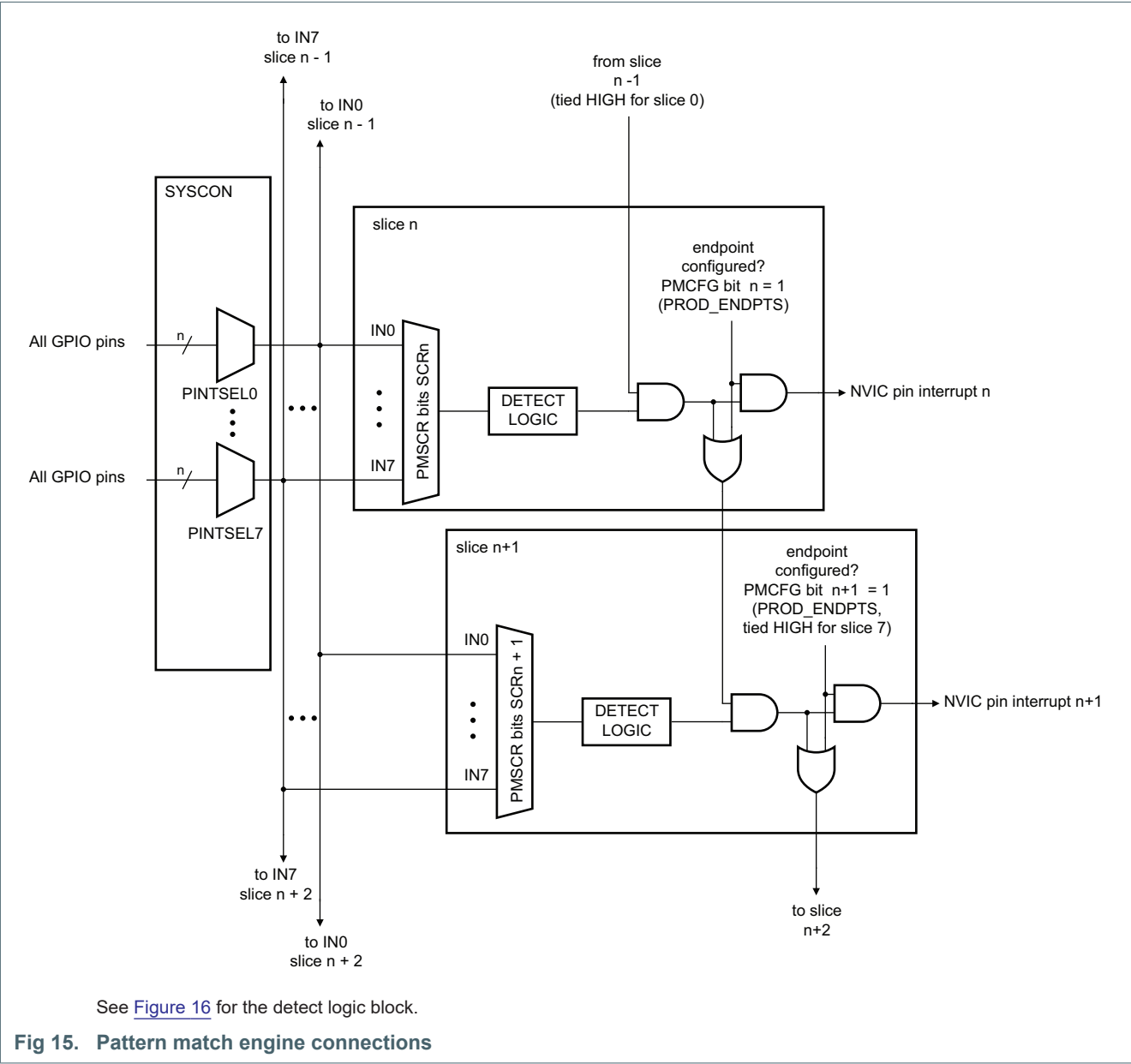
From all available GPIO pins, up to eight pins can be selected in the system control block to serve as external interrupt pins (see [Table 77](#)). The external interrupt pins are connected to eight individual interrupts in the NVIC and are created based on rising or falling edges or on the input level on the pin.



### 11.5.2 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.



The detect logic of each slice can detect the following events on the selected input:

- Edge with memory (sticky): A rising edge, a falling edge, or a rising or falling edge that is detected at any time after the edge-detection mechanism has been cleared. The input qualifies as detected (the detect logic output remains HIGH) until the pattern match engine detect logic is cleared again.
- Event (non-sticky): Every time an edge (rising or falling) is detected, the detect logic output for this pin goes HIGH. This bit is cleared after one clock cycle, and the detect logic can detect another edge,
- Level: A HIGH or LOW level on the selected input.

Figure 16 shows the details of the edge detection logic for each slice.

You can combine a sticky event with non-sticky events to create a pin interrupt whenever a rising or falling edge occurs after a qualifying edge event.

You can create a time window during which rising or falling edges can create a pin interrupt by combining a level detect with an event detect. See [Section 11.7.3](#) for details.

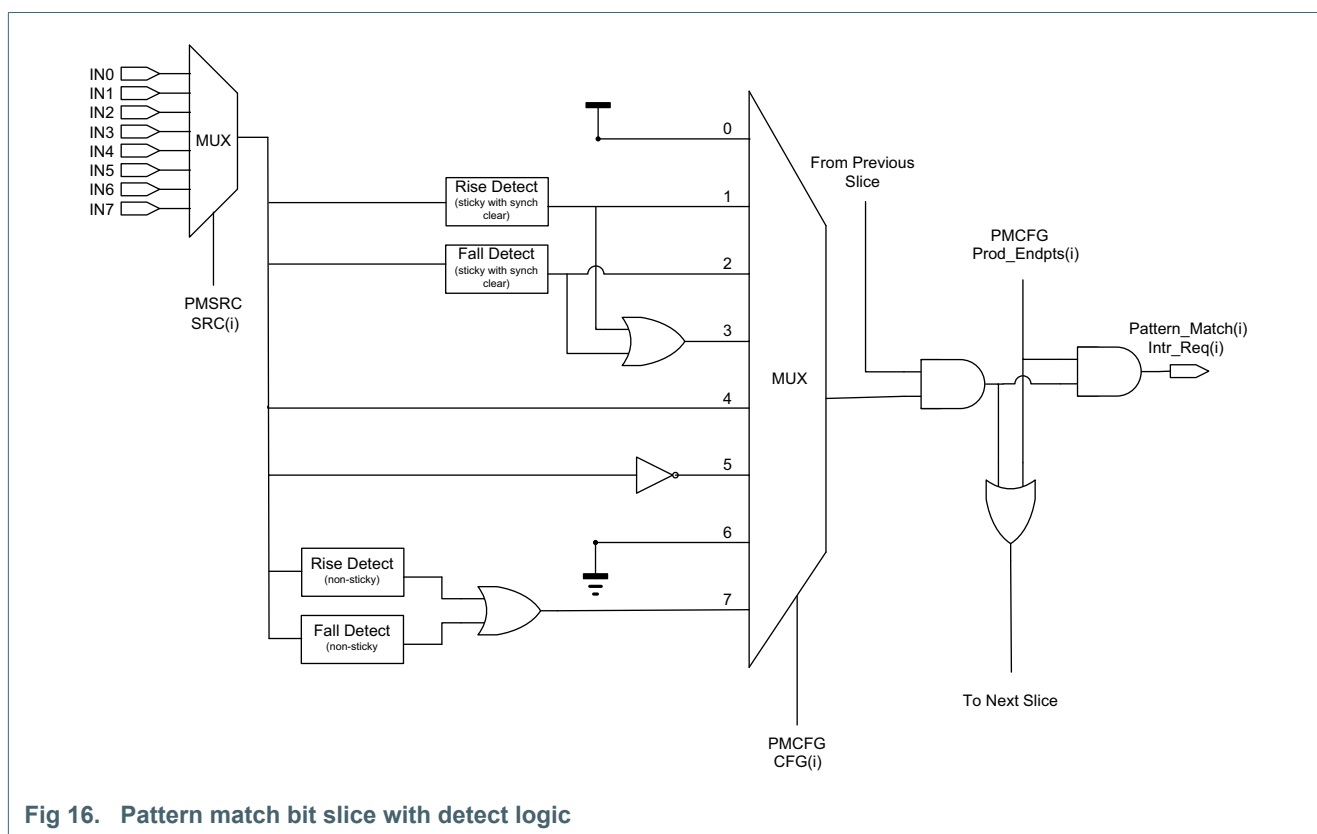


Fig 16. Pattern match bit slice with detect logic

### 11.5.2.1 Inputs and outputs of the pattern match engine

The connections between the pins and the pattern match engine are shown in [Figure 15](#). All inputs to the pattern match engine are selected in the SYSCON block and can be GPIO port pins or another pin function depending on the switch matrix configuration.

The pattern match logic continuously monitors the eight inputs and generates interrupts when any one or more minterms (product terms) of the specified boolean expression is matched. A separate interrupt request is generated for each individual minterm.

In addition, the pattern match module can be enabled to generate a Receive Event (RXEV) output to the Arm core when a boolean expression is true (i.e. when any minterm is matched).

The RXEV output is also be routed to GPIO\_INT\_BMAT pin. This allows the GPIO module to provide a rudimentary programmable logic capability employing up to eight inputs and one output.

The pattern match function utilizes the same eight interrupt request lines as the pin interrupts, so these two features are mutually exclusive as far as interrupt generation is concerned. A control bit is provided to select whether interrupt requests are generated in response to the standard pin interrupts or to pattern matches. Note that, if the pin interrupts are selected, the RXEV request to the CPU can still be enabled for pattern matches.

**Remark:** Pattern matching cannot be used to wake the part up from deep-sleep or power-down mode. Pin interrupts must be selected to use the pins for wake up.

### 11.5.2.2 Boolean expressions

The pattern match module is constructed of eight bit-slice elements. Each bit slice is programmed to represent one component of one minterm (product term) within the boolean expression. The interrupt request associated with the last bit slice for a particular minterm will be asserted whenever that minterm is matched.  
(See bit slice drawing [Figure 16](#)).

The pattern match capability can be used to create complex software state machines. Each minterm (and its corresponding individual interrupt) represents a different transition event to a new state. Software can then establish the new set of conditions (that is a new boolean expression) that will cause a transition out of the current state.

Example:

Assume the expression:  $(IN0) \sim (IN1)(IN3)^{\wedge} + (IN1)(IN2) + (IN0) \sim (IN3) \sim (IN4)$  is specified through the registers PMSRC ([Table 145](#)) and PMCFG ([Table 146](#)). Each term in the boolean expression,  $(IN0)$ ,  $\sim (IN1)$ ,  $(IN3)^{\wedge}$ , etc., represents one bit slice of the pattern match engine.

- In the first minterm  $(IN0) \sim (IN1)(IN3)^{\wedge}$ , 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  $(IN0) \sim (IN3) \sim (IN4)$ , bit slice 5 monitors input  $(IN0)$  for a high level, bit slice 6 monitors input  $(IN3)$  for a low level, and bit slice 7 monitors input  $(IN4)$  for a low level. If this combination is detected, the interrupt associated with bit slice 7 (PININT7\_IRQ) will be asserted.

- The ORed result of all three minterms asserts the RXEV request to the CPU and the GPIO\_INT\_BMAT output. That is, if any of the three minterms are true, the output is asserted.

Related links:

[Section 11.7.2](#)

## 11.6 Register description

**Table 133. 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	<a href="#">Table 134</a>
IENR	R/W	0x004	Pin interrupt level or rising edge interrupt enable register	0	<a href="#">Table 135</a>
SIENR	WO	0x008	Pin interrupt level or rising edge interrupt set register	NA	<a href="#">Table 136</a>
CIENR	WO	0x00C	Pin interrupt level (rising edge interrupt) clear register	NA	<a href="#">Table 137</a>
IENF	R/W	0x010	Pin interrupt active level or falling edge interrupt enable register	0	<a href="#">Table 138</a>
SIENF	WO	0x014	Pin interrupt active level or falling edge interrupt set register	NA	<a href="#">Table 139</a>
CIENF	WO	0x018	Pin interrupt active level or falling edge interrupt clear register	NA	<a href="#">Table 140</a>
RISE	R/W	0x01C	Pin interrupt rising edge register	0	<a href="#">Table 141</a>
FALL	R/W	0x020	Pin interrupt falling edge register	0	<a href="#">Table 142</a>
IST	R/W	0x024	Pin interrupt status register	0	<a href="#">Table 143</a>
PMCTRL	R/W	0x028	Pattern match interrupt control register	0	<a href="#">Table 144</a>
PMSRC	R/W	0x02C	Pattern match interrupt bit-slice source register	0	<a href="#">Table 145</a>
PMCFG	R/W	0x030	Pattern match interrupt bit slice configuration register	0	<a href="#">Table 146</a>

### 11.6.1 Pin interrupt mode register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), one bit in the ISEL register determines whether the interrupt is edge or level sensitive.

**Table 134. 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.	-	-

### 11.6.2 Pin interrupt level or rising edge interrupt enable register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), 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 135. 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.	-	-

### 11.6.3 Pin interrupt level or rising edge interrupt set register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), 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 136. 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.	-	-

### 11.6.4 Pin interrupt level or rising edge interrupt clear register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), one bit in the CIENR register clears the corresponding bit in the IENR register depending on the pin interrupt mode configured in the ISEL register:

- If the pin interrupt mode is edge sensitive (PMODE = 0), the rising edge interrupt is cleared.
- If the pin interrupt mode is level sensitive (PMODE = 1), the level interrupt is cleared.

**Table 137. 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.	-	-

### 11.6.5 Pin interrupt active level or falling edge interrupt enable register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), 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 138. 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.	-	-

### 11.6.6 Pin interrupt active level or falling edge interrupt set register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), 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 139. 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.	-	-

### 11.6.7 Pin interrupt active level or falling edge interrupt clear register

For each of the 8 pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)), 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 140. 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.	-	-

### 11.6.8 Pin interrupt rising edge register

This register contains ones for pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)) 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 141. 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.	-	-



### 11.6.9 Pin interrupt falling edge register

This register contains ones for pin interrupts selected in the PINTSELn registers (see [Section 6.6.23](#)) 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 142. 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.	-	-

### 11.6.10 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 143. 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.	-	-

### 11.6.11 Pattern Match Interrupt Control Register

The pattern match control register contains one bit to select pattern-match interrupt generation (as opposed to pin interrupts which share the same interrupt request lines), and another to enable the RXEV output to the cpu. This register also allows the current state of any pattern matches to be read.

If the pattern match feature is not used (either for interrupt generation or for RXEV assertion) bits SEL\_PMATCH and ENA\_RXEV of this register should be left at 0 to conserve power.

**Remark:** Set up the pattern-match configuration in the PMSRC and PMCFG registers before writing to this register to enable (or re-enable) the pattern-match functionality. This eliminates the possibility of spurious interrupts as the feature is being enabled.

**Table 144. 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

### 11.6.12 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 6.6.23](#). 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 145. 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 145. 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 145. 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 145. 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 145. 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.	

### 11.6.13 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\_ENPTS<sub>n</sub> 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 146. 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 146. 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 146. 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 146. 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 146. 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 146. 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.	

## 11.7 Functional description

### 11.7.1 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 147](#).

Table 147. 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.

### 11.7.2 Pattern Match engine example

Suppose the desired boolean pattern to be matched is:

$$(IN1) + (IN1 * IN2) + (\sim IN2 * \sim IN3 * IN6fe) + (IN5 * IN7ev)$$

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 145](#)):
  - 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 146](#)):
  - 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 144](#)):

- 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.

### 11.7.3 Pattern match engine edge detect examples

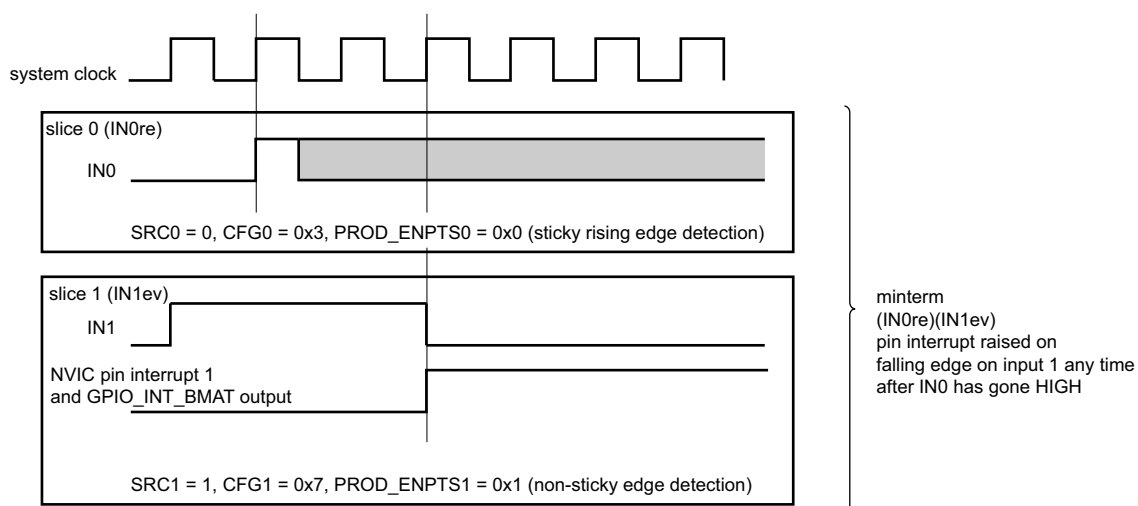


Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

**Fig 17. Pattern match engine examples: sticky edge detect**

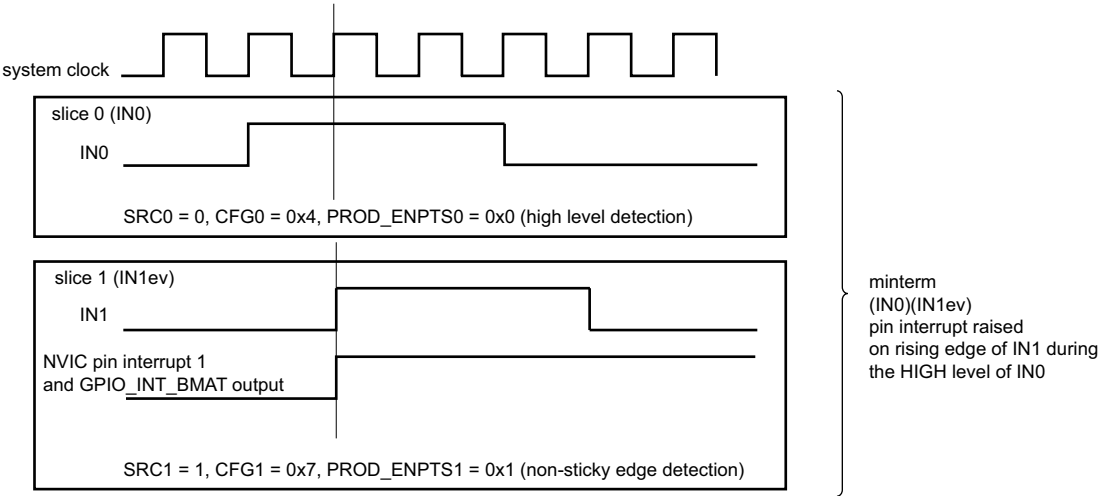


Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

Fig 18. Pattern match engine examples: Windowed non-sticky edge detect evaluates as true

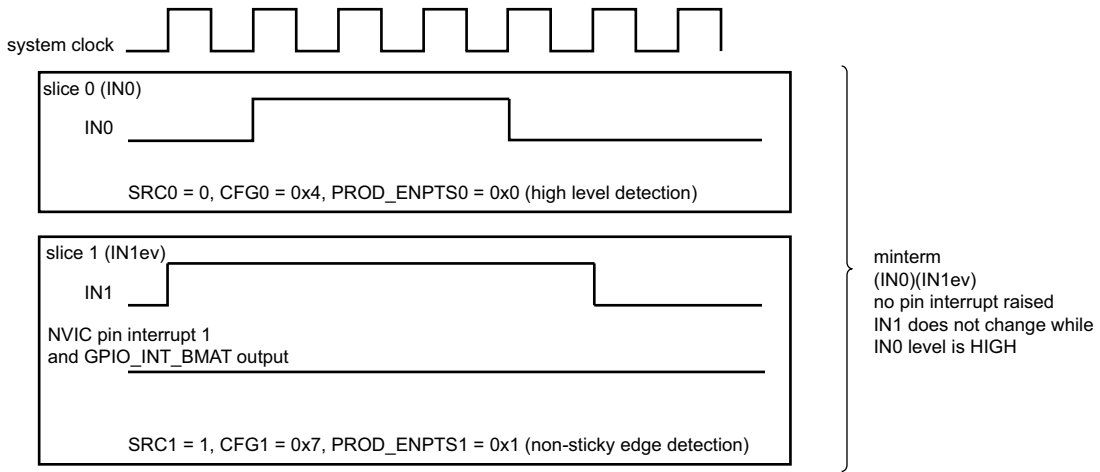


Figure shows pattern match functionality only and accurate timing is not implied. Inputs (INn) are shown synchronized to the system clock for simplicity.

Fig 19. Pattern match engine examples: Windowed non-sticky edge detect evaluates as false

### 12.1 How to read this chapter

---

This chapter provides an overview of power related information about LPC802 devices and allows the user to configure the reduced power modes deep-sleep mode, power-down mode, and deep power-down mode.

To turn analog components on or off in active and sleep modes, use the PDRUNCFG register ([Section 6.6.28 “Power configuration register”](#)). PDSLEEPCFG register can be used to turn analog peripherals on or off for deep-sleep mode and power-down modes.

### 12.2 Features

---

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

### 12.3 Basic configuration

---

#### 12.3.1 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](#).

##### 12.3.1.1 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, in combination with the PMU PCON register ([Table 151](#)), selects the sleep/power-down mode for the Arm core and the entire system. The SLEEPDEEP bit selects between: 0 = sleep mode, that stops the Arm core clock; 1 = deep-sleep mode, that stops the system clock.



Table 148. 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 <code>SEV</code> instruction.	0
31:5	-	Reserved.	0

## 12.4 Pin description

A few `WAKEUP` pins can be used to wake up only from deep power-down mode, if enabled. `PIO0_15`, `PIO0_9`, `PIO0_8`, `PIO0_17`, `PIO0_13`, `PIO0_4`, `PIO0_11`, `PIO0_10` are pins that have `WAKEUP` functions. Select the `WAKEUP` function in the `WUENAREG` register.

**Remark:** When entering deep power-down mode, an external pull-up resistor is required on the `WAKEUP` pin to hold it HIGH if this pin is enabled for the `WAKEUP` function.

## 12.5 General description

Power on the LPC802 is controlled by the PMU, 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. The peripherals receive no internal clocks. The internal 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.

- a. In deep-sleep mode, the flash is in standby mode.
- b. In power-down mode, the flash memory is powered down.

3. Deep power-down mode:

For maximal power savings, the entire system is shut down except for the general purpose registers in the PMU. Only the general purpose registers in the PMU maintain their internal states. The part can wake up on a pulse on one of the WAKEUP pins. On wake-up, the part reboots.

**Remark:** The part is in active mode when it is fully powered and operational after booting.

### 12.5.1 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 81 “Wake-up configuration register \(PDAWAKECFG, address 0x4004 8234\) bit description”](#).
- To use external interrupts for wake-up: [Table 78 “Start logic 0 pin wake-up enable register 0 \(STARTERP0, address 0x4004 8204\) bit description”](#) and [Table 77 “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 79 “Start logic 1 interrupt wake-up enable register \(STARTERP1, address 0x4004 8214\) bit description”](#)
- To configure the USART to wake up the part: [Section 13.3.2 “Configure the USART for wake-up”](#)
- For configuring the self-wake-up timer: [Section 18.5](#)
- For a list of all wake-up sources: [Table 149 “Wake-up sources for reduced power modes”](#)

Table 149. Wake-up sources for reduced power modes

power mode	Wake-up source	Conditions
Sleep	Any interrupt	Enable interrupt in NVIC.
Deep-sleep and power-down	Pin interrupts	Enable pin interrupts in NVIC and STARTERP0 registers.
	BOD interrupt	<ul style="list-style-type: none"> <li>• Enable interrupt in NVIC and STARTERP1 registers.</li> <li>• Enable interrupt in BODCTRL register.</li> <li>• BOD powered in PDSLEEPCFG register.</li> </ul>
	BOD reset	<ul style="list-style-type: none"> <li>• Enable reset in BODCTRL register.</li> <li>• BOD powered in PDSLEEPCFG register.</li> </ul>
	WWDT interrupt	<ul style="list-style-type: none"> <li>• Enable interrupt in NVIC and STARTERP1 registers.</li> <li>• WWDT running. Enable WWDT in WWDT MOD register and feed.</li> <li>• Enable interrupt in WWDT MOD register.</li> <li>• LPOSC powered in PDSLEEPCFG register.</li> </ul>
	WWDT reset	<ul style="list-style-type: none"> <li>• WWDT running.</li> <li>• Enable reset in WWDT MOD register.</li> <li>• LPOSC powered in PDSLEEPCFG register.</li> </ul>
	Self-Wake-up Timer (WKT) time-out	<ul style="list-style-type: none"> <li>• Enable interrupt in NVIC and STARTERP1 registers.</li> <li>• Enable low-power oscillator in the LPOSCCLKEN register in the SYSCON block.</li> <li>• Select low-power clock for WKT clock in the WKT CTRL register.</li> <li>• Start the WKT by writing a time-out value to the WKT COUNT register.</li> </ul>
Deep power-down	Interrupt from USART/SPI/I2C peripheral	<ul style="list-style-type: none"> <li>• Enable interrupt in NVIC and STARTERP1 registers.</li> <li>• Enable USART/I2C/SPI interrupts.</li> <li>• Provide an external clock signal to the peripheral.</li> <li>• Configure the USART in synchronous slave mode and I2C and SPI in slave mode.</li> </ul>
	WAKEUP pins	Enable the WAKEUP function in the WUENAREG register in the PMU.

## 12.6 Register description

Table 150. Register overview: PMU (base address 0x4002 0000)

Name	Access	Address offset	Description	Reset value	Reference
PCON	R/W	0x000	Power control register	0x0	<a href="#">Table 151</a>
GPREG0	R/W	0x004	General purpose register 0	0x0	<a href="#">Table 152</a>
GPREG1	R/W	0x008	General purpose register 1	0x0	<a href="#">Table 152</a>
GPREG2	R/W	0x00C	General purpose register 2	0x0	<a href="#">Table 152</a>
GPREG3	R/W	0x010	General purpose register 3	0x0	<a href="#">Table 152</a>
GPREG4	R/W	0x014	General purpose register 4	0x0	<a href="#">Table 152</a>
Deep power-down wake-up source status register	R/W	0x020	Pin wake-up status register	0x0	<a href="#">Table 153</a>
Deep power-down wake-up enable register	R/W	0x024	Pin wake-up enable register	0x0	<a href="#">Table 154</a>

### 12.6.1 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 151. Power control register (PCON, address 0x4002 0000) bit description

Bit	Symbol	Value	Description	Reset value
2:0	PM		Power mode	0
		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. Directly enter deep power-down mode.	
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
11	DPDFLAG		Deep power-down flag	0
		0	Not deep power-down. Read: deep power-down mode <b>not</b> 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

### 12.6.2 General purpose registers 0 to 4

The general purpose registers retain data through the deep power-down mode when power is still applied to the V<sub>DD</sub> 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 152. General purpose registers 0 to 4 (GPREG[0:4], address 0x4002 0004 (GPREG0) to 0x4002 0014 (GPREG4)) bit description

Bit	Symbol	Description	Reset value
31:0	GPDATA	Data retained during deep power-down mode.	0x0

### 12.6.3 Deep power-down wake-up source status register

The WUSRCREG register provides a status on the wake-up pin. The associated bit is set when the corresponding wake-up pin changes to LOW. When the wake-up pin changes to HIGH, writing a 1 to the corresponding bit clears the previous status. Bits should be cleared (writing 1) before deep power-down mode.

**Table 153. WUSRCREG, address 0x4002 0020 bit description**

Bit	Symbol	Bit	Description	Reset value
7:0	WUSRCREG		Pin wake-up status register.	0
	WUSRCREG0	0	PIO0_15	0
	WUSRCREG1	1	PIO0_9	0
	WUSRCREG2	2	PIO0_8	0
	WUSRCREG3	3	PIO0_17	0
	WUSRCREG4	4	PIO0_13	0
	WUSRCREG5	5	PIO0_4	0
	WUSRCREG6	6	PIO0_11	0
	WUSRCREG7	7	PIO0_10	0
31:8	-	-	Reserved	-

### 12.6.4 Deep power-down wake-up enable register

The WUENAREG register is used to enable the wake-up pins for deep power-down mode.

**Table 154. WUENAREG, address 0x4002 0024 bit description**

Bit	Symbol	Bit	Description	Reset value
7:0	WUENAREG		Pin wake-up enable register.	0
	WUENAREG0	0	PIO0_15	0
	WUENAREG1	1	PIO0_9	0
	WUENAREG2	2	PIO0_8	0
	WUENAREG3	3	PIO0_17	0
	WUENAREG4	4	PIO0_13	0
	WUENAREG5	5	PIO0_4	0
	WUENAREG6	6	PIO0_11	0
	WUENAREG7	7	PIO0_10	0
31:8	-	-	Reserved	-

## 12.7 Functional description

### 12.7.1 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 155. 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	standby	off	off
BOD	software configurable	software configurable	software configurable	off
LPOSC/WWDT	software configurable	software configurable	software configurable	off
Digital peripherals	software configurable	off	off	off
Wake-up buffers	software configurable (cannot be used as wake-up source)	software configurable (cannot be used as wake-up source)	software configurable (cannot be used as wake-up source)	software configurable

**Remark:** The Debug mode is not supported in sleep, deep-sleep, power-down, or deep power-down modes.

### 12.7.2 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 low power oscillator to be on independently of the deep-sleep and power-down mode software configuration through the PDSLEEPCFG register. For details see [Section 17.5.3 “Using the WWDT lock features”](#).

### 12.7.3 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.

### 12.7.3.1 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 63](#)).
- The power to various analog blocks (oscillators, the BOD circuit, and the flash block) can be controlled at any time individually through the PDRUNCFG register ([Table 82](#) “Power configuration register (PDRUNCFG, address 0x4004 8238) bit description”).
- The clock source for the system clock can be selected from the FRO (default), the external clock, or the low power oscillator (see [Figure 6](#) and related registers).
- The system clock frequency can be selected by the MAINCLKSEL ([Table 57](#)) and the SYSAHBCLKDIV register ([Table 59](#)).
- The ADC clock and CLKOUT use their own clock dividers. The clocks can be shut down through the corresponding clock divider registers. Select the "none" option in the clock select register to shut down the clock to the peripheral and its divider.
- The two USARTs, I<sup>2</sup>C and SPI share the same Fractional Rate Divider and its clock select multiplexer. Select the "none" option in the FRG0CLKSEL register to shut down the clock.

### 12.7.4 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.

#### 12.7.4.1 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.

#### 12.7.4.2 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 148](#)).
3. Use the Arm Cortex-M0+ Wait-For-Interrupt (WFI) instruction.

### 12.7.4.3 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.

### 12.7.5 Deep-sleep mode

In deep-sleep mode, the system clock to the processor is disabled as in sleep mode. All analog blocks are powered down, except for the BOD circuit and the low power oscillator, which can be selected or deselected during deep-sleep mode in the PDSLEEPCFG register. The main clock, and therefore all peripheral clocks, are disabled except for the clock to the watchdog timer if the low power oscillator is enabled. 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.

#### 12.7.5.1 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 80](#)) register:

- The low power 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.

#### 12.7.5.2 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 151](#)).
2. Select the power configuration in deep-sleep mode in the PDSLEEPCFG ([Table 80](#)) register.
3. Select the power configuration after wake-up in the PDAWAKECFG ([Table 81](#)) 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 78](#), [Table 79](#)) and in the NVIC.
5. Select the FRO as the main clock.
6. Write one to the SLEEPDEEP bit in the Arm Cortex-M0+ SCR register ([Table 148](#)).
7. Use the Arm WFI instruction.

#### 12.7.5.3 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 77](#). Each pin interrupt must also be enabled in the STARTERP0 register ([Table 78](#)) 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 79](#)). 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 73](#)).
- WWDT signal, if the low power oscillator is enabled in the PDSLEEPCFG register:
  - WWDT interrupt using the interrupt wake-up register 1 ([Table 79](#)). 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 low power oscillator must be running in deep-sleep mode (see PDSLEEPCFG register), and the WDT must be enabled in the SYSAHBCLKCTRL register.
- Via any of the USART blocks if the USART is configured in synchronous mode. See [Section 13.3.2 “Configure the USART for wake-up”](#).
- Via the I<sup>2</sup>C. See [Section 15.3.3](#).
- Via the SPI. See [Section 14.3.1](#).

**Remark:** To disable the BOD in deep sleep mode if enabled in active mode, disable BOD reset (bit 4 in the BODCTRL register) before entering power-down mode. After wake-up, enable BOD reset (bit 4 in the BODCTRL register).

## 12.7.6 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 low power oscillator, which must be selected or deselected during power-down mode in the PDSLEEPCFG register. The main clock and therefore all peripheral clocks are disabled except for the clock to the watchdog timer if the low power oscillator is enabled. 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.

### 12.7.6.1 Power configuration in power-down mode

Power consumption in power-down mode can be configured by the power configuration setting in the PDSLEEPCFG ([Table 80](#)) register in the same way as for deep-sleep mode (see [Section 12.7.5.1](#)):

- The low power oscillator can be left running in power-down mode if required for the WWDT.
- The BOD circuit can be left running in power-down mode if required by the application.

### 12.7.6.2 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 151](#)).
2. Select the power configuration in power-down mode in the PDSLEEPCFG ([Table 80](#)) register.
3. Select the power configuration after wake-up in the PDAWAKECFG ([Table 81](#)) 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 78](#), [Table 79](#)) and in the NVIC.
5. Select the FRO as the main clock.
6. Write one to the SLEEPDEEP bit in the Arm Cortex-M0+ SCR register ([Table 148](#)).
7. Use the Arm WFI instruction.

### 12.7.6.3 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 77](#). Each pin interrupt must also be enabled in the STARTERP0 register ([Table 78](#)) 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 79](#)). 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 73](#)).
- WWDT signal, if the low power oscillator is enabled in the PDSLEEPCFG register:
  - WWDT interrupt using the interrupt wake-up register 1 ([Table 79](#)). 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 enable must be set in the WWDT MOD register.
  - Via any of the USART blocks. See [Section 13.3.2 “Configure the USART for wake-up”](#).
  - Via the I<sup>2</sup>C. See [Section 15.3.3](#).
  - Via the SPI. See [Section 14.3.1](#).

**Remark:** If the BOD is enabled in active mode and the user needs to disable the BOD in power-down mode, disable the BOD reset (bit 4 in the BODCTRL register) before entering power-down mode. After wake-up, enable the BOD reset (bit 4 in the BODCTRL register).

## 12.7.7 Deep power-down mode

In deep power-down mode, power and clocks are shut off to the entire chip with the exception of the  $\overline{\text{WAKEUP}}$  pins and general purpose registers. Five general-purpose registers are available to store information during Deep power-down mode

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  $\overline{\text{WAKEUP}}$  pin.

**Remark:** Setting bit 3 in the PCON register ([Table 151](#)) prevents the part from entering deep-power down mode.

#### 12.7.7.1 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  $\overline{\text{WAKEUP}}$  pins are powered.

#### 12.7.7.2 Programming deep power-down mode using the $\overline{\text{WAKEUP}}$ pin:

The following steps must be performed to enter deep power-down mode when using the  $\overline{\text{WAKEUP}}$  pin for waking up:

1. Pull the  $\overline{\text{WAKEUP}}$  pin externally HIGH.
2. Ensure that bit 3 in the PCON register ([Table 151](#)) is cleared.
3. Write 0x3 to the PM bits in the PCON register (see [Table 151](#)).
4. Store data to be retained in the general purpose registers ([Section 12.6.2](#)).
5. Write one to the SLEEPDEEP bit in the Arm Cortex-M0+ SCR register ([Table 148](#)).
6. Use the Arm WFI instruction.

#### 12.7.7.3 Wake-up from deep power-down mode using the $\overline{\text{WAKEUP}}$ pin:

Pulling the  $\overline{\text{WAKEUP}}$  pins LOW wakes up the LPC802 from deep power-down, and the part goes through the entire reset process.

1. On the  $\overline{\text{WAKEUP}}$  pins, 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 GPREG0 to GPREG4 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 151](#)) 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 151](#)).
4. (Optional) Read the stored data in the general purpose registers ([Section 12.6.2](#)).
5. Set up the PMU for the next deep power-down cycle.

### 13.1 How to read this chapter

---

Two USARTs are available on all parts depending on the switch matrix configuration. UART1 does not support RTS/CTS hardware signaling for automatic flow control.

### 13.2 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.

### 13.3 Basic configuration

---

Configure the USARTs for receiving and transmitting data:

- In the SYSAHBCLKCTRL register, set bit 14 and 15 ([Table 63](#)) to enable the clock to the register interface.
- Clear the USART peripheral resets using the PRESETCTRL register. See [Table 64](#).
- Enable or disable the USART interrupts in slots #3 and #4 in the NVIC. See [Table 38](#).
- Configure the USART pin functions through the switch matrix.
- Configure the USART clock and baud rate. See [Section 13.3.1](#).

For wake-up from deep-sleep and power-down modes the USART must be configured in synchronous mode. See [Section 13.3.2](#) for details.

### 13.3.1 Configure the USART clock and baud rate

Two USARTs have a shared clock selection that include a shared fractional divider. The fractional divider for the baud rate calculation is set up in the SYSCON block as follows (see [Figure 20](#)):

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

$$= (\text{FRGINPUTCLK}) / (1 + (\text{MULT}/\text{DIV}))$$

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.

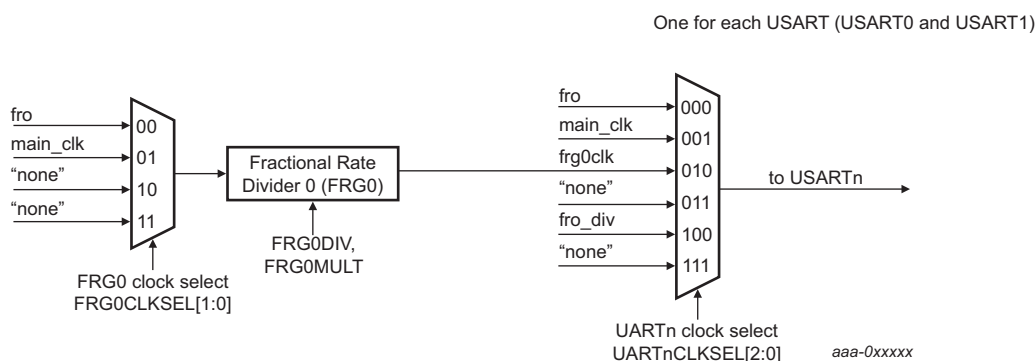
$$\text{baud rate} = \text{FCLK} / ((\text{OSRVAL} + 1) \times (\text{BRGVAL} + 1)).$$

$$\text{BRGVAL} = \text{FCLK} / ((\text{OSRVAL} + 1) \times \text{baud rate}) - 1$$

(assumes FCLK ≥ oversample rate × baud rate)

[Section 13.6.9 “USART Baud Rate Generator register”](#)

3. In synchronous mode: The serial clock is Un\_SCLK = FCLK/(BRGVAL+1).



See [Figure 7 “LPC802 clock generation \(continued\)”](#).

**Fig 20. USART clocking**

For details on the clock configuration see:

### [Section 13.7.1 “Clocking and baud rates”](#)

## 13.3.2 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 have two options for configuring USART for wake-up:

- If the USART is configured for 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.

### 13.3.2.1 Wake-up from sleep mode

- Configure the USART in either asynchronous mode or synchronous mode. See [Table 158](#).
- 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 161](#)).

### 13.3.2.2 Wake-up from deep-sleep or power-down mode

- Configure the USART in synchronous slave mode. See [Table 158](#). 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 79 “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 161](#)), you can customize when the wake-up occurs in the USART receive/transmit protocol.

## 13.4 Pin description

Table 156. USART pin description

Function	I/O	Type	Connect to	Use register	Reference	Description
U0_TXD	O	external to pin	any pin	PINASSIGN0	<a href="#">Table 89</a>	Transmitter output for USART0. Serial transmit data.
U0_RXD	I	external to pin	any pin	PINASSIGN0	<a href="#">Table 89</a>	Receiver input for USART0.
U0_RTS	O	external to pin	any pin	PINASSIGN0	<a href="#">Table 89</a>	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.
U0_CTS	I	external to pin	any pin	PINASSIGN0	<a href="#">Table 89</a>	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	<a href="#">Table 90</a>	Serial clock input/output for USART0 in synchronous mode.
U1_TXD	O	external to pin	any pin	PINASSIGN1	<a href="#">Table 90</a>	Transmitter output for USART1. Serial transmit data.
U1_RXD	I	external to pin	any pin	PINASSIGN1	<a href="#">Table 90</a>	Receiver input for USART1.
U1_SCLK	I/O	external to pin	any pin	PINASSIGN1	<a href="#">Table 90</a>	Serial clock input/output for USART1 in synchronous mode.

## 13.5 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.

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).

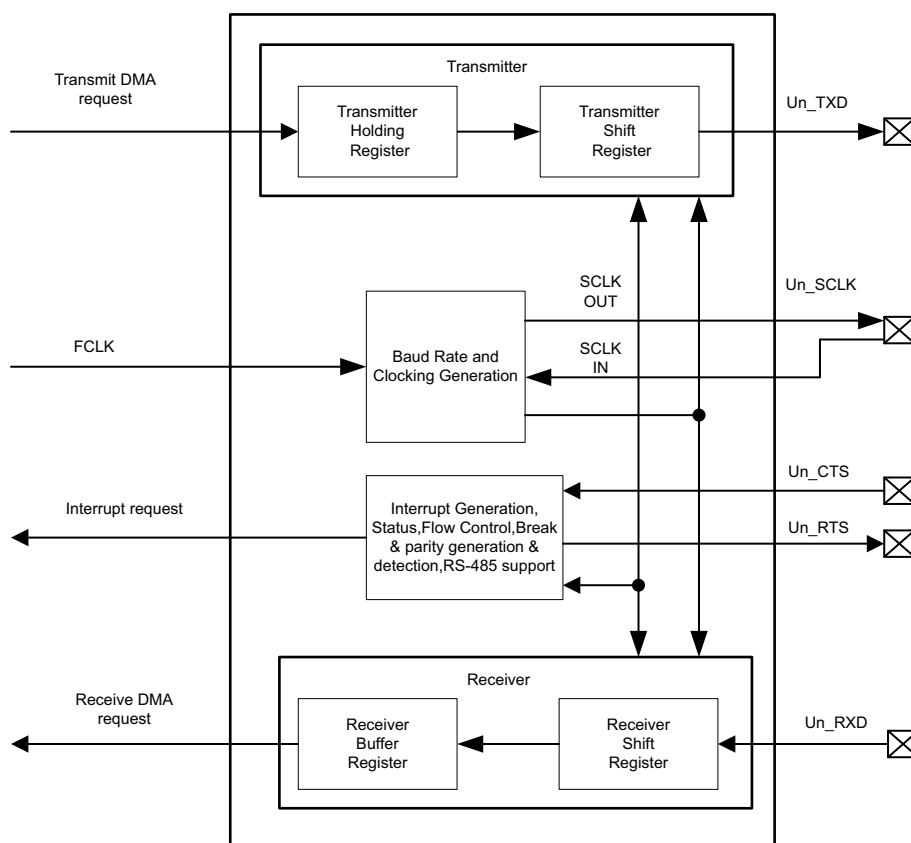
The USART transmitter block accepts data written by the CPU and buffers the data in the transmit holding register. When the transmitter is available, the transmit shift register takes that data, formats it, and serializes it to the serial output, Un\_TXD.

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

$$U\_PCLK = FRGCLKDIV / (1 + MULT / DIV)$$

Fig 21. USART block diagram



## 13.6 Register description

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

**Table 157. Register overview: USART (base address 0x4006 4000 (USART0) and 0x4006 8000 (USART1))**

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	<a href="#">Table 158</a>
CTL	R/W	0x004	USART Control register. USART control settings that are more likely to change during operation.	0	<a href="#">Table 159</a>
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	<a href="#">Table 160</a>
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	<a href="#">Table 161</a>
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.	-	<a href="#">Table 162</a>
RXDAT	R	0x014	Receiver Data register. Contains the last character received.	-	<a href="#">Table 163</a>
RXDATSTAT	R	0x018	Receiver Data with Status register. Combines the last character received with the current USART receive status. Allows software to recover incoming data and status together.	-	<a href="#">Table 164</a>
TXDAT	R/W	0x01C	Transmit Data register. Data to be transmitted is written here.	0	<a href="#">Table 165</a>
BRG	R/W	0x020	Baud Rate Generator register. 16-bit integer baud rate divisor value.	0	<a href="#">Table 166</a>
INTSTAT	R	0x024	Interrupt status register. Reflects interrupts that are currently enabled.	0x0005	<a href="#">Table 167</a>
OSR	R/W	0x028	Oversample selection register for asynchronous communication.	0xF	<a href="#">Table 168</a>
ADDR	R/W	0x02C	Address register for automatic address matching.	0	<a href="#">Table 169</a>

### 13.6.1 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 158. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1)) 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 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) because the transmitter has been reset and is therefore available.	
		1	Enabled. The USART is enabled for operation.	
1	-		Reserved. Read value is undefined, only zero should be written.	NA
3:2	DATALEN		Selects the data size for the USART.	00
		0x0	7 bit Data length.	
		0x1	8 bit Data length.	
		0x2	9 bit data length. The 9th bit is commonly used for addressing in multidrop mode. See the ADDRDET bit in the CTL register.	
		0x3	Reserved.	
5:4	PARITYSEL		Selects what type of parity is used by the USART.	00
		0x0	No parity.	
		0x1	Reserved.	
		0x2	Even parity. Adds a bit to each character such that the number of 1s in a transmitted character is even, and the number of 1s in a received character is expected to be even.	
		0x3	Odd parity. Adds a bit to each character such that the number of 1s in a transmitted character is odd, and the number of 1s in a received character is expected to be odd.	
6	STOPLEN		Number of stop bits appended to transmitted data. Only a single stop bit is required for received data.	0
		0	1 stop bit.	
		1	2 stop bits. This setting should only be used for asynchronous communication.	
8:7	-		Reserved. Read value is undefined, only zero should be written.	NA

**Table 158. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1)) bit description ...continued**

Bit	Symbol	Value	Description	Reset Value
9	CTSEN		CTS Enable. Determines whether CTS is used for flow control. CTS can be from the input pin, or from the USART's own RTS if loopback mode is enabled.	0
		0	No flow control. The transmitter does not receive any automatic flow control signal.	
		1	Flow control enabled. The transmitter uses the CTS input (or RTS output in loopback mode) for flow control purposes.	
10	-		Reserved. Read value is undefined, only zero should be written.	NA
11	SYNCEN		Selects synchronous or asynchronous operation.	0
		0	Asynchronous mode is selected.	
		1	Synchronous mode is selected.	
12	CLKPOL		Selects the clock polarity and sampling edge of received data in synchronous mode.	0
		0	Falling edge. Un_RXD is sampled on the falling edge of SCLK.	
		1	Rising edge. Un_RXD is sampled on the rising edge of SCLK.	
13	-		Reserved. Read value is undefined, only zero should be written.	NA
14	SYNCMST		Synchronous mode Master select.	0
		0	Slave. When synchronous mode is enabled, the USART is a slave.	
		1	Master. When synchronous mode is enabled, the USART is a master.	
15	LOOP		Selects data loopback mode.	0
		0	Normal operation.	
		1	Loopback mode. This provides a mechanism to perform diagnostic loopback testing for USART data. Serial data from the transmitter (Un_TXD) is connected internally to serial input of the receive (Un_RXD). Un_TXD and Un_RTS activity will also appear on external pins if these functions are configured to appear on device pins. The receiver RTS signal is also looped back to CTS and performs flow control if enabled by CTSEN.	
17:16	-		Reserved. Read value is undefined, only zero should be written.	NA
18	OETA		Output Enable Turnaround time enable for RS-485 operation.	0
		0	De-asserted. If selected by OESEL, the Output Enable signal de-asserted at the end of the last stop bit of a transmission.	
		1	Asserted. If selected by OESEL, the Output Enable signal remains asserted for 1 character time after then end the last stop bit of a transmission. OE will also remain asserted if another transmit begins before it is de-asserted.	
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.	

**Table 158. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1)) bit description ...continued**

Bit	Symbol	Value	Description	Reset Value
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

### 13.6.2 USART Control register

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

**Table 159. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1)) 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.	

Table 159. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1)) bit description

Bit	Symbol	Value	Description	Reset value
2	ADDRDET		Enable address detect mode.	0
		0	Disabled. The USART presents all incoming data.	
		1	Enabled. The USART receiver ignores incoming data that does not have the most significant bit of the data (typically the 9th bit) = 1. When the data MSB bit = 1, the receiver treats the incoming data normally, generating a received data interrupt. Software can then check the data to see if this is an address that should be handled. If it is, the ADDRDET bit is cleared by software and further incoming data is handled normally.	
5:3	-		Reserved. Read value is undefined, only zero should be written.	NA
6	TXDIS		Transmit Disable.	0
		0	Not disabled. USART transmitter is not disabled.	
		1	Disabled. USART transmitter is disabled after any character currently being transmitted is complete. This feature can be used to facilitate software flow control.	
7	-		Reserved. Read value is undefined, only zero should be written.	NA
8	CC		Continuous Clock generation. By default, SCLK is only output while data is being transmitted in synchronous mode.	0
		0	Clock on character. In synchronous mode, SCLK cycles only when characters are being sent on Un_TXD or to complete a character that is being received.	
		1	Continuous clock. SCLK runs continuously in synchronous mode, allowing characters to be received on Un_RxD independently from transmission on Un_TXD).	
9	CLRCCONRX		Clear Continuous Clock.	0
		0	No effect on the CC bit.	
		1	Auto-clear. The CC bit is automatically cleared when a complete character has been received. This bit is cleared at the same time.	
15:10	-		Reserved. Read value is undefined, only zero should be written.	NA
16	AUTOBAUD		Autobaud enable.	0
		0	Disabled. UART is in normal operating mode.	
		1	Enabled. UART is in autobaud mode. This bit should only be set when the UART is enabled in the CFG register and the UART receiver is idle. The first start bit of RX is measured and used to 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

### 13.6.3 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 162](#)).

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 160. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1)) bit description**

Bit	Symbol	Description	Reset value	Access <a href="#">[1]</a>
0	RXRDY	Receiver Ready flag. When 1, indicates that data is available to be read from the receiver buffer. Cleared after a read of the RXDAT or RXDATSTAT registers.	0	RO
1	RXIDLE	Receiver Idle. When 0, indicates that the receiver is currently in the process of receiving data. When 1, indicates that the receiver is not currently in the process of receiving data.	1	RO
2	TXRDY	Transmitter Ready flag. When 1, this bit indicates that data may be written to the transmit buffer. Previous data may still be in the process of being transmitted. Cleared when data is written to TXDAT. Set when the data is moved from the transmit buffer to the transmit shift register.	1	RO
3	TXIDLE	Transmitter Idle. When 0, indicates that the transmitter is currently in the process of sending data. When 1, indicate that the transmitter is not currently in the process of sending data.	1	RO
4	CTS	This bit reflects the current state of the CTS signal, regardless of the setting of the CTSEN bit in the CFG register. This will be the value of the CTS input pin unless loopback mode is enabled.	NA	RO
5	DELTACTS	This bit is set when a change in the state is detected for the CTS flag above. This bit is cleared by software.	0	W1
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

Table 160. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1)) bit description

Bit	Symbol	Description	Reset value	Access [1]
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.

### 13.6.4 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 161. USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1)) 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



**Table 161. USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1)) bit description ...continued**

Bit	Symbol	Description	Reset Value
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

### 13.6.5 USART Interrupt Enable Clear register

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

**Table 162. USART Interrupt Enable clear register (INTENCLR, address 0x4006 4010 (USART0), 0x4006 8010 (USART1)) 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

### 13.6.6 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 163. USART Receiver Data register (RXDAT, address 0x4006 4014 (USART0), 0x4006 8014 (USART1)) 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



### 13.6.7 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.

**Remark:** Reading this register changes the status flags.

**Table 164. USART Receiver Data with Status register (RXDATSTAT, address 0x4006 4018 (USART0), 0x4006 8018 (USART1)) 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 <a href="#">Table 160</a> .	0
31:16	-	Reserved, the value read from a reserved bit is not defined.	NA

### 13.6.8 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 165. USART Transmitter Data Register (TXDAT, address 0x4006 401C (USART0), 0x4006 801C (USART1)) 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

### 13.6.9 USART Baud Rate Generator register

The Baud Rate Generator is a simple 16-bit integer divider controlled by the BRG register. The BRG register contains the value used to divide the base clock in order to produce the clock used for USART internal operations.

A 16-bit value allows producing standard baud rates from 300 baud and lower at the highest frequency of the device, up to 921,600 baud from a base clock as low as 14.7456 MHz.

Typically, the baud rate clock is 16 times the actual baud rate. This overclocking allows for centering the data sampling time within a bit cell, and for noise reduction and detection by taking three samples of incoming data.

Details on how to select the right values for BRG can be found in [Section 13.7.1](#).

**Remark:** If software needs to change the baud rate, the following sequence should be used: 1) Make sure the USART is not currently sending or receiving data. 2) Disable the USART by writing a 0 to the Enable bit (0 may be written to the entire registers). 3) Write the new BRGVAL. 4) Write to the CFG register to set the Enable bit to 1.

**Table 166. USART Baud Rate Generator register (BRG, address 0x4006 4020 (USART0), 0x4006 8020 (USART1)) 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

### 13.6.10 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 160](#) for detailed descriptions of the interrupt flags.

**Table 167. USART Interrupt Status register (INTSTAT, address 0x4006 4024 (USART0), 0x4006 8024 (USART1)) 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

### 13.6.11 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 168. USART Oversample selection register (OSR, address 0x4006 4028 (USART0), 0x4006 4028 (USART1)) 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

### 13.6.12 USART Address register

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

**Table 169. USART Address register (ADDR, address 0x4006 402C (USART0), 0x4006 802C (USART1)) 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

## 13.7 Functional description

### 13.7.1 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 20](#).

#### 13.7.1.1 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  $1 + (\text{MULT} / 256)$ , 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.

For details see [Section 13.3.1 “Configure the USART clock and baud rate”](#).

#### 13.7.1.2 Baud Rate Generator (BRG)

The Baud Rate Generator (see [Section 13.6.9](#)) 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.

#### 13.7.1.3 Baud rate calculations

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

### 13.7.2 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.

### 13.7.3 Flow control

The USART supports both hardware and software flow control.

### 13.7.3.1 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 22](#) shows an overview of RTS and CTS within the USART.

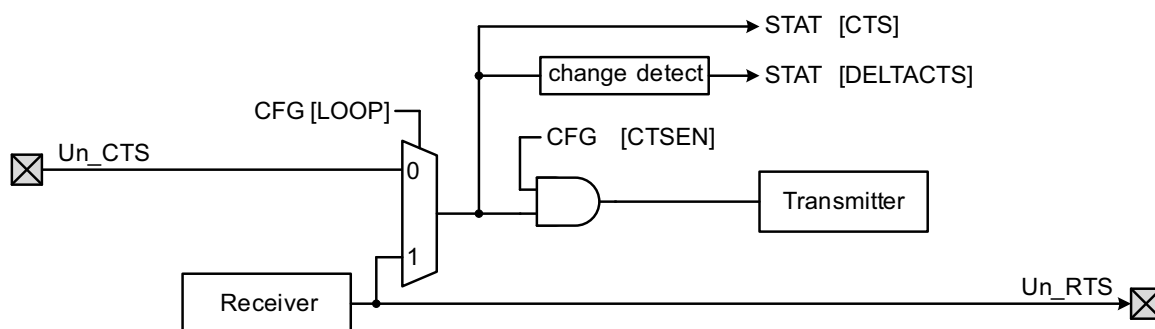


Fig 22. Hardware flow control using RTS and CTS

### 13.7.3.2 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).

### 13.7.4 Autobaud function

The autobaud function 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.

### 13.7.5 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 13.6.1](#) and the ADDR register in [Section 13.6.12](#)), as well as software address recognition (see the ADDRDET bit in the CTL register in [Section 13.6.2](#)).

Automatic data direction control with the RTS pin can be set up using the OESEL, OEPOL, and OETA bits in the CFG register ([Section 13.6.1](#)). Data direction control can also be implemented in software using a GPIO pin.

### 13.7.6 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.

### 14.1 How to read this chapter

The SPI interface is available on all parts.

### 14.2 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 two Slave Select input/outputs with selectable polarity and flexible usage.

**Remark:** Texas Instruments SSI and National Microwire modes are not supported.

### 14.3 Basic configuration

Configure SPI using the following registers:

- In the SYSAHBCLKCTRL register, set bit 11 ([Table 63](#)) to enable the clock to the register interface.
- Clear the SPI peripheral resets using the PRESETCTRL register ([Table 64](#)).
- Enable/disable the SPI interrupts in interrupt slot #0 in the NVIC.
- Configure the SPI pin functions through the switch matrix. See [Section 14.4](#).
- Using the SPI0CLKSEL register, the peripheral clock sources for the SPI can be FRO, main\_clk, frg0clk, and fro\_div. See [Figure 6 “LPC802 clock generation”](#).

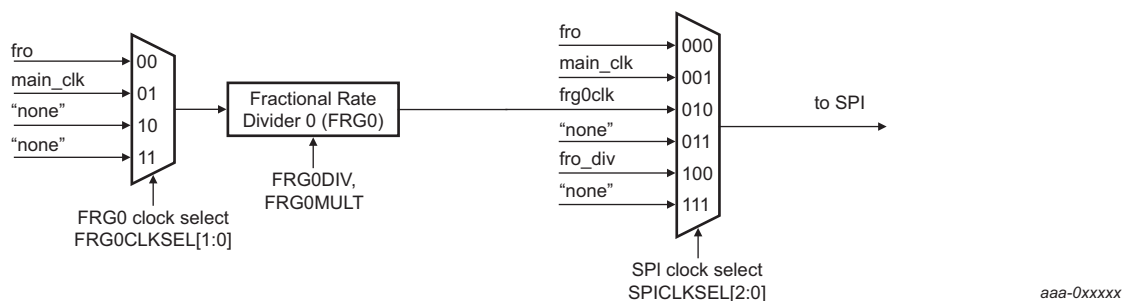


Fig 23. SPI clocking

### 14.3.1 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.

#### 14.3.1.1 Wake-up from sleep mode

- Configure the SPI in either master or slave mode. See [Table 172](#).
- 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 175](#)).

#### 14.3.1.2 Wake up from deep-sleep or power-down mode

- Configure the SPI in slave mode. See [Table 172](#). 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 79 “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 175](#)). Examples are the following wake-up events:
  - A change in the state of the SSEL pins.
  - Data available to be received.
  - Receiver overrun.

## 14.4 Pin description

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

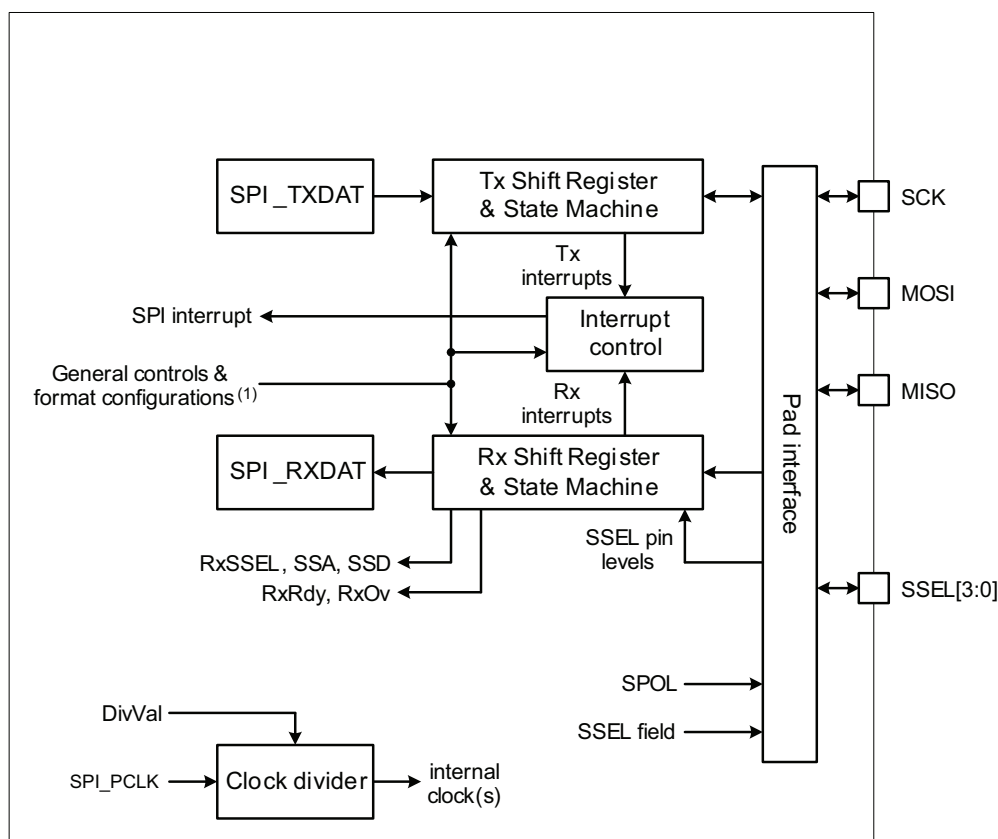
See [Section 8.3.1 “Connect an internal signal to a package pin”](#) to assign the SPI functions to pins on the part.



Table 170. SPI Pin Description

Function	I/O	Type	Connect to	Use register	Reference	Description
SPI0_SCK	I/O	external to pin	any pin	PINASSIGN2	<a href="#">Table 91</a>	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	PINASSIGN2	<a href="#">Table 91</a>	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	PINASSIGN2	<a href="#">Table 91</a>	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	PINASSIGN2	<a href="#">Table 91</a>	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	PINASSIGN3	<a href="#">Table 92</a>	Slave Select 1.

## 14.5 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 24. SPI block diagram

## 14.6 Register description

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

Table 171. Register overview: SPI (base address 0x4005 8000 (SPI))

Name	Access	Offset	Description	Reset value	Reference
CFG	R/W	0x000	SPI Configuration register	0	<a href="#">Table 172</a>
DLY	R/W	0x004	SPI Delay register	0	<a href="#">Table 173</a>
STAT	R/W	0x008	SPI Status. Some status flags can be cleared by writing a 1 to that bit position	0x0102	<a href="#">Table 174</a>
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	<a href="#">Table 175</a>

Table 171. Register overview: SPI (base address 0x4005 8000 (SPI))

Name	Access	Offset	Description	Reset value	Reference
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	<a href="#">Table 176</a>
RXDAT	R	0x014	SPI Receive Data	NA	<a href="#">Table 177</a>
TXDATCTL	R/W	0x018	SPI Transmit Data with Control	0	<a href="#">Table 178</a>
TXDAT	R/W	0x01C	SPI Transmit Data	0	<a href="#">Table 179</a>
TXCTL	R/W	0x020	SPI Transmit Control	0	<a href="#">Table 180</a>
DIV	R/W	0x024	SPI clock Divider	0	<a href="#">Table 181</a>
INTSTAT	R	0x028	SPI Interrupt Status	0x02	<a href="#">Table 182</a>

### 14.6.1 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 174](#)) 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 172. SPI Configuration register (CFG, addresses 0x4005 8000 (SPI)) 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).	
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

Table 172. SPI Configuration register (CFG, addresses 0x4005 8000 (SPI) bit description ...continued

Bit	Symbol	Value	Description	Reset value
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	SPOLO		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.	
31:10	-		Reserved. Read value is undefined, only zero should be written.	NA

### 14.6.2 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 14.7.2.1 “Pre\\_delay and Post\\_delay”](#)

[Section 14.7.2.2 “Frame\\_delay”](#)

[Section 14.7.2.3 “Transfer\\_delay”](#)

Table 173. SPI Delay register (DLY, addresses 0x4005 8004 (SPI) 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

### 14.6.3 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 174. SPI Status register (STAT, addresses 0x4005 8008 (SPI) bit description**

Bit	Symbol	Description	Reset value	Access <a href="#">[1]</a>
0	RXRDY	Receiver Ready flag. When 1, indicates that data is available to be read from the receiver buffer. Cleared after a read of the RXDAT register.	0	RO
1	TXRDY	Transmitter Ready flag. When 1, this bit indicates that data may be written to the transmit buffer. Previous data may still be in the process of being transmitted. Cleared when data is written to TXDAT or TXDATCTL until the data is moved to the transmit shift register.	1	RO
2	RXOV	Receiver Overrun interrupt flag. This flag applies only to slave mode (Master = 0). This flag is set when the beginning of a received character is detected while the receiver buffer is still in use. If this occurs, the receiver buffer contents are preserved, and the incoming data is lost. Data received by the SPI should be considered undefined if RxOv is set.	0	W1
3	TXUR	Transmitter Underrun interrupt flag. This flag applies only to slave mode (Master = 0). In this case, the transmitter must begin sending new data on the next input clock if the transmitter is idle. If that data is not available in the transmitter holding register at that point, there is no data to transmit and the TXUR flag is set. Data transmitted by the SPI should be considered undefined if TXUR is set.	0	W1
4	SSA	Slave Select Assert. This flag is set whenever any slave select transitions from deasserted to asserted, in both master and slave modes. This allows determining when the SPI transmit/receive functions become busy, and allows waking up the device from reduced power modes when a slave mode access begins. This flag is cleared by software.	0	W1
5	SSD	Slave Select Deassert. This flag is set whenever any asserted slave selects transition to deasserted, in both master and slave modes. This allows determining when the SPI transmit/receive functions become idle. This flag is cleared by software.	0	W1
6	STALLED	Stalled status flag. This indicates whether the SPI is currently in a stall condition.	0	RO

Table 174. SPI Status register (STAT, addresses 0x4005 8008 (SPI) 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.

#### 14.6.4 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 174](#) for details of the interrupts.

Table 175. SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI) bit description

Bit	Symbol	Value	Description	Reset value
0	RXRDYEN		Determines whether an interrupt occurs when receiver data is available.	0
		0	No interrupt will be generated when receiver data is available.	
		1	An interrupt will be generated when receiver data is available in the RXDAT register.	
1	TXRDYEN		Determines whether an interrupt occurs when the transmitter holding register is available.	0
		0	No interrupt will be generated when the transmitter holding register is available.	
		1	An interrupt will be generated when data may be written to TXDAT.	
2	RXOVEN		Determines whether an interrupt occurs when a receiver overrun occurs. This happens in slave mode when there is a need for the receiver to move newly received data to the RXDAT register when it is already in use.  The interface prevents receiver overrun in Master mode by not allowing a new transmission to begin when a receiver overrun would otherwise occur.	0
		0	No interrupt will be generated when a receiver overrun occurs.	
		1	An interrupt will be generated if a receiver overrun occurs.	
3	TXUREN		Determines whether an interrupt occurs when a transmitter underrun occurs. This happens in slave mode when there is a need to transmit data when none is available.	0
		0	No interrupt will be generated when the transmitter underruns.	
		1	An interrupt will be generated if the transmitter underruns.	

Table 175. SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI) 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.	
7:6	-		Reserved. Read value is undefined, only zero should be written.	NA
8	MSTIDLEEN		Master idle interrupt enable.	0
		0	No interrupt will be generated when the SPI master function is idle.	
		1	An interrupt will be generated when the SPI master function is fully idle.	
31:6	-		Reserved. Read value is undefined, only zero should be written.	NA

### 14.6.5 SPI Interrupt Enable Clear register

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

Table 176. SPI Interrupt Enable clear register (INTENCLR, addresses 0x4005 8010 (SPI) 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
7:6	-	Reserved. Read value is undefined, only zero should be written.	NA
8	MSTIDLE	Writing 1 clears the corresponding bit in the INTENSET register	0
31:6	-	Reserved. Read value is undefined, only zero should be written.	NA

### 14.6.6 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 14.7.4](#).



Table 177. SPI Receiver Data register (RXDAT, addresses 0x4005 8014 (SPI) 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

### 14.6.7 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.

**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 14.6.8](#)) instead of the TXDATCTL register. Control information can then be written separately via the TXCTL register (see [Section 14.6.9](#)). 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 14.7.4](#).

For details on using multiple consecutive data transmits for transfer lengths larger than 16 bit, see [Section 14.7.5 “Data lengths greater than 16 bits”](#).

**Table 178. SPI Transmitter Data and Control register (TXDATCTL, addresses 0x4005 8018 (SPI) 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. <b>Remark:</b> 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. <b>Remark:</b> The active state of the SSEL1 pin is configured by bits in the CFG register.	0
		0	SSEL1 asserted.	
		1	SSEL1 not asserted.	
19:18	-		Reserved. Read value is undefined, only zero should be written.	NA
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.	

Table 178. SPI Transmitter Data and Control register (TXDATCTL, addresses 0x4005 8018 (SPI) bit description

Bit	Symbol	Value	Description	Reset value
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.	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

### 14.6.8 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 14.6.7](#)). That data will be sent to the transmit shift register when it is available, and another character may then be written to TXDAT.

Table 179. SPI Transmitter Data Register (TXDAT, addresses 0x4005 801C (SPI) 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

### 14.6.9 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 14.6.7](#)). 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 14.6.7](#)) instead of TXDAT. Control information can then be written along with data.

Table 180. SPI Transmitter Control register (TXCTL, addresses 0x4005 8020 (SPI) 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
19:18	-	Reserved. Read value is undefined, only zero should be written.	NA
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

#### 14.6.10 SPI Divider register

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

For details on clocking, see [Section 14.7.3 “Clocking and data rates”](#).

Table 181. SPI Divider register (DIV, addresses 0x4005 8024 (SPI) 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

#### 14.6.11 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 174](#) for detailed descriptions of the interrupt flags.

Table 182. SPI Interrupt Status register (INTSTAT, addresses 0x4005 8028 (SPI) 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
7:6	-	Reserved. Read value is undefined, only zero should be written.	NA
8	MSTIDLE	Writing 1 clears the corresponding bit in the INTENSET register	0
31:9	-	Reserved. Read value is undefined, only zero should be written.	NA

14.7 Functional description

14.7.1 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 183](#) and shown in [Figure 25](#). CPOL and CPHA are configured by bits in the CFG register ([Section 14.6.1](#)).

Table 183. 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

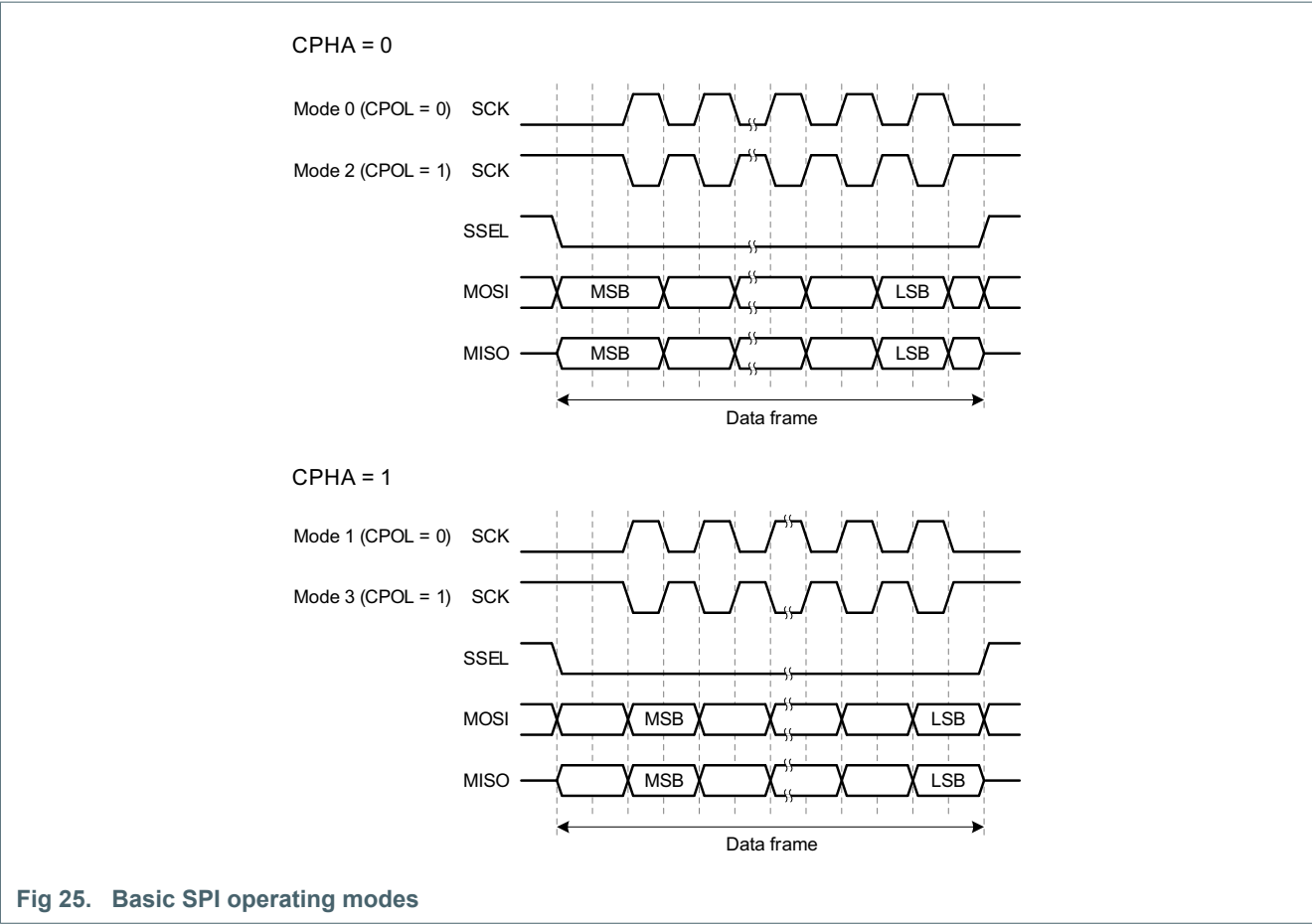


Fig 25. Basic SPI operating modes

## 14.7.2 Frame delays

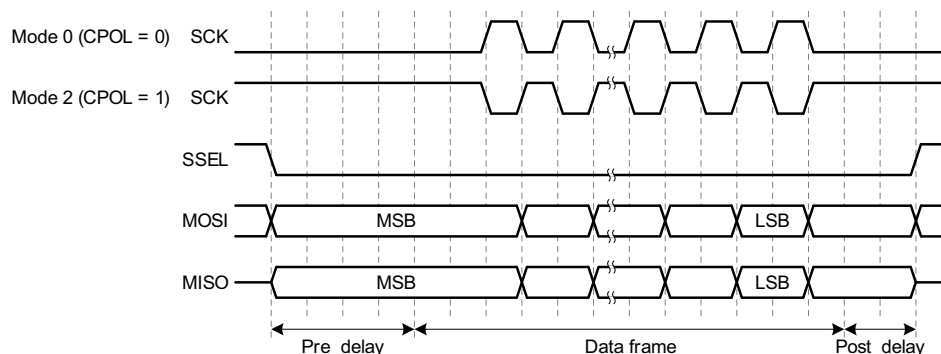
Several delays can be specified for SPI frames. These include:

- **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.

### 14.7.2.1 Pre\_delay and Post\_delay

**Pre\_delay** and **Post\_delay** are illustrated by the examples in [Figure 26](#). 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.

Pre- and post-delay: CPHA = 0, Pre\_delay = 2, Post\_delay = 1



Pre- and post-delay: CPHA = 1, Pre\_delay = 2, Post\_delay = 1

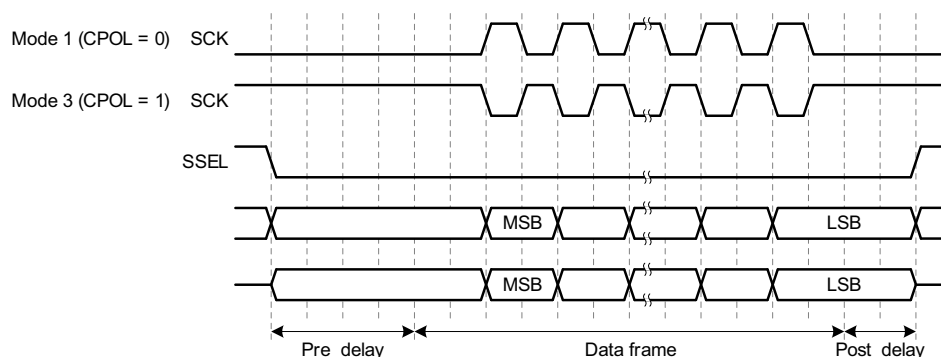
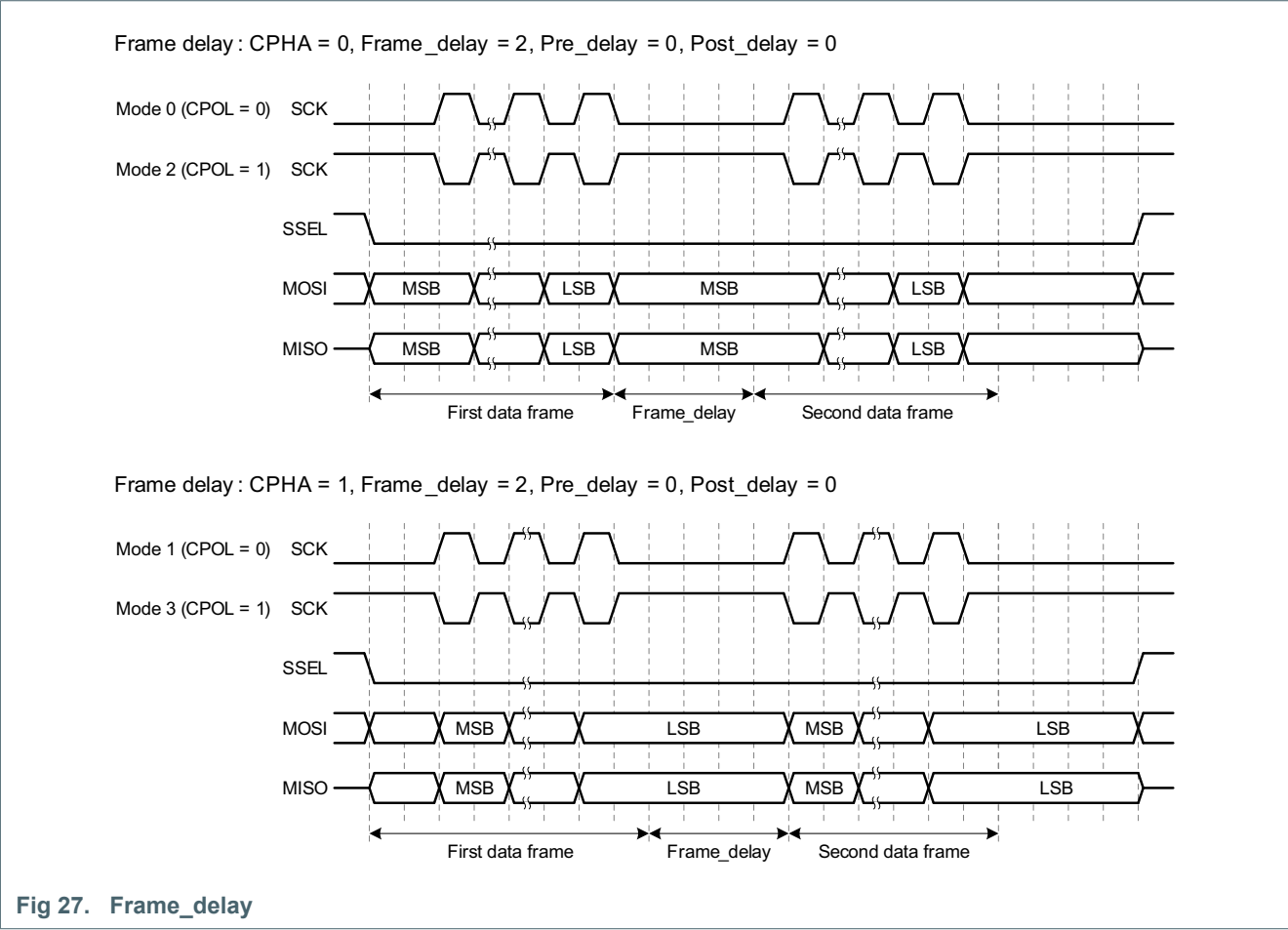


Fig 26. Pre\_delay and Post\_delay

14.7.2.2 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 27](#). Note that frame boundaries occur only where specified. This is because frame lengths can be any size, involving multiple data writes. See [Section 14.7.5](#) for more information.



14.7.2.3 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 28](#).

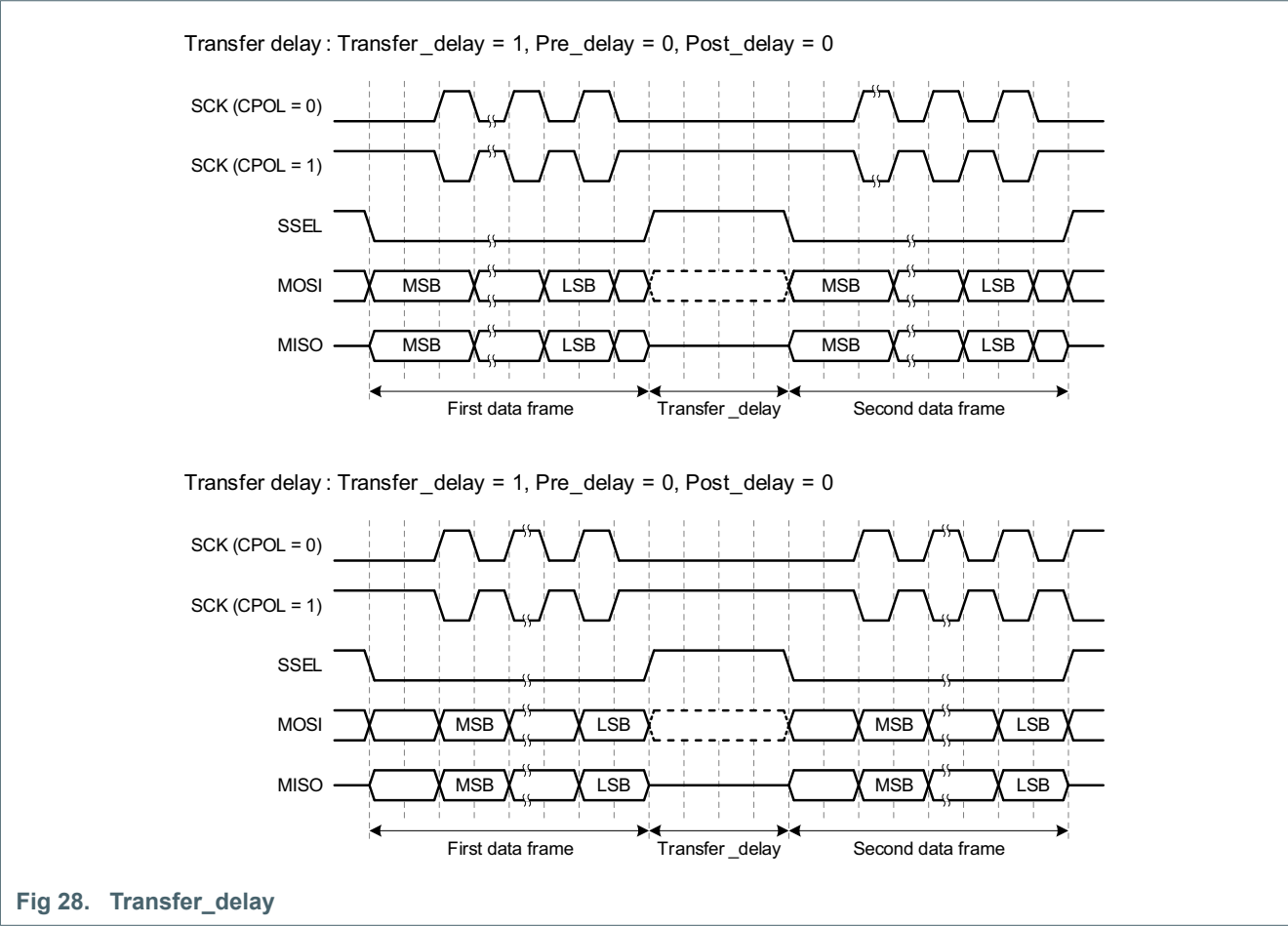


Fig 28. Transfer\_delay



### 14.7.3 Clocking and data rates

In order to use the SPI, clocking details must be defined. This includes configuring the system clock, peripheral clock, and selection of the clock divider value in DIV. See [Figure 23](#).

#### 14.7.3.1 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 FCLK, or at lower integer divide rates. The SPI rate will be  $= \text{FCLK}$ .

In slave mode, the clock is taken from the SCK input and the SPI clock divider is not used.

### 14.7.4 Slave select

The SPI block provides for two 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 two 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 two SSELs is saved along with received data in the RXSSEL\_N field of the RXDAT register.

### 14.7.5 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 be 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.

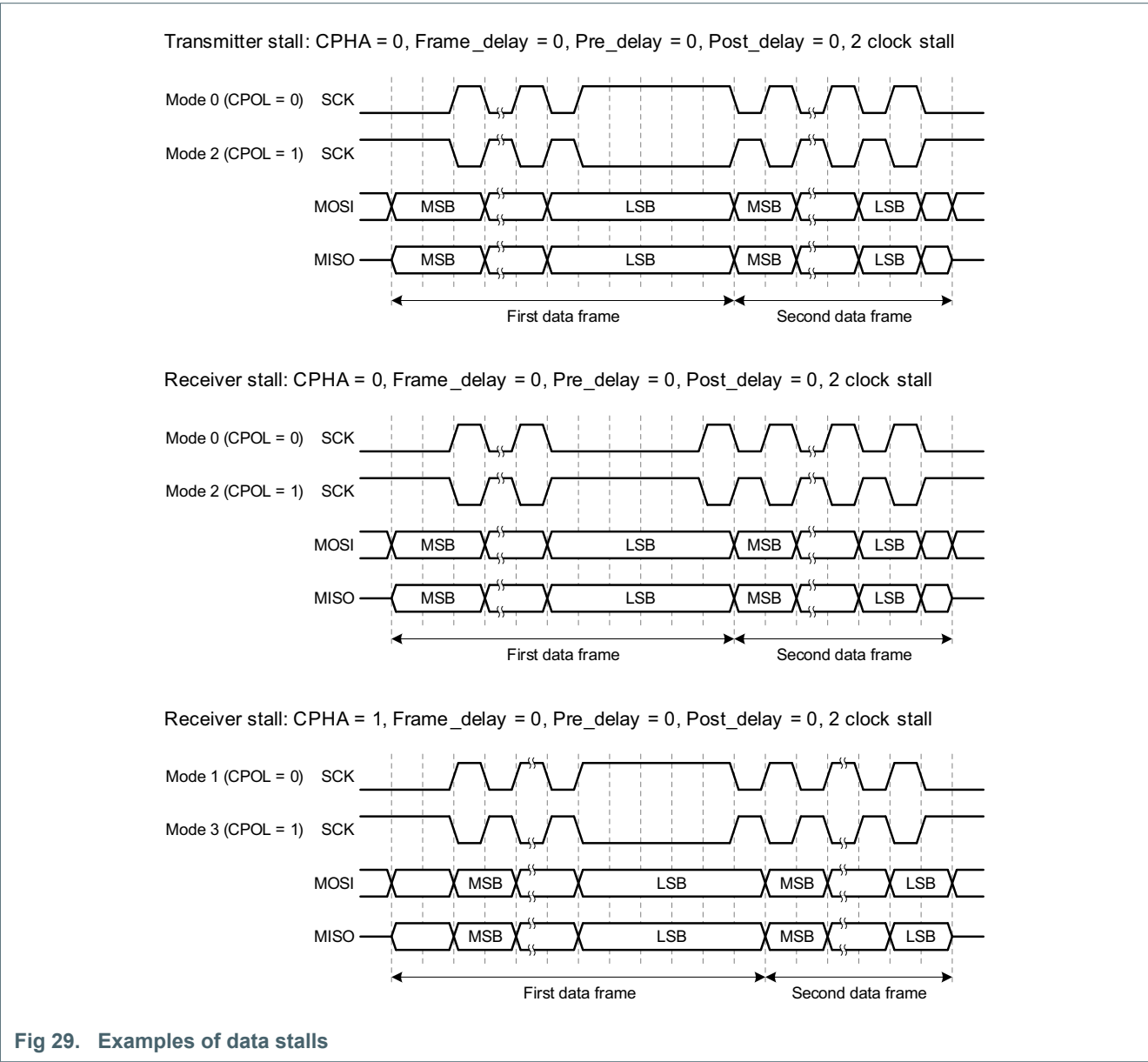
### 14.7.6 Data stalls

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.



### 15.1 How to read this chapter

The I<sup>2</sup>C interface is available on all parts.

Read this chapter if you want to understand the I<sup>2</sup>C operation and the software interface and want to learn how to use the I<sup>2</sup>C for wake-up from reduced power modes.

### 15.2 Features

- Independent Master, Slave, and Monitor functions.
- Supports both Multi-master and Multi-master with Slave functions.
- Multiple I<sup>2</sup>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<sup>2</sup>C bus addresses.
- 10-bit addressing supported with software assist.
- Supports System Management (SMBus).
- Supports the I<sup>2</sup>C-bus specification Fast-mode.

### 15.3 Basic configuration

Configure the I<sup>2</sup>C interfaces using the following registers:

- In the SYSAHBCLKCTRL register, set the corresponding bits to enable the clocks to the register interfaces. See [Table 63](#).
- Clear the I<sup>2</sup>C peripheral resets using the PRESETCTRL register ([Table 64](#)).
- Enable/disable the I<sup>2</sup>C interrupt in interrupt slots in the NVIC. See [Table 38](#).
- Configure the I<sup>2</sup>C pin functions through the switch matrix. See [Table 184](#).
- The peripheral clock for the I<sup>2</sup>C can be selected by the I2C0CLKSEL register among the FRO, main\_clk, frg0clk, and fro\_div clock sources (see [Figure 30](#)).

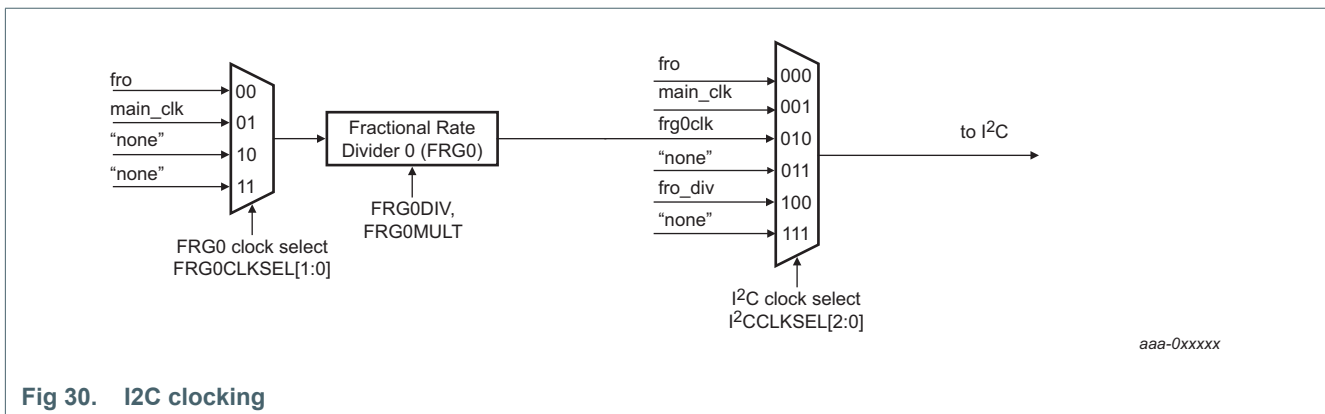


Fig 30. I2C clocking

### 15.3.1 I<sup>2</sup>C transmit/receive in master mode

In this example, the I<sup>2</sup>C 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 15 MHz and the bit rate is approximately 400 kHz. You can assign the SCL and SDA functions to pins through the switch matrix. See [Table 184](#).

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 I<sup>2</sup>C bit rate:

- Select a source for the FCLK that will allow for the required I<sup>2</sup>C-bus rate. Divide the clock as needed. See [Table 193 “I<sup>2</sup>C Clock Divider register \(CLKDIV, address 0x4005 0014 \(I2C0\) bit description”](#).
- Set the SCL high and low times to 2 clock cycles each. This is the default. See [Table 196 “Master Time register \(MSTTIME, address 0x4005 0024 \(I2C0\) bit description”](#). The result is an SCL clock of 375 kHz.

### 15.3.1.1 Master write to slave

Configure the I<sup>2</sup>C as master: Set the MSTEN bit to 1 in the CFG register. See [Table 186](#).

Write data to the slave:

1. Write the slave address with the  $\overline{RW}$  bit set to 0 to the Master data register MSTDAT. See [Table 197](#).
2. Start the transmission by setting the MSTSTART bit to 1 in the Master control register. See [Table 195](#). The following happens:
  - The pending status is cleared and the I<sup>2</sup>C-bus is busy.
  - The I2C master sends the start bit and address with the  $\overline{RW}$  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 195](#). The following happens:
  - The pending status is cleared and the I2C-bus is busy.
  - The I<sup>2</sup>C 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 195](#).

### 15.3.1.2 Master read from slave

Configure the I<sup>2</sup>C as master: Set the MSTEN bit to 1 in the CFG register. See [Table 186](#).

Read data from the slave:

1. Write the slave address with the  $\overline{RW}$  bit set to 1 to the Master data register MSTDAT. See [Table 197](#).
2. Start the transmission by setting the MSTSTART bit to 1 in the Master control register. See [Table 195](#). The following happens:
  - The pending status is cleared and the I<sup>2</sup>C - bus is busy.
  - The I<sup>2</sup>C master sends the start bit and address with the  $\overline{RW}$  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 195](#).

## 15.3.2 I2C receive/transmit in slave mode

In this example, the I<sup>2</sup>C is configured as the slave. The slave receives 8 bits from the master and sends 8 bits to the slave. The system clock is set to 15 MHz and the bit rate is approximately 400 kHz. You must assign the SCL and SDA functions to the required pins through the switch matrix. See [Table 184](#).

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.

#### 15.3.2.1 Slave read from master

Configure the I<sup>2</sup>C as slave with address x:

- Set the SLVEN bit to 1 in the CFG register. See [Table 186](#).
- Write the slave address x to the address 0 match register. See [Table 200](#).

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 198](#).
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 199](#).
5. Acknowledge (“ack”) the data by setting SLVCONTINUE = 1 in the slave control register. See [Table 198](#).

#### 15.3.2.2 Slave write to master

- Set the SLVEN bit to 1 in the CFG register. See [Table 186](#).
- Write the slave address x to the address 0 match register. See [Table 200](#).

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 198](#).
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 199](#).
5. Continue the transaction by setting SLVCONTINUE = 1 in the slave control register. See [Table 198](#).

### 15.3.3 Configure the I<sup>2</sup>C for wake-up

In sleep mode, any activity on the I<sup>2</sup>C -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 I<sup>2</sup>C clock and I2C\_PCLK remains active in sleep mode, the I<sup>2</sup>C can wake up the part independently of whether the I<sup>2</sup>C block is configured in master or slave mode.

In deep-sleep or power-down mode, the I<sup>2</sup>C clock is turned off as are all peripheral clocks. However, if the I<sup>2</sup>C is configured in slave mode and an external master on the I<sup>2</sup>C-bus provides the clock signal, the I<sup>2</sup>C block can create an interrupt asynchronously. This interrupt, if enabled in the NVIC and in the INTENCLR register of the INTENCLR register I<sup>2</sup>C block, can then wake up the core.

#### 15.3.3.1 Wake-up from sleep mode

- Enable the I<sup>2</sup>C interrupt in the NVIC.
- Enable the I<sup>2</sup>C wake-up event in the I<sup>2</sup>C 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

#### 15.3.3.2 Wake-up from deep-sleep and power-down modes

- Enable the I<sup>2</sup>C interrupt in the NVIC.
- Enable the I<sup>2</sup>C 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 79 “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 I<sup>2</sup>C in slave mode.
- Enable the I<sup>2</sup>C the interrupt in the I<sup>2</sup>C 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



## 15.4 Pin description

Pins for the I<sup>2</sup>C interfaces are movable functions and can be assigned to any pin. However, 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 184. I<sup>2</sup>C-bus pin description

Function	Direction	Type	Connect to	Use register	Reference	Description
I2C0_SDA	I/O	external to pin	any pin	PINASSIGN5	<a href="#">Table 94</a>	I2C0 serial data.
I2C0_SCL	I/O	external to pin	any pin	PINASSIGN5	<a href="#">Table 94</a>	I2C0 serial clock.

## 15.5 General description

The architecture of the I<sup>2</sup>C-bus interface is shown in [Figure 31](#).

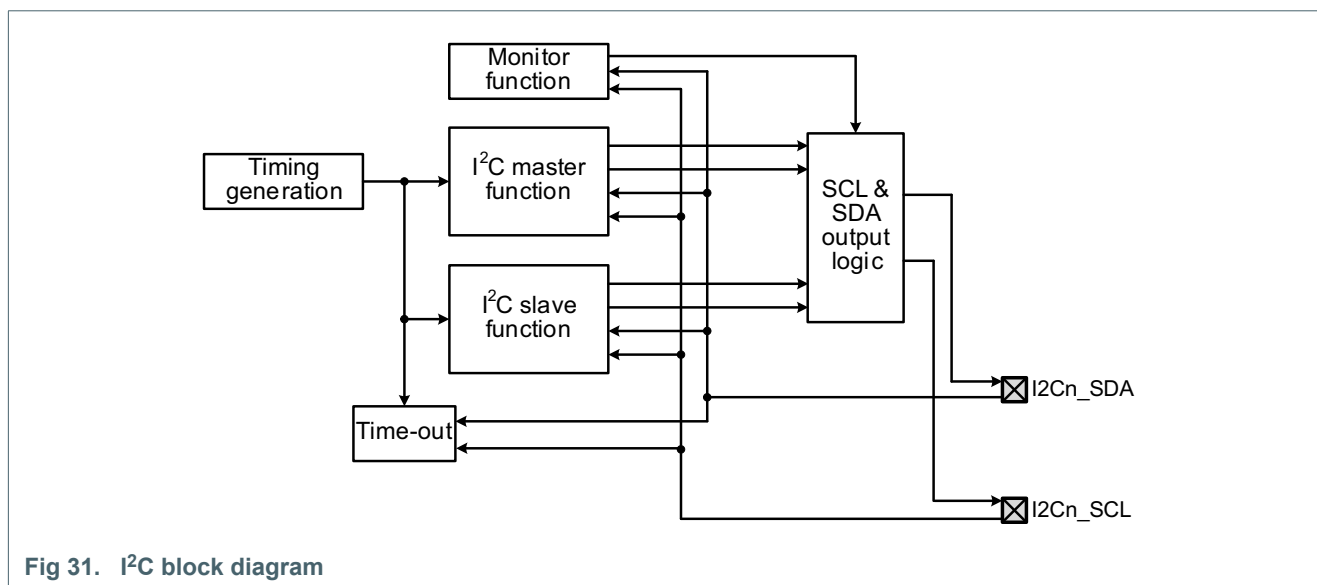


Fig 31. I<sup>2</sup>C block diagram

## 15.6 Register description

The register functions can be grouped as follows:

- Common registers:
  - [Table 186 “I<sup>2</sup>C Configuration register \(CFG, address 0x4005 0000 \(I2C0\) bit description”](#)
  - [Table 187 “I<sup>2</sup>C Status register \(STAT, address 0x4005 0004 \(I2C0\) bit description”](#)
  - [Table 194 “I<sup>2</sup>C Interrupt Status register \(INTSTAT, address 0x4005 0018 \(I2C0\) bit description”](#)
  - [Table 190 “Interrupt Enable Set and read register \(INTENSET, address 0x4005 0008 \(I2C0\) bit description”](#)

- [Table 191 “Interrupt Enable Clear register \(INTENCLR, address 0x4005 000C \(I2C0\) bit description”](#)
- [Table 192 “Time-out value register \(TIMEOUT, address 0x4005 0010 \(I2C0\) bit description”](#)
- [Table 193 “I<sup>2</sup>C Clock Divider register \(CLKDIV, address 0x4005 0014 \(I2C0\) bit description”](#)
- Master function registers:
  - [Table 195 “Master Control register \(MSTCTL, address 0x4005 0020 \(I2C0\) bit description”](#)
  - [Table 196 “Master Time register \(MSTTIME, address 0x4005 0024 \(I2C0\) bit description”](#)
  - [Table 197 “Master Data register \(MSTDAT, address 0x4005 0028 \(I2C0\) bit description”](#)
- Slave function registers:
  - [Table 198 “Slave Control register \(SLVCTL, address 0x4005 0040 \(I2C0\) bit description”](#)
  - [Table 198 “Slave Control register \(SLVCTL, address 0x4005 0040 \(I2C0\) bit description”](#)
  - [Table 200 “Slave Address registers \(SLVADR\[0:3\], address 0x4005 0048 \(SLVADR0\) to 0x4005 0054 \(SLVADR3\) \(I2C0\) bit description”](#)
  - [Table 201 “Slave address Qualifier 0 register \(SLVQUAL0, address 0x4005 0058 \(I2C0\) bit description”](#)
- Monitor function register: [Table 202 “Monitor data register \(MONRXDAT, address 0x4005 0080 \(I2C0\) bit description”](#)

Table 185. Register overview: I<sup>2</sup>C (base address 0x4005 0000 (I2C0))

Name	Access	Offset	Description	Reset value	Reference
CFG	R/W	0x00	Configuration for shared functions.	0	<a href="#">Table 186</a>
STAT	R/W	0x04	Status register for Master, Slave, and Monitor functions.	0x000801	<a href="#">Table 187</a>
INTENSET	R/W	0x08	Interrupt Enable Set and read register.	0	<a href="#">Table 190</a>
INTENCLR	W	0x0C	Interrupt Enable Clear register.	NA	<a href="#">Table 191</a>
TIMEOUT	R/W	0x10	Time-out value register.	0xFFFF	<a href="#">Table 192</a>
CLKDIV	R/W	0x14	Clock pre-divider for the entire I <sup>2</sup> C block. This determines what time increments are used for the MSTTIME register.	0	<a href="#">Table 193</a>
INTSTAT	R	0x18	Interrupt Status register for Master, Slave, and Monitor functions.	0	<a href="#">Table 194</a>
MSTCTL	R/W	0x20	Master control register.	0	<a href="#">Table 195</a>
MSTTIME	R/W	0x24	Master timing configuration.	0x77	<a href="#">Table 196</a>
MSTDAT	R/W	0x28	Combined Master receiver and transmitter data register.	NA	<a href="#">Table 197</a>
SLVCTL	R/W	0x40	Slave control register.	0	<a href="#">Table 198</a>
SLVDAT	R/W	0x44	Combined Slave receiver and transmitter data register.	NA	<a href="#">Table 199</a>
SLVADR0	R/W	0x48	Slave address 0.	0x01	<a href="#">Table 200</a>
SLVADR1	R/W	0x4C	Slave address 1.	0x01	<a href="#">Table 200</a>
SLVADR2	R/W	0x50	Slave address 2.	0x01	<a href="#">Table 200</a>
SLVADR3	R/W	0x54	Slave address 3.	0x01	<a href="#">Table 200</a>
SLVQUAL0	R/W	0x58	Slave Qualification for address 0.	0	<a href="#">Table 201</a>
MONRXDAT	RO	0x80	Monitor receiver data register.	0	<a href="#">Table 202</a>

### 15.6.1 I<sup>2</sup>C Configuration register

The CFG register contains mode settings that apply to Master, Slave, and Monitor functions.

Table 186. I<sup>2</sup>C Configuration register (CFG, address 0x4005 0000 (I2C0)) bit description

Bit	Symbol	Value	Description	Reset Value
0	MSTEN		Master Enable. When disabled, configurations settings for the Master function are not changed, but the Master function is internally reset.	0
		0	Disabled. The I <sup>2</sup> C Master function is disabled.	
		1	Enabled. The I <sup>2</sup> C Master function is enabled.	
1	SLVEN		Slave Enable. When disabled, configurations settings for the Slave function are not changed, but the Slave function is internally reset.	0
		0	Disabled. The I <sup>2</sup> C slave function is disabled.	
		1	Enabled. The I <sup>2</sup> C slave function is enabled.	
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 <sup>2</sup> C monitor function is disabled.	
		1	Enabled. The I <sup>2</sup> C monitor function is enabled.	

Table 186. I<sup>2</sup>C Configuration register (CFG, address 0x4005 0000 (I2C0) bit description

Bit	Symbol	Value	Description	Reset Value
3	TIMEOUTEN		I <sup>2</sup> 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 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 to ensure that software can read all incoming data supplied by the monitor function.	
31:5	-		Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.2 I<sup>2</sup>C Status register

The STAT register provides status flags and state information about all of the functions of the I<sup>2</sup>C block. Some information in this register is read-only and some flags can be cleared by writing a 1 to them.

Access to bits in this register varies. RO = Read-only, W1 = write 1 to clear.

Details on the master and slave states described in the MSTSTATE and SLVSTATE bits in this register are listed in [Table 188](#) and [Table 189](#).

**Table 187. I<sup>2</sup>C Status register (STAT, address 0x4005 0004 (I2C0) bit description**

Bit	Symbol	Value	Description	Reset value	Access
0	MSTPENDING		Master Pending. Indicates that the Master is waiting to continue communication on the I2C-bus (pending) or is idle. When the master is pending, the MSTSTATE bits indicate what type of software service if any the master expects. This flag will cause an interrupt when set if, enabled via the INTENSET register. 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 187. I<sup>2</sup>C Status register (STAT, address 0x4005 0004 (I2C0) bit description ...continued

Bit	Symbol	Value	Description	Reset value	Access
6	MSTSTSPERR		Master Start/Stop Error flag. This flag can be cleared by software writing a 1 to this bit. It is also cleared automatically a 1 is written to MSTCONTINUE.	0	W1
		0	No Start/Stop Error has occurred.		
		1	Start/stop error has occurred. The Master function has experienced a Start/Stop Error.  A Start or Stop was detected at a time when it is not allowed by the I <sup>2</sup> C specification. The Master interface has stopped driving the bus and gone to an idle state, no action is required. A request for a Start could be made, or software could attempt to insure that the bus has not stalled.		
7	-		Reserved. Read value is undefined, only zero should be written.	NA	NA
8	SLVPENDING		Slave Pending. Indicates that the Slave function is waiting to continue communication on the I <sup>2</sup> C-bus and needs software service. This flag will cause an interrupt when set if enabled via INTENSET. The SLVPENDING flag is read-only and is automatically cleared when a 1 is written to the SLVCONTINUE bit in the MSTCTL register.	0	RO
		0	In progress. The Slave function does not currently need service.		
		1	Pending. The Slave function needs service. Information on what is needed can be found in the adjacent SLVSTATE field.		
10:9	SLVSTATE		Slave State code. Each value of this field indicates a specific required service for the Slave function. All other values are reserved.	0	RO
		0x0	Slave address. Address plus R/W received. At least one of the four slave addresses has been matched by hardware.		
		0x1	Slave receive. Received data is available (Slave Receiver mode).		
		0x2	Slave transmit. Data can be transmitted (Slave Transmitter mode).		
		0x3	Reserved.		
11	SLVNOTSTR		Slave Not Stretching. Indicates when the slave function is stretching the I <sup>2</sup> C clock. This is needed in order to gracefully invoke deep Sleep or power-down modes during slave operation. This read-only flag reflects the slave function status in real time.	1	RO
		0	Stretching. The slave function is currently stretching the I <sup>2</sup> C bus clock. Deep-Sleep or power-down mode cannot be entered at this time.		
		1	Not stretching. The slave function is not currently stretching the I <sup>2</sup> C bus clock. Deep-sleep or power-down mode could be entered at this time.		
13:12	SLVIDX		Slave address match Index. This field is valid when the I <sup>2</sup> C slave function has been selected by receiving an address that matches one of the slave addresses defined by any enabled slave address registers, and provides an identification of the address that was matched. It is possible that more than one address could be matched, but only one match can be reported here.	0	RO
		0x0	Slave address 0 was matched.		
		0x1	Slave address 1 was matched.		
		0x2	Slave address 2 was matched.		
		0x3	Slave address 3 was matched.		

Table 187. I<sup>2</sup>C Status register (STAT, address 0x4005 0004 (I2C0) bit description ...continued

Bit	Symbol	Value	Description	Reset value	Access
14	SLVSEL		Slave selected flag. SLVSEL is set after an address match when software tells the Slave function to acknowledge the address. It is cleared when another address cycle presents an address that does not match an enabled address on the Slave function, when slave software decides to NACK a matched address, or when there is a Stop detected on the bus. SLVSEL is not cleared if software Nacks data.	0	RO
		0	Not selected. The Slave function is not currently selected.		
		1	Selected. The Slave function is currently selected.		
15	SLVDESEL		Slave Deselected flag. This flag will cause an interrupt when set if enabled via INTENSET. This flag can be cleared by writing a 1 to this bit.	0	W1
		0	Not deselected. The Slave function has not become deselected. This does not mean that it is currently selected. That information can be found in the SLVSEL flag.		
		1	Deselected. The Slave function has become deselected. This is specifically caused by the SLVSEL flag changing from 1 to 0. See the description of SLVSEL for details on when that event occurs.		
16	MONRDY		Monitor Ready. This flag is cleared when the MONRXDAT register is read.	0	RO
		0	No data. The Monitor function does not currently have data available.		
		1	Data waiting. The Monitor function has data waiting to be read.		
17	MONOV		Monitor Overflow flag.	0	W1
		0	No overrun. Monitor data has not overrun.		
		1	Overrun. A Monitor data overrun has occurred. This can only happen when Monitor clock stretching not enabled via the MONCLKSTR bit in the CFG register. Writing 1 to this bit clears the flag.		
18	MONACTIVE		Monitor Active flag. This flag indicates when the Monitor function considers the I <sup>2</sup> C bus to be active. Active is defined here as when some Master is on the bus: a bus Start has occurred more recently than a bus Stop.	0	RO
		0	Inactive. The Monitor function considers the I <sup>2</sup> C bus to be inactive.		
		1	Active. The Monitor function considers the I <sup>2</sup> C bus to be active.		
19	MONIDLE		Monitor Idle flag. This flag is set when the Monitor function sees the I <sup>2</sup> C bus change from active to inactive. This can be used by software to decide when to process data accumulated by the Monitor function. This flag will cause an interrupt when set if enabled via the INTENSET register. The flag can be cleared by writing a 1 to this bit.	0	W1
		0	Not idle. The I <sup>2</sup> C bus is not idle, or this flag has been cleared by software.		
		1	Idle. The I <sup>2</sup> C bus has gone idle at least once since the last time this flag was cleared by software.		
23:20	-		Reserved. Read value is undefined, only zero should be written.	NA	NA

Table 187. I<sup>2</sup>C Status register (STAT, address 0x4005 0004 (I2C0) bit description ...continued

Bit	Symbol	Value	Description	Reset value	Access
24	EVENTTIMEOUT		Event Time-out Interrupt flag. Indicates when the time between events has been longer than the time specified by the TIMEOUT register. Events include Start, Stop, and clock edges. The flag is cleared by writing a 1 to this bit. No time-out is created when the I2C-bus is idle.	0	W1
		0	No time-out. I <sup>2</sup> C bus events have not caused a time-out.		
		1	Event time-out. The time between I <sup>2</sup> C bus events has been longer than the time specified by the I2C TIMEOUT register.		
25	SCLTIMEOUT		SCL Time-out Interrupt flag. Indicates when SCL has remained low longer than the time specific by the TIMEOUT register. The flag is cleared by writing a 1 to this bit.	0	W1
		0	No time-out. SCL low time has not caused a time-out.		
		1	Time-out. SCL low time has caused a time-out.		
31:26	-		Reserved. Read value is undefined, only zero should be written.	NA	NA

Table 188. Master function state codes (MSTSTATE)

MSTSTATE	Description	Actions
0x0	<b>Idle.</b> 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.
0x1	<b>Received data is available (Master Receiver mode).</b> Address plus Read was previously sent and Acknowledged by slave.	Read data and either continue, send a Stop, or send a Repeated Start.
0x2	<b>Data can be transmitted (Master Transmitter mode).</b> Address plus Write was previously sent and Acknowledged by slave.	Send data and continue, or send a Stop or Repeated Start.
0x3	<b>Slave NACKed address.</b>	Send a Stop or Repeated Start.
0x4	<b>Slave NACKed transmitted data.</b>	Send a Stop or Repeated Start.

Table 189. Slave function state codes (SLVSTATE)

SLVSTATE	Description	Actions
0	SLVST_ADDR <b>Address plus R/W received.</b> 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 <a href="#">Section 15.7.3</a> regarding 10-bit addressing.
1	SLVST_RX <b>Received data is available (Slave Receiver mode).</b>	Read data reply with an ACK or a NACK.
2	SLVST_TX <b>Data can be transmitted (Slave Transmitter mode).</b>	Send data.
3	-	Reserved.



### 15.6.3 Interrupt Enable Set and read register

The INTENSET register controls which I<sup>2</sup>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 190. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0) bit description**

Bit	Symbol	Value	Description	Reset value
0	MSTPENDINGEN		Master Pending interrupt Enable.	0
		0	The MstPending interrupt is disabled.	
		1	The MstPending interrupt is enabled.	
3:1	-		Reserved. Read value is undefined, only zero should be written.	NA
4	MSTARBLOSSEN		Master Arbitration Loss interrupt Enable.	0
		0	The MstArbLoss interrupt is disabled.	
		1	The MstArbLoss interrupt is enabled.	
5	-		Reserved. Read value is undefined, only zero should be written.	NA
6	MSTSTSTPERREN		Master Start/Stop Error interrupt Enable.	0
		0	The MstStStpErr interrupt is disabled.	
		1	The MstStStpErr interrupt is enabled.	
7	-		Reserved. Read value is undefined, only zero should be written.	NA
8	SLVPENDINGEN		Slave Pending interrupt Enable.	0
		0	The SlvPending interrupt is disabled.	
		1	The SlvPending interrupt is enabled.	
10:9	-		Reserved. Read value is undefined, only zero should be written.	NA
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

Table 190. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0) bit description

Bit	Symbol	Value	Description	Reset value
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

### 15.6.4 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 191. Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0) bit description

Bit	Symbol	Description	Reset value
0	MSTPENDINGCLR	Master Pending interrupt clear. Writing 1 to this bit clears the corresponding bit in the INTENSET register if implemented.	0
3:1	-	Reserved. Read value is undefined, only zero should be written.	NA
4	MSTARLOSSCLR	Master Arbitration Loss interrupt clear.	0
5	-	Reserved. Read value is undefined, only zero should be written.	NA
6	MSTSTSPERRCLR	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

### 15.6.5 Time-out value register

The TIMEOUT register allows setting an upper limit to certain I<sup>2</sup>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 15.7.2 “Time-out”](#).

**Table 192. Time-out value register (TIMEOUT, address 0x4005 0010 (I2C0) bit description**

Bit	Symbol	Description	Reset value
3:0	TOMIN	Time-out time value, bottom four bits. These are hard-wired to 0xF. This gives a minimum time-out of 16 I <sup>2</sup> C function clocks and also a time-out resolution of 16 I <sup>2</sup> C function clocks.	0xF
15:4	TO	Time-out time value. Specifies the time-out interval value in increments of 16 I <sup>2</sup> C function clocks, as defined by the CLKDIV register. To change this value while I <sup>2</sup> C is in operation, disable all time-outs, write a new value to TIMEOUT, then re-enable time-outs. 0x000 = A time-out will occur after 16 counts of the I <sup>2</sup> C function clock. 0x001 = A time-out will occur after 32 counts of the I <sup>2</sup> C function clock. ... 0xFFFF = A time-out will occur after 65,536 counts of the I <sup>2</sup> C function clock.	0xFFFF
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

## 15.6.6 Clock Divider register

The CLKDIV register divides down the FCLK to produce the I<sup>2</sup>C function clock that is used to time various aspects of the I<sup>2</sup>C interface. The I<sup>2</sup>C function clock is used for some internal operations in the I<sup>2</sup>C block and to generate the timing required by the I<sup>2</sup>C bus specification, some of which are user configured in the MSTTIME register for Master operation.

See [Section 15.7.1.1 “Rate calculations”](#) for details on bus rate setup.

**Table 193. I<sup>2</sup>C Clock Divider register (CLKDIV, address 0x4005 0014 (I2C0) bit description**

Bit	Symbol	Description	Reset value
15:0	DIVVAL	This field controls how the clock (FCLK) is used by the I <sup>2</sup> C functions that need an internal clock in order to operate. 0x0000 = FCLK is used directly by the I <sup>2</sup> C function. 0x0001 = FCLK is divided by 2 before use by the I <sup>2</sup> C function. 0x0002 = FCLK is divided by 3 before use by the I <sup>2</sup> C function. ... 0xFFFF = FCLK is divided by 65,536 before use by the I <sup>2</sup> C function.	0
31:16	-	Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.7 Interrupt Status register

The INTSTAT register provides a view of those interrupt flags that are currently enabled. This can simplify software handling of interrupts. See [Table 187](#) for detailed descriptions of the interrupt flags.

Table 194. I<sup>2</sup>C Interrupt Status register (INTSTAT, address 0x4005 0018 (I2C0) bit description

Bit	Symbol	Description	Reset value
0	MSTPENDING	Master Pending.	1
3:1	-	Reserved.	
4	MSTARBLOSS	Master Arbitration Loss flag.	0
5	-	Reserved. Read value is undefined, only zero should be written.	NA
6	MSTSTSPERR	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

### 15.6.8 Master Control register

The MSTCTL register contains bits that control various functions of the I<sup>2</sup>C Master interface. Only write to this register when the master is pending (MSTPENDING = 1 in the STAT register, [Table 187](#)).

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.

Table 195. Master Control register (MSTCTL, address 0x4005 0020 (I2C0) bit description

Bit	Symbol	Value	Description	Reset value
0	MSTCONTINUE		Master Continue. This bit is write-only.	0
		0	No effect.	
		1	Continue. Informs the Master function to continue to the next operation. This must be done after writing transmit data, reading received data, or any other housekeeping related to the next bus operation.	
1	MSTSTART		Master Start control. This bit is write-only.	0
		0	No effect.	
		1	Start. A Start will be generated on the I <sup>2</sup> C bus at the next allowed time.	
2	MSTSTOP		Master Stop control. This bit is write-only.	0
		0	No effect.	
		1	Stop. A Stop will be generated on the I <sup>2</sup> C bus at the next allowed time, preceded by a NACK to the slave if the master is receiving data from the slave (Master Receiver mode).	
31:3	-		Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.9 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 193](#).

Table 196. Master Time register (MSTTIME, address 0x4005 0024 (I2C0) bit description

Bit	Symbol	Value	Description	Reset value
2:0	MSTSCLOW		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 <sup>2</sup> C bus specification. I <sup>2</sup> C bus specification parameters $t_{BUF}$ and $t_{SU,STA}$ have the same values and are also controlled by MSTSCLOW.	0x7
		0x0	2 clocks. Minimum SCL low time is 2 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x1	3 clocks. Minimum SCL low time is 3 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x2	4 clocks. Minimum SCL low time is 4 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x3	5 clocks. Minimum SCL low time is 5 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x4	6 clocks. Minimum SCL low time is 6 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x5	7 clocks. Minimum SCL low time is 7 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x6	8 clocks. Minimum SCL low time is 8 clocks of the I <sup>2</sup> C clock pre-divider.	
		0x7	9 clocks. Minimum SCL low time is 9 clocks of the I <sup>2</sup> C clock pre-divider.	
3	-		Reserved.	0

Table 196. Master Time register (MSTTIME, address 0x4005 0024 (I2C0) 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 $t_{\text{HIGH}}$ in the I <sup>2</sup> C bus specification. I <sup>2</sup> C bus specification parameters $t_{\text{SU,STO}}$ and $t_{\text{HD,STA}}$ have the same values and are also controlled by MSTSCLHIGH.	0x7
		0x0	2 clocks. Minimum SCL high time is 2 clock of the I <sup>2</sup> C clock pre-divider.	
		0x1	3 clocks. Minimum SCL high time is 3 clocks of the I <sup>2</sup> C clock pre-divider .	
		0x2	4 clocks. Minimum SCL high time is 4 clock of the I <sup>2</sup> C clock pre-divider.	
		0x3	5 clocks. Minimum SCL high time is 5 clock of the I <sup>2</sup> C clock pre-divider.	
		0x4	6 clocks. Minimum SCL high time is 6 clock of the I <sup>2</sup> C clock pre-divider.	
		0x5	7 clocks. Minimum SCL high time is 7 clock of the I <sup>2</sup> C clock pre-divider.	
		0x6	8 clocks. Minimum SCL high time is 8 clock of the I <sup>2</sup> C clock pre-divider.	
		0x7	9 clocks. Minimum SCL high time is 9 clocks of the I <sup>2</sup> C clock pre-divider.	
31:7	-		Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.10 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 197. Master Data register (MSTDAT, address 0x4005 0028 (I2C0) bit description

Bit	Symbol	Description	Reset value
7:0	DATA	Master function data register. Read: read the most recently received data for the Master function. Write: transmit data using the Master function.	0
31:8	-	Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.11 Slave Control register

The SLVCTL register contains bits that control various functions of the I<sup>2</sup>C Slave interface. Only write to this register when the slave is pending (SLVPENDING = 1 in the STAT register, [Table 187](#)).

Table 198. Slave Control register (SLVCTL, address 0x4005 0040 (I2C0) bit description

Bit	Symbol	Value	Description	Reset Value
0	SLVCONTINUE		Slave Continue.	0
		0	No effect.	
		1	Continue. Informs the Slave function to continue to the next operation. This must done after writing transmit data, reading received data, or any other housekeeping related to the next bus operation.	

Table 198. Slave Control register (SLVCTL, address 0x4005 0040 (I2C0) bit description

Bit	Symbol	Value	Description	Reset Value
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
31:3	-		Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.12 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 199. Slave Data register (SLVDAT, address 0x4005 0044 (I2C0) bit description

Bit	Symbol	Description	Reset Value
7:0	DATA	Slave function data register. Read: read the most recently received data for the Slave function. Write: transmit data using the Slave function.	0
31:8	-	Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.13 Slave Address registers

The SLVADR[0:3] registers allow enabling and defining one of the addresses that can be automatically recognized by the I<sup>2</sup>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 15.6.14](#)).

The I<sup>2</sup>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 200. Slave Address registers (SLVADR[0:3], address 0x4005 0048 (SLVADR0) to 0x4005 0054 (SLVADR3) (I2C0) bit description

Bit	Symbol	Value	Description	Reset value
0	SADISABLE		Slave Address n Disable.	1
		0	Enabled. Slave Address n is enabled and will be recognized with any changes specified by the SLVQUAL0 register.	
		1	Ignored Slave Address n is ignored.	
7:1	SLVADR		Seven bit slave address that is compared to received addresses if enabled.	0
31:8	-		Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.14 Slave address Qualifier 0 register

The SLVQUAL0 register can alter how Slave Address 0 is interpreted.



Table 201. Slave address Qualifier 0 register (SLVQUAL0, address 0x4005 0058 (I2C0) bit description

Bit	Symbol	Value	Description	Reset Value
0	QUALMODE0		Reserved. Read value is undefined, only zero should be written.	0
		0	The SLVQUAL0 field is used as a logical mask for matching address 0.	
		1	The SLVQUAL0 field is used to extend address 0 matching in a range of addresses.	
7:1	SLVQUAL0		Slave address Qualifier for address 0. A value of 0 causes the address in SLVADR0 to be used as-is, assuming that it is enabled.  If QUALMODE0 = 0, any bit in this field which is set to 1 will cause an automatic match of the corresponding bit of the received address when it is compared to the SLVADR0 register.  If QUALMODE0 = 1, an address range is matched for address 0. This range extends from the value defined by SLVADR0 to the address defined by SLVQUAL0 (address matches when SLVADR0[7:1] <= received address <= SLVQUAL0[7:1]).	0
31:8	-		Reserved. Read value is undefined, only zero should be written.	NA

### 15.6.15 Monitor data register

The read-only MONRXDAT register provides information about events on the I<sup>2</sup>C bus, primarily to facilitate debugging of the I<sup>2</sup>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<sup>2</sup>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 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<sup>2</sup>C bus.

Table 202. Monitor data register (MONRXDAT, address 0x4005 0080 (I2C0) bit description

Bit	Symbol	Value	Description	Reset value
7:0	MONRXDAT		Monitor function Receiver Data. This reflects every data byte that passes on the I <sup>2</sup> C pins, and adds indication of Start, Repeated Start, and data NACK.	0
8	MONSTART		Monitor Received Start.	0
		0	No detect. The monitor function has not detected a Start event on the I <sup>2</sup> C bus.	
		1	Start detect. The monitor function has detected a Start event on the I <sup>2</sup> C bus.	
9	MONRESTART		Monitor Received Repeated Start.	0
		0	No start detect. The monitor function has not detected a Repeated Start event on the I <sup>2</sup> C bus.	
		1	Repeated start detect. The monitor function has detected a Repeated Start event on the I <sup>2</sup> C bus.	



Table 202. Monitor data register (MONRXDAT, address 0x4005 0080 (I2C0) bit description

Bit	Symbol	Value	Description	Reset value
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

## 15.7 Functional description

### 15.7.1 Bus rates and timing considerations

Due to the nature of the I<sup>2</sup>C bus, it is generally not possible to guarantee a specific clock rate on the SCL pin. On the I2C-bus, 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 (that is, when it is the only master on the bus, or during arbitration between masters).

Rate calculations give a base frequency that represents the fastest that the I<sup>2</sup>C bus could operate if nothing slows it down.

#### 15.7.1.1 Rate calculations

SCL high time (in I<sup>2</sup>C function clocks) = (CLKDIV + 1) \* (MSTSCLHIGH + 2)

SCL low time (in I<sup>2</sup>C function clocks) = (CLKDIV + 1) \* (MSTSCLLOW + 2)

Nominal SCL rate = I<sup>2</sup>C function clock rate / (SCL high time + SCL low time)

**Remark:** DIVVAL must be  $\geq 1$ .

**Remark:** For 400 KHz clock rate, the clock frequency after the I<sup>2</sup>C divider (divval) must be  $\leq 2$  MHz. [Table 203](#) shows the recommended settings for 400 KHz clock rate.

Table 203. 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

### 15.7.2 Time-out

A time-out feature on an I<sup>2</sup>C interface can be used to detect a “stuck” bus and potentially do something to alleviate the condition. Two different types of time-out are supported. Both types apply whenever the I<sup>2</sup>C block and the time-out function are both enabled, Master, Slave, or Monitor functions do not need to be enabled.

In the first type of time-out, reflected by the EVENTTIMEOUT flag in the STAT register, the time between bus events governs the time-out check. These events include Start, Stop, and all changes on the I<sup>2</sup>C clock (SCL). This time-out is asserted when the time between any of these events is longer than the time configured in the TIMEOUT register. This time-out could be useful in monitoring an I<sup>2</sup>C bus within a system as part of a method to keep the bus running of problems occur.

The second type of I<sup>2</sup>C time-out is reflected by the SCLTIMEOUT flag in the STAT register. This time-out is asserted when the SCL signal remains low longer than the time configured in the TIMEOUT register. This corresponds to SMBus time-out parameter T<sub>TIMEOUT</sub>. In this situation, a slave could reset its own I<sup>2</sup>C interface in case it is the

offending device. If all listening slaves (including masters that can be addressed as slaves) do this, then the bus will be released unless it is a current master causing the problem. Refer to the SMBus specification for more details.

Both types of time-out are generated when the I<sup>2</sup>C bus is considered busy.

### 15.7.3 Ten-bit addressing

Ten-bit addressing is accomplished by the I<sup>2</sup>C master sending a second address byte to extend a particular range of standard 7-bit addresses. In the case of the master writing to the slave, the I<sup>2</sup>C frame simply continues with data after the 2 address bytes. For the master to read from a slave, it needs to reverse the data direction after the second address byte. This is done by sending a Repeated Start, followed by a repeat of the same standard 7-bit address, with a Read bit. The slave must remember that it had been addressed by the previous write operation and stay selected for the subsequent read with the correct partial I<sup>2</sup>C address.

For the Master function, the I<sup>2</sup>C is 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 15.6.14](#)) 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.

### 15.7.4 Clocking and power considerations

The Master function of the I<sup>2</sup>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<sup>2</sup>C Slave function recognizes an address. Monitor mode can similarly wake up the device from a reduced power mode when information becomes available.

### 15.7.5 Interrupt handling

The I2C provides a single interrupt output that handles all interrupts for Master, Slave, and Monitor functions.

### 16.1 How to read this chapter

---

The standard timer is available on all LPC802 devices.

### 16.2 Features

---

- 32-bit counter/timer with a programmable 32-bit prescaler. The timer includes external capture and match pin connections.
- Counter or timer operation.
- Three 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 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.
- Four 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.

### 16.3 Basic configuration

---

- Set the appropriate bit to enable the clock to timer: CTIMER in the AHBCLKCTRL0 register ([Section 6.6.9](#)).
- Clear the timer reset using the PRESETCTRL1 register ([Table 64](#) 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 211](#)) and CCR ([Table 213](#)) for match and capture events. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register. For interrupt connections, see [Table 38](#).

## 16.4 General description

The Counter/timer 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).

### 16.4.1 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 16.6.11](#).

### 16.4.2 Match outputs

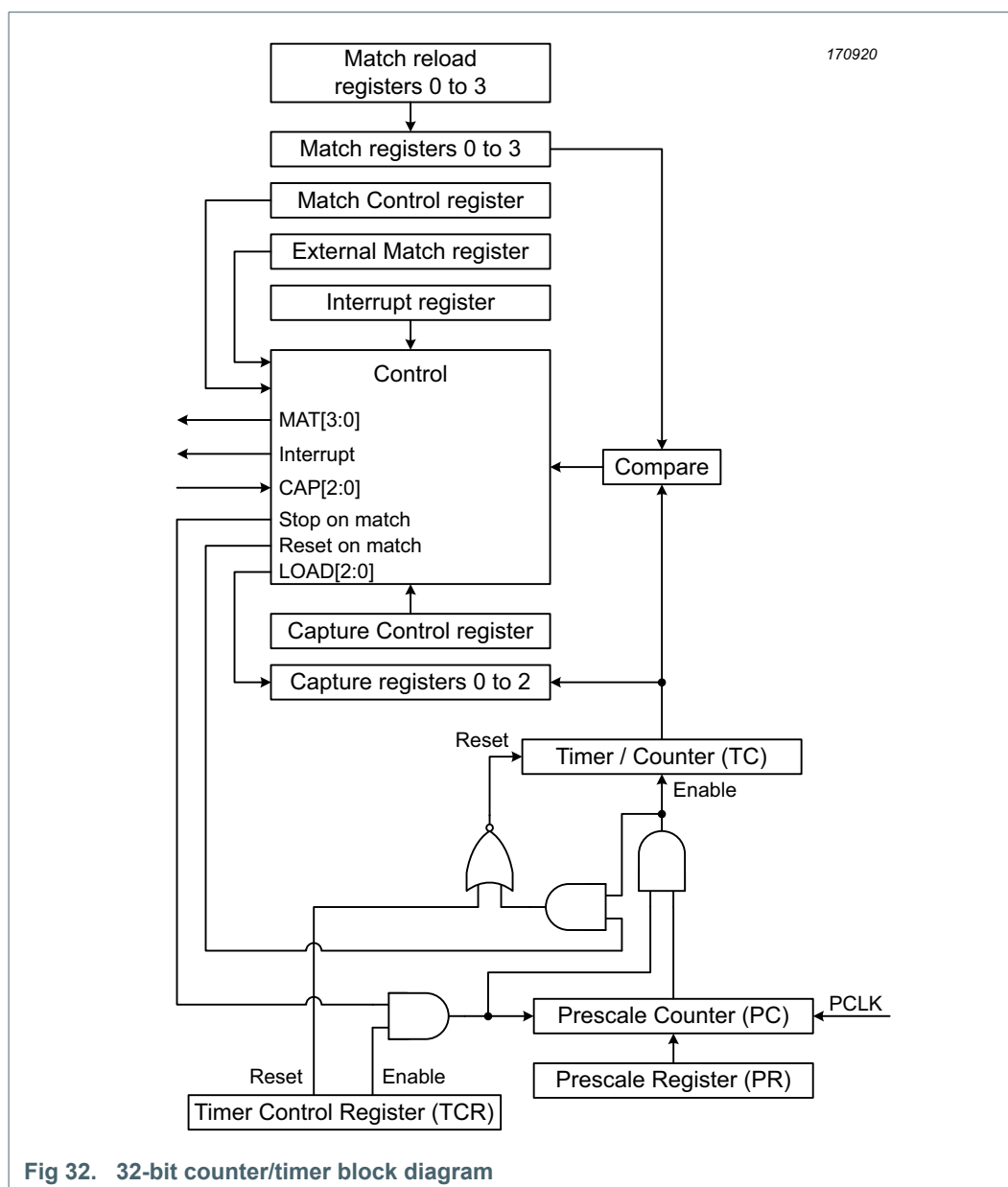
When a match register equals the timer counter (TC), the corresponding match output can either toggle, go LOW, go HIGH, or do nothing. The External Match Register (EMR) and the PWM Control Register (PWMCON) control the functionality of this output.

### 16.4.3 Applications

- Interval timer for counting internal events.
- Pulse Width Modulator via match outputs.
- Pulse Width Demodulator via capture input.
- Free running timer.

### 16.4.4 Architecture

[Figure 32](#) shows the block diagram for the timer.



## 16.5 Pin description

[Table 204](#) gives a brief summary of each of the Timer/Counter related pins.

**Table 204. Timer/Counter pin description**

Function	Type	Connect to	Use register	Description
To_CAPx	Input	Any pin	PINASSIGN3	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 <a href="#">Section 16.6.11</a> .
To_MATx	Output	Any pin	PINASSIGN4	External Match Output - When a match register (MR3:0) 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.

### 16.5.1 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.

## 16.6 Register description

The timer/counter contains the registers shown in [Table 205](#).

**Table 205. Register overview: CTIMER (register base addresses 0x4003 8000)**

Name	Access	Offset	Description	Reset value <sup>[1]</sup>	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	<a href="#">16.6.1</a>
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	<a href="#">16.6.2</a>
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	<a href="#">16.6.3</a>
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	<a href="#">16.6.4</a>
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	<a href="#">16.6.5</a>
MCR	R/W	0x14	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	<a href="#">16.6.6</a>
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	<a href="#">16.6.7</a>
MR1	R/W	0x1C	Match Register 1. See MR0 description.	0	<a href="#">16.6.7</a>
MR2	R/W	0x20	Match Register 2. See MR0 description.	0	<a href="#">16.6.7</a>
MR3	R/W	0x24	Match Register 3. See MR0 description.	0	<a href="#">16.6.7</a>
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	<a href="#">16.6.8</a>
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	<a href="#">16.6.9</a>
CR1	RO	0x30	Capture Register 1. See CR0 description.	0	<a href="#">16.6.9</a>
CR2	RO	0x34	Capture Register 2. See CR0 description.	0	<a href="#">16.6.9</a>
EMR	R/W	0x3C	External Match Register. The EMR controls the match function and the external match pins.	0	<a href="#">16.6.10</a>
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	<a href="#">16.6.11</a>
PWMC	R/W	0x74	PWM Control Register. The PWMCON enables PWM mode for the external match pins.	0	<a href="#">16.6.12</a>
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	<a href="#">16.6.13</a>



Table 205. Register overview: CTIMER (register base addresses 0x4003 8000)

Name	Access	Offset	Description	Reset value <sup>[1]</sup>	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	<a href="#">16.6.13</a>
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	<a href="#">16.6.13</a>
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	<a href="#">16.6.13</a>

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

### 16.6.1 Interrupt Register

The Interrupt Register consists of 4 bits for the match interrupts and 3 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.

Table 206. Interrupt Register (IR, offset 0x000) 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
31:7	-	Reserved. Read value is undefined, only zero should be written.	-

### 16.6.2 Timer Control Register

The Timer Control Register (TCR) is used to control the operation of the Timer/Counter.

Table 207. Timer Control Register (TCR, offset 0x004) 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.	NA

### 16.6.3 Timer Counter registers

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 208. Timer counter registers (TC, offset 0x08) bit description

Bit	Symbol	Description	Reset value
31:0	TCVAL	Timer counter value.	0

### 16.6.4 Prescale register

The 32-bit Prescale register specifies the maximum value for the Prescale Counter.

**Table 209. Timer prescale registers (PR, offset 0x00C) bit description**

Bit	Symbol	Description	Reset value
31:0	PRVAL	Prescale counter value.	0

### 16.6.5 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 210. Timer prescale counter registers (PC, offset 0x010) bit description**

Bit	Symbol	Description	Reset value
31:0	PCVAL	Prescale counter value.	0

### 16.6.6 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 211. Match Control Register (MCR, offset 0x014) 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 211. Match Control Register (MCR, offset 0x014) 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.	NA
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.	NA

### 16.6.7 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 212. Timer match registers (MR[0:3], offset [0x018:0x024]) bit description

Bit	Symbol	Description	Reset value
31:0	MATCH	Timer counter match value.	0

### 16.6.8 Capture Control Register

The Capture Control Register is used to control whether one of the three 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.

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 2 CAP inputs.

Table 213. Capture Control Register (CCR, offset 0x028) 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
31:9	-	Reserved. Read value is undefined, only zero should be written.	NA

### 16.6.9 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 214. Timer capture registers (CR[0:2], offsets [0x02C:0x034]) bit description

Bit	Symbol	Description	Reset value
31:0	CAP	Timer counter capture value.	0

### 16.6.10 External Match Register

The External Match Register provides both control and status of the external match pins.

If the match outputs are configured as PWM output, the function of the external match registers is determined by the PWM rules ([Section 16.7.1 “Rules for single edge controlled PWM outputs” on page 224](#)).

Table 215. Timer external match registers (EMR, offset 0x03C) 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 SWM.	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 SWM.	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 SWM.	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 SWM.	0
5:4	EMC0		External Match Control 0. Determines the functionality of External Match 0.	00
		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.	00
		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.	00
		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.	00
		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.	NA

### 16.6.11 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/LOW 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 216. Count Control Register (CTCR, offset 0x070) 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.	00
		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 00, these bits select which CAP pin is sampled for clocking. <b>Note:</b> 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 2 CAP inputs.	0
		0x0	Channel 0. CAP0	
		0x1	Channel 1. CAP1	
		0x2	Channel 2. CAP2	
		0x3	Reserved	
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

Table 216. Count Control Register (CTCR, offset 0x070) 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. Values 0x6 to 0x7 are reserved.	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	Reserved	
		0x7	Reserved	
31:8	-	-	Reserved. Read value is undefined, only zero should be written.	NA

### 16.6.12 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 217. PWM Control Register (PWMC, offset 0x074) 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 217. PWM Control Register (PWMC, offset 0x074) bit description

Bit	Symbol	Value	Description	Reset value
3	PWMEN3		PWM mode enable for channel3. <b>Note:</b> 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.	NA

### 16.6.13 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. Software will then have one full counter cycle to modify the contents of the Match Shadow Register(s) before the next reload occurs.

Table 218. Timer 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

## 16.7 Functional description

Figure 33 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 34 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.

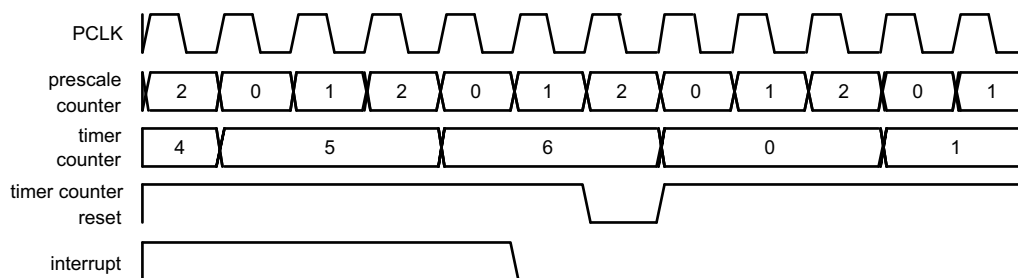


Fig 33. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled

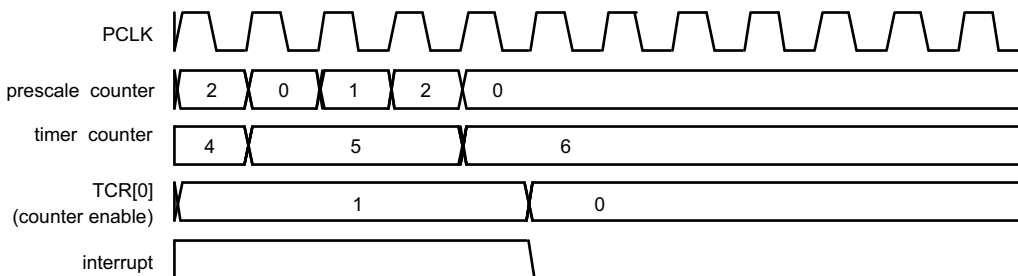


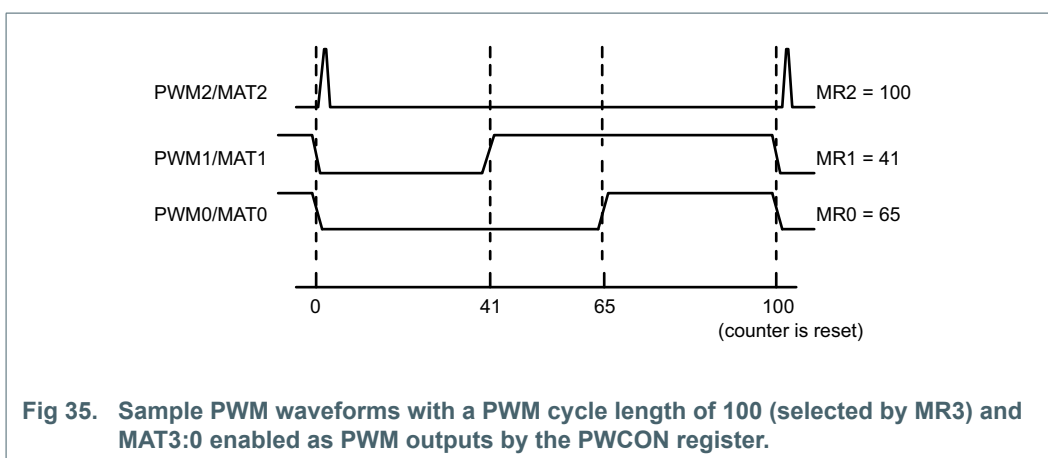
Fig 34. A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled

### 16.7.1 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.



### 17.1 How to read this chapter

---

The watchdog timer is identical on all LPC802 parts.

### 17.2 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 / watchdog feed sequence can optionally be protected such that it can only be performed after the “warning interrupt” time is reached.
- Flag to indicate Watchdog reset.
- The Watchdog clock (WDCLK) source is the low power oscillator.
- The Watchdog timer can be configured to run in deep-sleep or power-down mode.
- Debug mode.

### 17.3 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 63](#).
- Enable the WWDT clock source (the low power oscillator) in the PDRUNCFG register ([Table 82](#)). 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 79](#)).

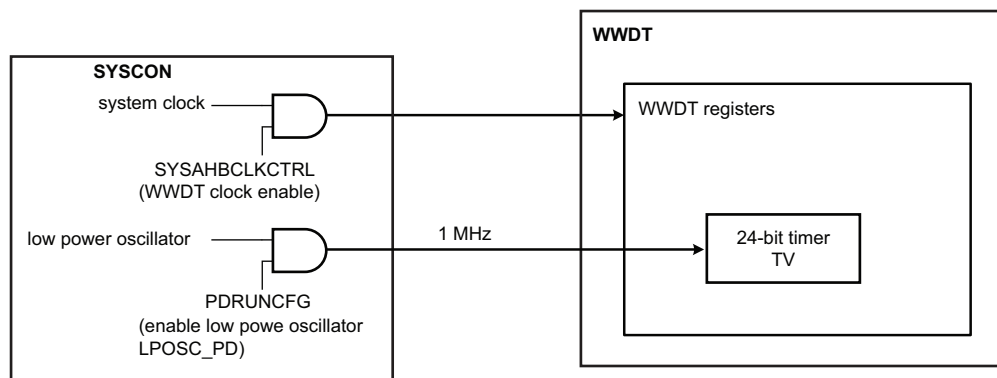


Fig 36. USART clocking

## 17.4 Pin description

The WWDT has no external pins.

## 17.5 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.
- Set the Watchdog timer update mode (WDPROTECT) in the MOD register after a delay of three WDCLK clock cycles.

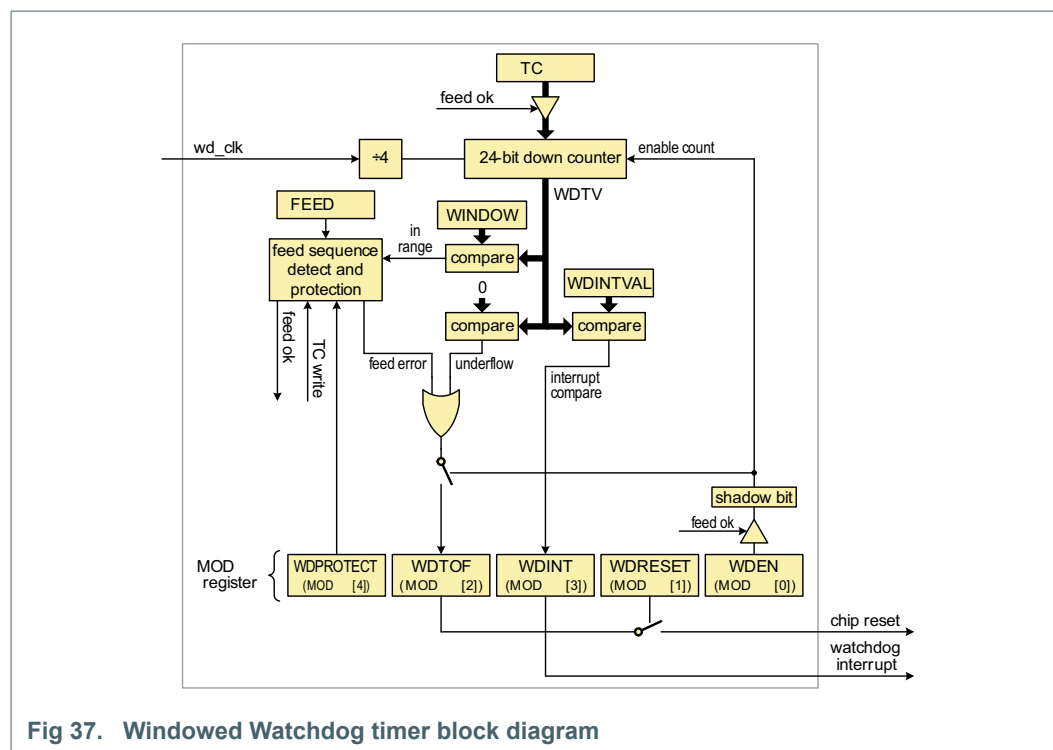
- The Watchdog must be fed again before the Watchdog counter reaches zero in order to prevent a watchdog event. If a window value is programmed, the feed must also occur after the watchdog counter passes that value.

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.

### 17.5.1 Block diagram

The block diagram of the Watchdog is shown below in the [Figure 37](#). The synchronization logic (PCLK - WDCLK) is not shown in the block diagram.



### 17.5.2 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 6](#)). The WDCLK is used for the watchdog timer counting and is derived from the low power oscillator.

The synchronization logic between the two clock domains works as follows: When the MOD and TC registers are updated by APB operations, the new value will take effect in 3 WDCLK cycles on the logic in the WDCLK clock domain.

When the watchdog timer is counting on WDCLK, the synchronization logic will first lock the value of the counter on WDCLK and then synchronize it with PCLK, so that the CPU can read the WDTV register.

**Remark:** Because of the synchronization step, software must add a delay of three WDCLK clock cycles between the feed sequence and the time the WDPROTECT bit is enabled in the MOD register. The length of the delay depends on the watchdog clock WDCLK.

### 17.5.3 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
- Performing the WWDT reload or WWDT feed sequence

#### 17.5.3.1 Disabling the WWDT clock source

If bit 5 in the WWDT MOD register is set, the WWDT clock source is locked and can not be disabled either by software or by hardware when sleep, deep-sleep or power-down modes are entered. Therefore, the user must ensure that the low power oscillator for each power mode is enabled **before** setting bit 5 in the MOD register.

In deep power-down mode, no clock locking mechanism is in effect because no clocks are running. However, an additional lock bit in the PMU can be set to prevent the part from even entering Deep power-down mode (see [Table 151](#)).

#### 17.5.3.2 Changing the WWDT reload value

If bit 4 is set in the WWDT MOD register, the watchdog reload or watchdog feed sequence can be performed only after the watchdog timer is below the value of WDWARNING and WDWINDOW.

The reload overwrite lock mechanism can only be disabled by a reset of any type.

## 17.6 Register description

The Watchdog Timer contains the registers shown in [Table 219](#).

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

**Table 219. 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	<a href="#">Table 220</a>
TC	R/W	0x004	Watchdog timer constant register. This 24-bit register determines the time-out value.	0xFF	<a href="#">Table 222</a>
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	<a href="#">Table 223</a>
TV	RO	0x00C	Watchdog timer value register. This 24-bit register reads out the current value of the Watchdog timer.	0xFF	<a href="#">Table 224</a>
-	-	0x010	Reserved	-	-
WARNINT	R/W	0x014	Watchdog Warning Interrupt compare value.	0	<a href="#">Table 225</a>
WINDOW	R/W	0x018	Watchdog Window compare value.	0xFF FFFF	<a href="#">Table 226</a>

### 17.6.1 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 220. 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 220. 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	Flexible. The watchdog reload or watchdog feed sequence can be performed when the watchdog timer is below the value of WDWINDOW.	
		1	Threshold. The watchdog reload or watchdog feed sequence can be performed only after the watchdog timer is below the value of WDWARNING and WDWINDOW.	
5	LOCK		A 1 in this bit prevents disabling or powering down the low power oscillator. This bit can be set once by software and is only cleared by any reset.	0
31:6	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Once the **WDEN**, **WDPROTECT**, or **WDRESET** bits are set they can not be cleared by software. Both flags are cleared by an external reset or a Watchdog timer reset.

**WDTOF** The Watchdog time-out flag is set when the Watchdog times out, when a feed error occurs, or when PROTECT =1 and an attempt is made to write to the TC register. This flag is cleared by software writing a 0 to this bit.

**WDINT** The Watchdog interrupt flag is set when the Watchdog counter reaches the value specified by WARNINT. This flag is cleared when any reset occurs, and is cleared by software by writing a 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 low power oscillator. The low power oscillator can be configured to keep running in Sleep, deep-sleep modes, and power-down modes.

If a watchdog interrupt occurs in sleep, deep-sleep mode, or power-down mode, and the WWDT interrupt is enabled in the NVIC, the device will wake up. Note that in deep-sleep and power-down modes, the WWDT interrupt must be enabled in the STARTERP1 register in addition to the NVIC.

See the following registers:

[Table 79 “Start logic 1 interrupt wake-up enable register \(STARTERP1, address 0x4004 8214\) bit description”](#)

Table 221. 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.

### 17.6.2 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 perform the watchdog reload or watchdog feed sequence before the watchdog timer is below the values of WDWARNINT and WDWINDOW will cause a watchdog feed error and set the WDTOF flag.

Table 222. 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

### 17.6.3 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 223. 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

### 17.6.4 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 224. 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

### 17.6.5 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 is less than or equal to 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 225. 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

### 17.6.6 Watchdog Timer Window register

The WINDOW register determines the highest WDTV value allowed when a watchdog feed is performed. If a feed sequence occurs when WDTV is greater than the value in WINDOW, a watchdog event will occur.

WINDOW resets to the maximum possible WDTV value, so windowing is not in effect.

Table 226. 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

## 17.7 Functional description

The following figures illustrate several aspects of Watchdog Timer operation.

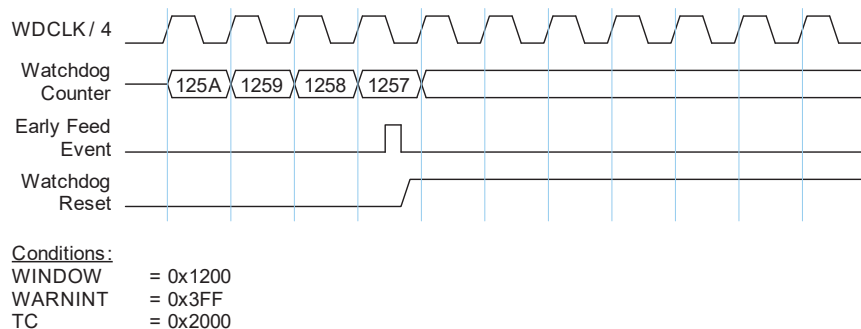


Fig 38. Early watchdog feed with windowed mode enabled

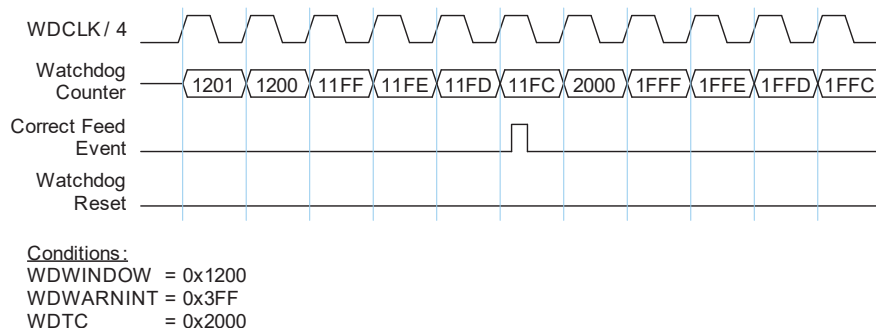


Fig 39. Correct watchdog feed with windowed mode enabled

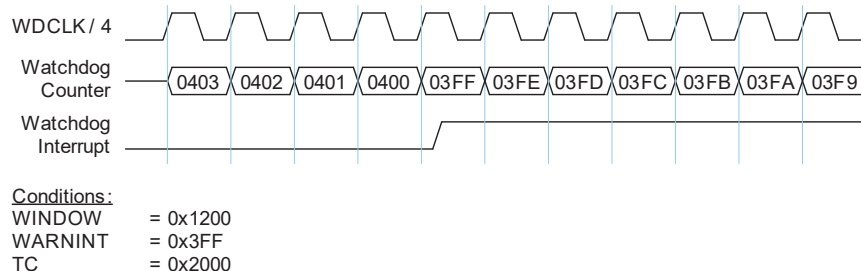


Fig 40. Watchdog warning interrupt

### 18.1 How to read this chapter

---

The self-wake-up timer is available on all LPC802 parts.

### 18.2 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 supports three clock sources: The FRO, the internal low-power oscillator, or the WKCLKIN pin. The low-power oscillator and the external clock can be configured to be valid clock sources in all power modes except 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 different low power modes or for general-purpose timing.

### 18.3 Basic configuration

---

- In the SYSAHBCLKCTRL register, set bit 9 ([Table 63](#)) to enable the clock to the register interface.
- Clear the WKT reset using the PRESETCTRL register ([Table 64](#)).
- The WKT interrupt is connected to interrupt #15 in the NVIC. See [Table 38](#).
- Enable the low power oscillator in the PDRUNCFG register if used as the clock source for the timer.
- Enable the FRO and FRO output in the PDRUNCFG register if used as the clock source for the timer ([Table 82](#)).
- To use an external clock source for the self-wake-up timer, enable the clock input for pin PIO0\_11 by setting the switch matrix PINENABLE0 register and enable the external clock option in the self-wake-up timer CTRL register (see [Table 228](#)).

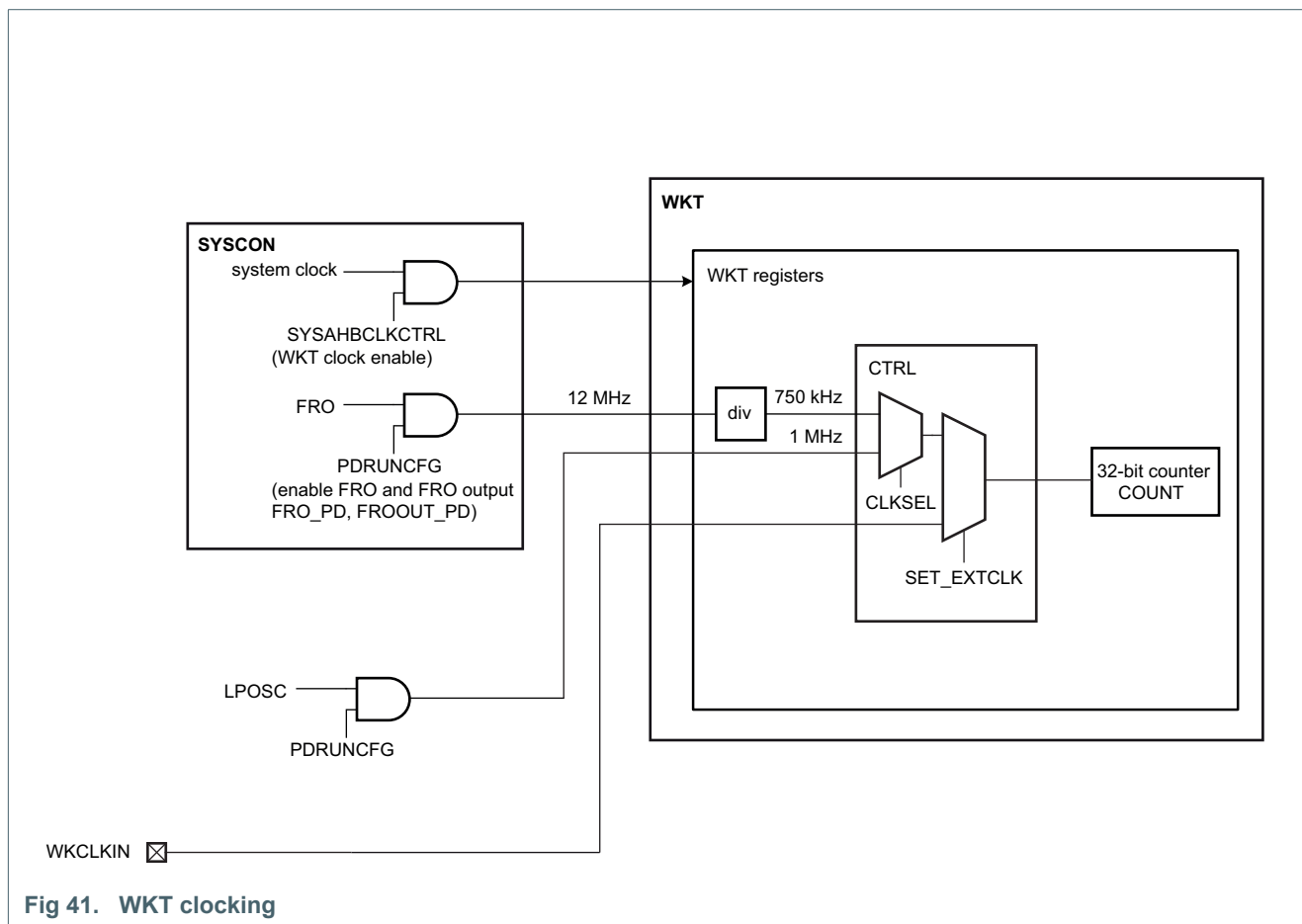


Fig 41. WKT clocking

## 18.4 Pin description

The WKT can use a clock input on the external pin PIO0\_11 for clocking the wake-up timer. Select the external clock source by setting bit SET\_EXTCLK in the CTRL register (see [Table 228](#)).

## 18.5 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.

### 18.5.1 WKT clock sources

The self-wake-up timer can be clocked from two alternative internal clock sources and one external clock:

- A 750 kHz clock derived from the FRO oscillator. This is the default clock.
- A 1 MHz, low-power clock with a dedicated on-chip oscillator as clock source.
- An external clock on the WKCLKIN 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 1 MHz, low-power clock, sourced from a dedicated oscillator. When the FRO clock is shut-down, the 1 MHz clock can be configured to be available during low-power modes (deep sleep and power down mode).

An external clock on the WKCLKIN pin can be used to time the self-wake-up timer in low power modes, except deep power-down.

## 18.6 Register description

Table 227. 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	<a href="#">Table 228</a>
COUNT	R/W	0xC	Counter register.	-	<a href="#">Table 229</a>

### 18.6.1 Control register

The WKT interrupt must be enabled in the NVIC to wake up the part using the self-wake-up counter.

Table 228. Control register (CTRL, address 0x4000 8000) bit description

Bit	Symbol	Value	Description	Reset value
0	CLKSEL		Select the self-wake-up timer clock source. <b>Remark:</b> 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 $\mu$ s increments. <b>Remark:</b> 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 1 MHz clock and provides time-out periods of up to approximately 71.6 minutes in 1 $\mu$ s increments. <b>Remark:</b> This clock is not available in deep power-down mode. Prior to use, enable the low-power oscillator.	
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. Writing a 1 clears this status bit.	

Table 228. Control register (CTRL, address 0x4000 8000) bit description

Bit	Symbol	Value	Description	Reset value
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 WKCLKIN pin.	
31:4	-		Reserved.	-

### 18.6.2 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 229. 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.	-



### 19.1 How to read this chapter

The MRT is available on all LPC802 parts.

### 19.2 Features

- 31-bit interrupt timer
- Two channels independently counting down from individually set values
- Repeat, bus-stall, and one-shot interrupt modes

### 19.3 Basic configuration

Configure the MRT using the following registers:

- In the SYSAHBCLKCTRL register, set bit 10 ([Table 63](#)) to enable the clock to the register interface.
- Clear the MRT reset using the PRESETCTRL register ([Table 64](#)).
- The global MRT interrupt is connected to interrupt #10 in the NVIC.

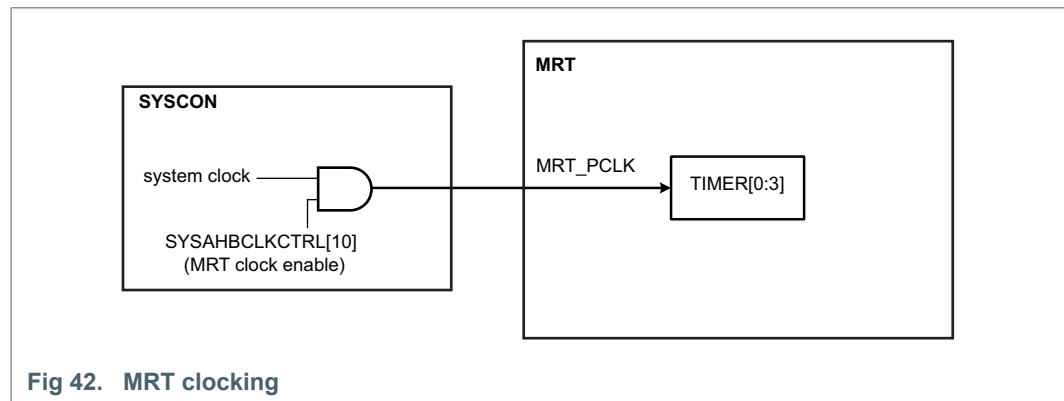


Fig 42. MRT clocking

### 19.4 Pin description

The MRT has no configurable pins.

### 19.5 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 19.5.1](#).

- One-shot interrupt mode. See [Section 19.5.2](#).
- Bus-stall mode.

The modes for each timer are set in the timer's control register. See [Table 233](#).

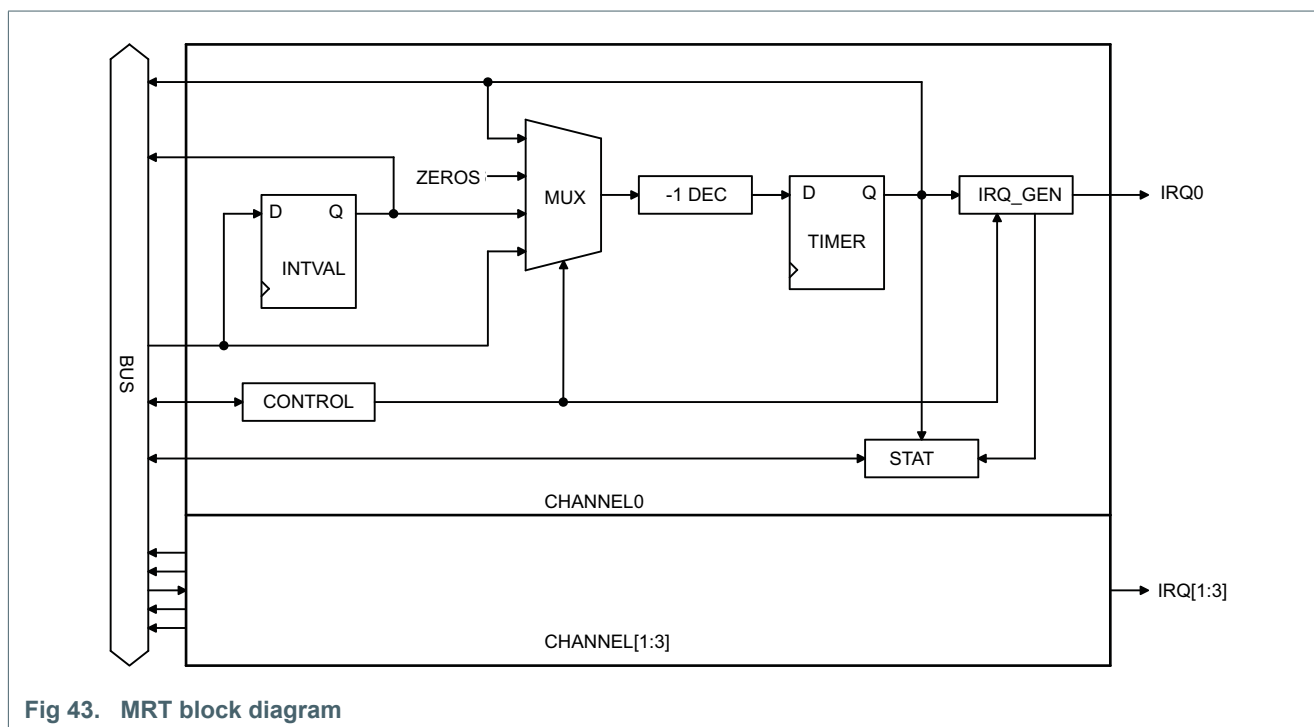


Fig 43. MRT block diagram

### 19.5.1 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.

### 19.5.2 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.

### 19.5.3 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  $\mu$ s 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.

## 19.6 Register description

The reset values shown in [Table 230](#) are POR reset values.

**Table 230. 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	<a href="#">Table 231</a>
TIMER0	RO	0x4	MRT0 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	<a href="#">Table 232</a>
CTRL0	R/W	0x8	MRT0 Control register. This register controls the MRT0 modes.	0	<a href="#">Table 233</a>
STAT0	R/W	0xC	MRT0 Status register.	0	<a href="#">Table 234</a>
INTVAL1	R/W	0x10	MRT1 Time interval value register. This value is loaded into the TIMER1 register.	0	<a href="#">Table 231</a>
TIMER1	RO	0x14	MRT1 Timer register. This register reads the value of the down counter.	0x7FFF FFFF	<a href="#">Table 232</a>
CTRL1	R/W	0x18	MRT1 Control register. This register controls the MRT1 modes.	0	<a href="#">Table 233</a>
STAT1	R/W	0x1C	MRT1 Status register.	0	<a href="#">Table 234</a>
MODCFG	R/W	0xF0	Module Configuration. This register provides information about this particular MRT instance, and allows choosing an overall mode for the idle channel feature.	0x0	<a href="#">Table 235</a>
IDLE_CH	R	0xF4	Idle channel register. This register returns the number of the first idle channel.	0	<a href="#">Table 236</a>
IRQ_FLAG	R/W	0xF8	Global interrupt flag register	0	<a href="#">Table 237</a>

### 19.6.1 Time interval register

This register contains the MRT load value and controls how the timer is reloaded. The load value is IVALUE -1.

**Table 231. 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		<p>Time interval load value. This value is loaded into the TIMERN register and the MRTn starts counting down from IVALUE -1.</p> <p>If the timer is idle, writing a non-zero value to this bit field starts the timer immediately.</p> <p>If the timer is running, writing a zero to this bit field does the following:</p> <ul style="list-style-type: none"> <li>• If LOAD = 1, the timer stops immediately.</li> <li>• If LOAD = 0, the timer stops at the end of the time interval.</li> </ul>	0

Table 231. Time interval register (INTVAL[0:3], address 0x4000 4000 (INTVAL0) to 0x4000 4030 (INTVAL3)) bit description

Bit	Symbol	Value	Description	Reset value
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.	

### 19.6.2 Timer register

The timer register holds the current timer value. This register is read-only.

Table 232. 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

### 19.6.3 Control register

The control register configures the mode for each MRT and enables the interrupt.

Table 233. 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

### 19.6.4 Status register

This register indicates the status of each MRT.

Table 234. 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

### 19.6.5 Module Configuration register

The MODCFG register provides the configuration (number of channels and timer width) for this MRT. See [Section 19.6.6 “Idle channel register”](#) for details.

**Table 235. Module Configuration register (MODCFG, offset 0xF0) bit description**

Bit	Symbol	Value	Description	Reset Value
3:0	NOC	-	Identifies the number of channels in this MRT. (2 channels on this device.)	0x2
8:4	NOB	-	Identifies the number of timer bits in this MRT. (31 bits wide on this device.)	0x1F
30:9	-	-	Reserved. Read value is undefined, only zero should be written.	-
31	MULTITASK		Selects the operating mode for the INUSE flags and the IDLE_CH register.	0x0
		0	Hardware status mode. In this mode, the INUSE(n) flags for all channels are reset.	
		1	Multi-task mode.	

### 19.6.6 Idle channel register

The idle channel register can be used to assist software in finding available channels in the MRT. This allows more flexibility by not giving hard assignments to software that makes use of the MRT, without the need to search for an available channel. Generally, IDLE\_CH returns the lowest available channel number.

IDLE\_CH can be used in two ways, controlled by the value of the MULTITASK bit in the MODCFG register. MULTITASK affects both the function of IDLE\_CH, and the function of the INUSE bit for each MRT channel as follows:

- MULTITASK = 0: hardware status mode. The INUSE flags for all MRT channels are reset. IDLECH returns the lowest idle channel number. A channel is considered idle if its RUN flag = 0, and there is no interrupt pending for that channel.
- MULTITASK = 1: multi-task mode. In this mode, the INUSE flags allow more control over when MRT channels are released for further use. When IDLE\_CH is read, returning a channel number of an idle channel, the INUSE flag for that channel is set by hardware. That channel will not be considered idle until its RUN flag = 0, there is no interrupt pending, and its INUSE flag = 0. This allows reserving an MRT channel with a single register read, and no need to start the channel before it is no longer considered idle by IDLE\_CH. It also allows software to identify a specific MRT channel that it can use, then use it more than once without releasing it, removing the need to ask for an available channel for every use.

**Table 236. 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

### 19.6.7 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 237. 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.	
31:2	-		Reserved.	0



### 20.1 How to read this chapter

The SysTick timer is available on all LPC802 parts.

### 20.2 Features

- Simple 24-bit timer.
- Uses dedicated exception vector.
- Clocked internally by the system clock or the system clock/2.

### 20.3 Basic configuration

The system tick timer is configured using the following registers:

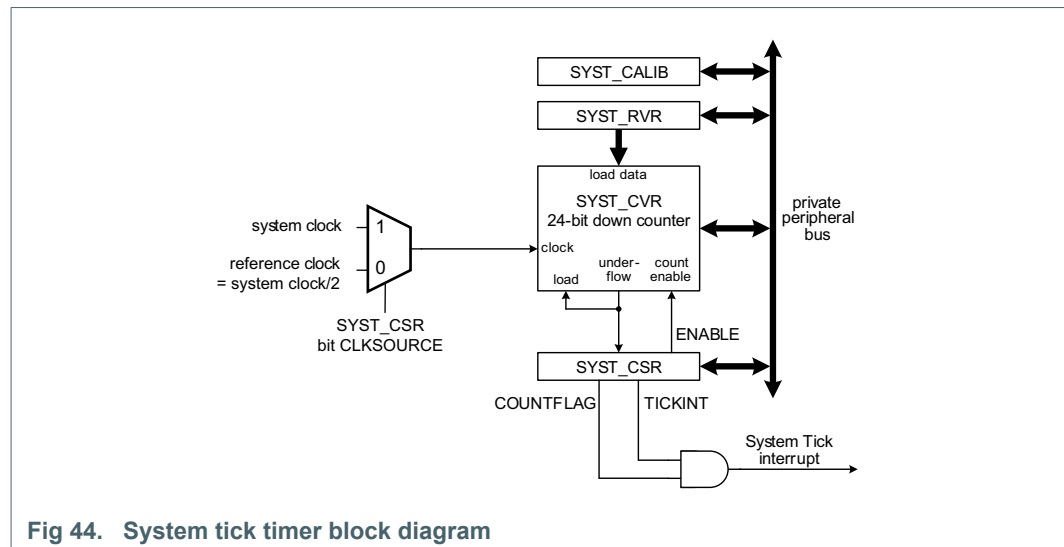
1. The system tick timer is enabled through the SysTick control register ([Table 239](#)). The system tick timer clock is fixed to half of the system clock frequency.
2. Enable the clock source for the SysTick timer in the SYST\_CSR register ([Table 239](#)).
3. The calibration value of the SysTick timer is contained in the SYSTCKCAL register in the system configuration block SYSCON (see [Table 74](#)).

### 20.4 Pin description

The SysTick has no configurable pins.

### 20.5 General description

The block diagram of the SysTick timer is shown in [Figure 44](#).



The SysTick timer is an integral part of the Cortex-M0+. The SysTick timer is intended to generate a fixed 10 millisecond interrupt for use by an operating system or other system management software.

Since the SysTick timer is a part of the Cortex-M0+, it facilitates porting of software by providing a standard timer that is available on Cortex-M0 based devices. The SysTick timer can be used for:

- An RTOS tick timer which fires at a programmable rate (for example 100 Hz) and invokes a SysTick routine.
- A high-speed alarm timer using the core clock.
- A simple counter. Software can use this to measure time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

Refer to [Ref. 5](#) for details.

## 20.6 Register description

The SysTick timer registers are located on the Arm Cortex-M0+ private peripheral bus (see [Figure 2](#)), and are part of the Arm Cortex-M0+ core peripherals. For details, see [Ref. 5](#).

**Table 238. Register overview: SysTick timer (base address 0xE000 E000)**

Name	Access	Address offset	Description	Reset value <sup>[1]</sup>
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.

### 20.6.1 System Timer Control and status register

The SYST\_CSR register contains control information for the SysTick timer and provides a status flag. This register is part of the Arm Cortex-M0+ core system timer register block. For a bit description of this register, see [Ref. 5](#).

This register determines the clock source for the system tick timer.

**Table 239. 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

### 20.6.2 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 240. 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

### 20.6.3 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 241. 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

### 20.6.4 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 74](#)).

**Table 242. System Timer Calibration value register (SYST\_CALIB, 0xE000 E01C) bit description**

Bit	Symbol	Value	Description	Reset value
23:0	TENMS		See <a href="#">Ref. 5</a> .	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 <a href="#">Ref. 5</a> .	0
31	NOREF		See <a href="#">Ref. 5</a> .	0

## 20.7 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 6](#)) 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.

### 20.7.1 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).

$RELOAD = (\text{system tick clock frequency} \times 10 \text{ ms}) - 1 = (20 \text{ MHz} \times 10 \text{ ms}) - 1 = 200000 - 1 = 199999 = 0x00030D3F$ .

### 21.1 How to read this chapter

The ADC is available on all parts. The number of available ADC channels depends on the package type.

**Table 243. Pinout summary**

Package	ADC channels available
TSSOP16	ADC_0, ADC_1, ADC_4, ADC_5, ADC_6, ADC_9, ADC_10, ADC_11.
TSSOP20 package with VDDIO	ADC_0, ADC_1, ADC_2, ADC_3, ADC_4, ADC_5, ADC_6, ADC_7, ADC_8, ADC_10, ADC_11.
TSSOP20 (SS)	ADC_0 to ADC_11.
HVQFN33	ADC_0 to ADC_11.

### 21.2 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 (GND) to VREFP (typically 3 V; not to exceed VDDA voltage level).
- 12-bit conversion rate of up to 480 Ksamples/s.
- Burst conversion mode for single or multiple inputs.
- Synchronous or asynchronous operation. Asynchronous operation maximizes flexibility in choosing the ADC clock frequency. Synchronous mode minimizes trigger latency and can eliminate uncertainty and jitter in response to a trigger.

### 21.3 Basic configuration

Configure the ADC as follows:

- Use the PDRUNCFG register to power the ADC. See [Table 82](#). 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 248](#) and [Section 21.7.5](#).
- Use the SYSAHBCLKCTRL register ([Table 63](#)) to enable the clock to the ADC register interface and the ADC clock, if asynchronous operating mode is selected.
- Use ADCASYNCCLKSEL and ADCASYNCDIV registers to control the clock to the ADC clock, if asynchronous operating mode is selected.

- The ADC block creates four interrupts with individual entries in the NVIC. See [Table 38](#).
- The ADC analog inputs are enabled in the switch matrix block. See [Table 97](#).
- The power to the ADC block is controlled by the PDRUNCFG register in the SYSCON block. See [Table 82](#).

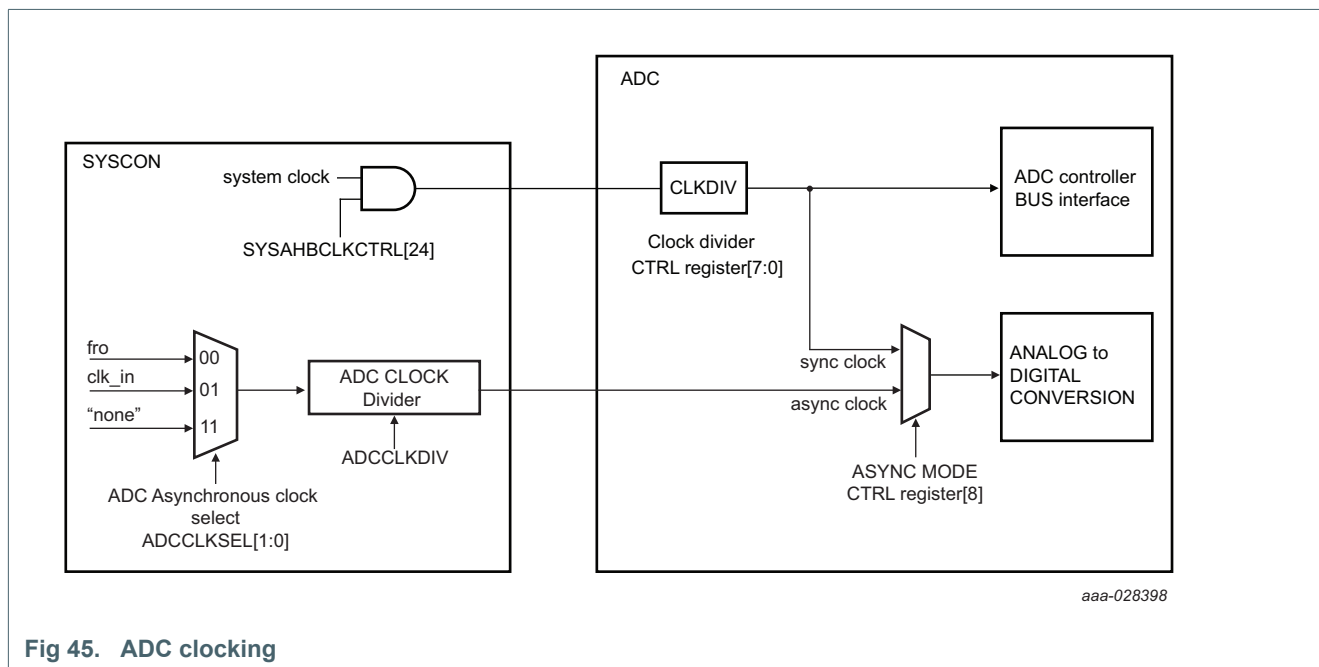


Fig 45. ADC clocking

### 21.3.1 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 21.7.2.1 “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  $(4095 \times VIN) / (VREFP - VREFN)$ . The result of an input voltage below VREFN is 0, and the result of an input voltage above VREFP is 4095 (0xFFF).

To perform a single ADC conversion for ADC0 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 15 MHz and select a CLKDIV value of 0 for a sampling rate of 480 Ksamples/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.

### 21.3.2 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 21.3.3](#).

To perform a single-step conversion on the first four channels of ADC0 triggered by a rising edge on PINT0 pin, follow these steps:

1. Enable the analog functions ADC\_0 to ADC\_3 through the switch matrix. See [Table 246](#).
2. Configure the system clock to be 15 MHz and select a CLKDIV value of 0 for a sampling rate of 480 ksamples/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 ADC0 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.

### 21.3.3 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 249 “A/D Conversion Sequence A Control Register \(SEQA\\_CTRL, address 0x4001 C008\) bit description”](#)
- [Table 250 “A/D Conversion Sequence B Control Register \(SEQB\\_CTRL, address 0x4001 C00C\) bit description”](#)

Table 244. 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.
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.

### 21.3.4 Sample Time

The analog input from the selected channel is sampled at the start of each new A/D conversion. The default (and shortest) duration of the sample period is 6.5 ADC clocks. With this default setting, the total duration of each A/D conversion is 31 ADC clocks. Under certain conditions longer sample times may be required. A variety of factors influence the required sampling time:

- Output impedance of the analog source driver.
- Operating conditions.
- ADC clock frequency.
- Selected ADC resolution (10 or 12 bits).
- Number of channels converted and if channel 0 (the slow channel) is included.

## 21.4 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 via the SWM.

**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 (GND) pins provide a positive and negative reference voltage input. The result of the conversion is  $(4095 \times \text{input voltage } V_{IN}) / (V_{REFP} - V_{REFN})$ . The result of an input voltage below VREFN is 0, and the result of an input voltage above VREFP is 4095 (0xFFFF).

When the ADC is not used, tie VREFP to VDD.

**Remark:** For best performance, select VREFP at the same voltage levels as  $V_{DD}$ . When selecting VREFP different from VDD, ensure that the voltage midpoints are the same:



$$(V_{REFP}-V_{REFN})/2 + V_{REFN} = V_{DD}/2$$

Table 245. ADC supply and reference voltage pins

Function	Description
V <sub>REFP</sub>	Positive voltage reference. The VREFP voltage level must be between 2.4 V and V <sub>DDA</sub> . For best performance, select VREFP = V <sub>DDA</sub> and VREFN = V <sub>SSA</sub> .
V <sub>REFN</sub>	GND pin.
V <sub>DDA</sub> = V <sub>DD</sub>	The analog supply voltage is internally connected to V <sub>DD</sub> .
V <sub>SSA</sub> = V <sub>SS</sub>	ADC ground is internally connected to V <sub>SS</sub> .

Table 246. ADC pin description

Function	Direction	Type	Connect to	Use register	Reference	Description
ADC_0	AI	external to pin	PIO0_1	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 0.
ADC_1	AI	external to pin	PIO0_7	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 1.
ADC_2	AI	external to pin	PIO0_14	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 2.
ADC_3	AI	external to pin	PIO0_16	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 3.
ADC_4	AI	external to pin	PIO0_9	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 4.
ADC_5	AI	external to pin	PIO0_8	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 5.
ADC_6	AI	external to pin	PIO0_11	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 6.
ADC_7	AI	external to pin	PIO0_10	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 7.
ADC_8	AI	external to pin	PIO0_15	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 8.
ADC_9	AI	external to pin	PIO0_17	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 9.
ADC_10	AI	external to pin	PIO0_13	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 10.
ADC_11	AI	external to pin	PIO0_4	PINENABLE0	<a href="#">Table 97</a>	Analog input channel 11.

### 21.4.1 ADC versus 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 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.

## 21.5 General description

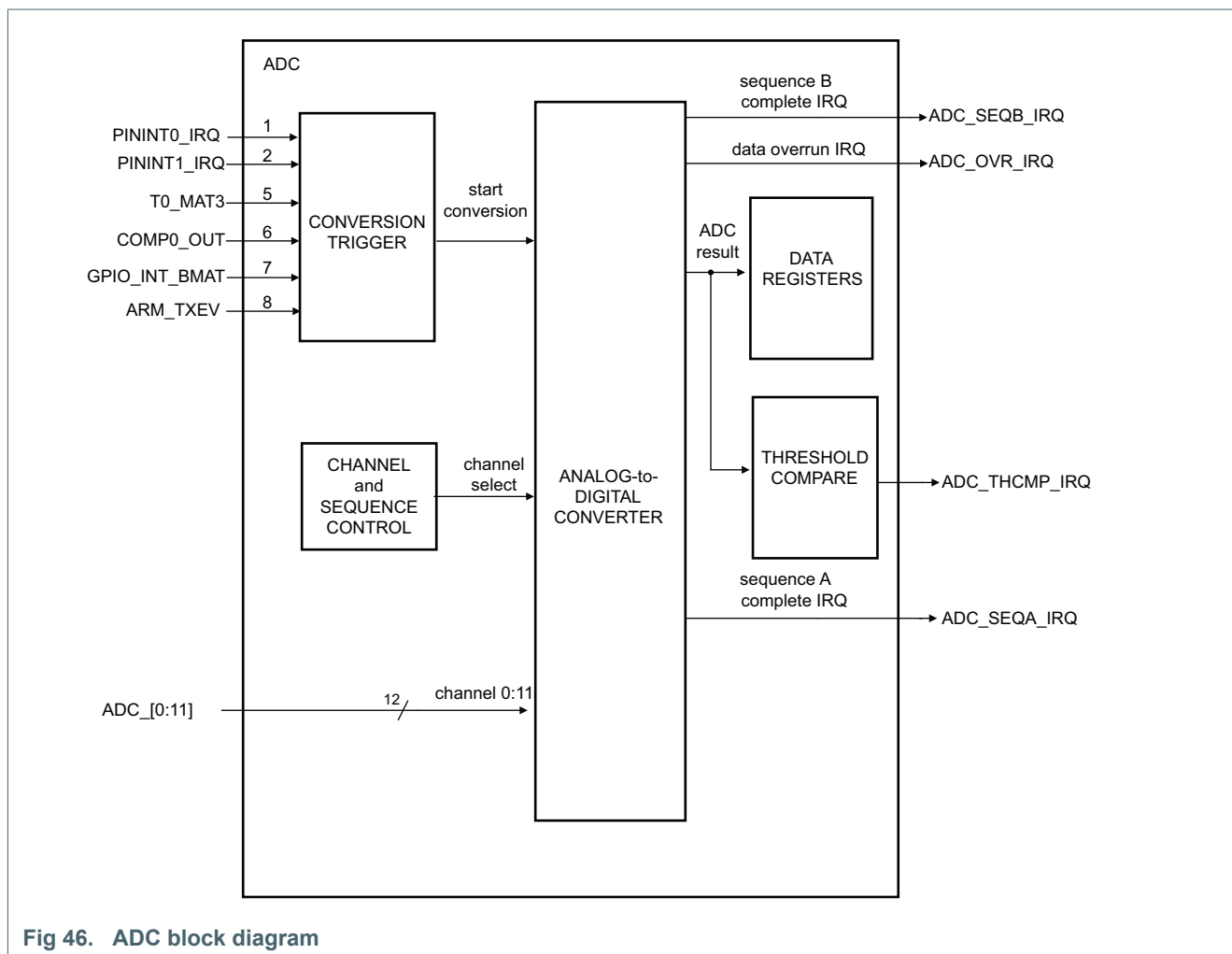


Fig 46. ADC block diagram

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 six hardware triggers.

Once the triggers are set up (software and hardware triggers can be mixed), the ADC runs through the pre-defined conversion sequence, converting a sample whenever a trigger signal arrives, until the sequence is disabled.

The ADC controller uses the system clock as a bus clock. The ADC clock is derived from the system clock, if the synchronous operating mode is selected. A programmable divider is included to scale the system clock to the maximum ADC clock rate of 15 MHz. The ADC clock drives the successive approximation process. In the asynchronous mode, an independent clock source is available for use as the ADC clock source. The maximum ADC clock rate in asynchronous mode is 15 MHz.

A fully accurate conversion requires 31 of these ADC clocks.

## 21.6 Register description

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

**Table 247. 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	<a href="#">Table 248</a>
-	-	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	<a href="#">Table 249</a>
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	<a href="#">Table 250</a>
SEQA_GDAT	RO	0x010	A/D Sequence-A Global Data Register. This register contains the result of the most recent A/D conversion performed under sequence-A	NA	<a href="#">Table 251</a>
SEQB_GDAT	RO	0x014	A/D Sequence-B Global Data Register. This register contains the result of the most recent A/D conversion performed under sequence-B	NA	<a href="#">Table 252</a>
DAT0	RO	0x020	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0.	NA	<a href="#">Table 253</a>
DAT1	RO	0x024	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.	NA	<a href="#">Table 253</a>
DAT2	RO	0x028	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.	NA	<a href="#">Table 253</a>
DAT3	RO	0x02C	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.	NA	<a href="#">Table 253</a>
DAT4	RO	0x030	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.	NA	<a href="#">Table 253</a>
DAT5	RO	0x034	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.	NA	<a href="#">Table 253</a>
DAT6	RO	0x038	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.	NA	<a href="#">Table 253</a>
DAT7	RO	0x03C	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	<a href="#">Table 253</a>
DAT8	RO	0x040	A/D Channel 8 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	<a href="#">Table 253</a>
DAT9	RO	0x044	A/D Channel 9 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	<a href="#">Table 253</a>
DAT10	RO	0x048	A/D Channel 10 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	<a href="#">Table 253</a>
DAT11	RO	0x04C	A/D Channel 11 Data Register. This register contains the result of the most recent conversion completed on channel 7.	NA	<a href="#">Table 253</a>
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	<a href="#">Table 254</a>

Table 247. Register overview: ADC (base address 0x4001 C000 )

Name	Access	Address offset	Description	Reset value	Reference
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	<a href="#">Table 255</a>
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	<a href="#">Table 256</a>
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	<a href="#">Table 257</a>
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	<a href="#">Table 258</a>
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	<a href="#">Table 259</a>
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	<a href="#">Table 260</a>

### 21.6.1 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 248. A/D Control Register (CTRL, addresses 0x4001 C000) bit description

Bit	Symbol	Value	Description	Reset value
7:0	CLKDIV		In synchronous mode only, 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 15 MHz for 480 Ksamples/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. This field is ignored in the asynchronous operating mode.	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 be 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

Table 248. A/D Control Register (CTRL, addresses 0x4001 C000) bit description

Bit	Symbol	Value	Description	Reset value
10	LPWRMODE		<p>Select low-power ADC mode.</p> <p>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.</p> <p>Using this mode can save an appreciable amount of current when conversions are required relatively infrequently.</p> <p>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.</p> <p><b>Remark:</b> This mode will NOT power-up the ADC when the ADC analog block is powered down in the system control block.</p>	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
31:30	-		Reserved.	0

### 21.6.2 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 21.3.1 “Perform a single ADC conversion using a software trigger”](#).

**Remark:** Set the BURST and SEQU\_ENA bits at the same time.

Table 249. A/D Conversion Sequence A Control Register (SEQA\_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
11:0	CHANNELS		<p>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.</p> <p>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.</p> <p><b>Remark:</b> 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.</p>	0x00
14:12	TRIGGER		<p>Selects which of the available hardware trigger sources will cause this conversion sequence to be initiated. Program the trigger input number in this field.</p> <p><b>Remark:</b> 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.</p>	0x0
17:15	-		Reserved.	-
18	TRIGPOL		Select the polarity of the selected input trigger for this conversion sequence.	0
		0	Negative edge. A negative edge launches the conversion sequence on the selected trigger input.	
		1	Positive edge. A positive edge launches the conversion sequence on the selected trigger input.	
19	SYNCBYPASS		<p>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:</p> <p>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.</p> <p>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 an 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.</p>	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		<p>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.</p> <p><b>Remark:</b> 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.</p>	0

Table 249. A/D Conversion Sequence A Control Register (SEQA\_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
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 its place. The conversion currently in progress will be terminated. The A sequence that was interrupted will automatically resume after the B sequence completes. The channel whose conversion was terminated will be re-sampled and the conversion sequence will resume from that point.	
30	MODE		Indicates whether the primary method for retrieving conversion results for this sequence will be accomplished via reading the global data register (SEQA_GDAT) at the end of each conversion, or the individual channel result registers at the end of the entire sequence. Impacts when conversion-complete interrupt 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 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 flag will be set when the entire set of sequence-A conversions completes. This flag will need to be explicitly cleared by software in this mode. The OVERRUN bit in the SEQA_GDAT register will NOT contribute to generation of an overrun interrupt trigger since it is assumed this register may not be utilized in this mode.	



Table 249. A/D Conversion Sequence A Control Register (SEQA\_CTRL, address 0x4001 C008) bit description

Bit	Symbol	Value	Description	Reset value
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.	

### 21.6.3 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 21.3.1 “Perform a single ADC conversion using a software trigger”](#).

**Remark:** Set the BURST and SEQB\_ENA bits at the same time.

Table 250. A/D Conversion Sequence B Control Register (SEQB\_CTRL, address 0x4001 C00C) bit description

Bit	Symbol	Value	Description	Reset value
11:0	CHANNELS		<p>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.</p> <p>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.</p> <p><b>Remark:</b> 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.</p>	0x00
14:12	TRIGGER		<p>Selects which of the available hardware trigger sources will cause this conversion sequence to be initiated. Program the trigger input number in this field.</p> <p><b>Remark:</b> 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.</p>	0x0
17:15	-		Reserved.	-



Table 250. A/D Conversion Sequence B Control Register (SEQB\_CTRL, address 0x4001 C00C) bit description

Bit	Symbol	Value	Description	Reset value
18	TRIGPOL		Select the polarity of the selected input trigger for this conversion sequence.  <b>Remark:</b> In order to avoid generating a spurious trigger, it is recommended writing to this field only when the SEQA_ENA bit (bit 31) is low. It is safe to change this field and set bit 31 in the same write.	0
		0	Negative edge. A negative edge launches the conversion sequence on the selected trigger input.	
		1	Positive edge. A positive edge launches the conversion sequence on the selected trigger input.	
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 an 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 a 1 to this bit if the BURST bit is set.  <b>Remark:</b> This bit is only set to a 1 momentarily when written to launch a conversion sequence. It will consequently always read-back as a zero.	0
27	BURST		Writing a 1 to this bit will cause this conversion sequence to be continuously cycled through. Other B triggers will be ignored while this bit is set.  Repeated conversions can be halted by clearing this bit. The sequence currently in progress will be completed before conversions are terminated.	0

Table 250. A/D Conversion Sequence B Control Register (SEQB\_CTRL, address 0x4001 C00C) bit description

Bit	Symbol	Value	Description	Reset value
28	SINGLESTEP		When this bit is set, a hardware trigger or a write to the START bit will launch a single conversion on the next channel in the sequence instead of the default response of launching an entire sequence of conversions. Once all of the channels comprising a sequence have been converted, a subsequent trigger will repeat the sequence beginning with the first enabled channel.  Interrupt generation will still occur either after each individual conversion or at the end of the entire sequence, depending on the state of the MODE bit.	0
29	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	N/A
30	MODE		Indicates whether the primary method for retrieving conversion results for this sequence will be accomplished via reading the global data register (SEQB_GDAT) at the end of each conversion, or the individual channel result registers at the end of the entire sequence.  Impacts when conversion-complete interrupt 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 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 flag will be set when the entire set of sequence B conversions completes. This flag will need to be explicitly cleared by software in this mode.  The OVERRUN bit in the SEQB_GDAT register will NOT contribute to generation of an overrun interrupt since it is assumed this register will not be utilized in this mode.	
31	SEQB_ENA		Sequence Enable. In order to avoid spuriously triggering the sequence, care should be taken to only set the SEQA_ENA bit when the selected trigger input is in its INACTIVE state (as defined by the TRIGPOL bit). If this condition is not met, the sequence will be triggered immediately upon being enabled.	0
		0	Disabled. Sequence B is disabled. Sequence B triggers are ignored. If this bit is cleared while sequence B is in progress, the sequence will be halted at the end of the current conversion. After the sequence is re-enabled, a new trigger will be required to restart the sequence beginning with the next enabled channel.	
		1	Enabled. Sequence B is enabled.	

### 21.6.4 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.

**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 251. 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	<p>This field contains the 12-bit A/D conversion result from the most recent conversion performed under conversion sequence associated with this register.</p> <p>The result is the a binary fraction representing the voltage on the currently-selected input channel as it falls within the range of <math>V_{REFP}</math> to <math>V_{REFN}</math>. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on <math>V_{REFN}</math>, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on <math>V_{REFP}</math>.</p> <p>DATAVALID = 1 indicates that this result has not yet been read.</p>	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
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	<p>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.</p> <p>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).</p>	0
31	DATAVALID	<p>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.</p> <p>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).</p>	0

Table 252. 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	<p>This field contains the 12-bit A/D conversion result from the most recent conversion performed under conversion sequence associated with this register.</p> <p>This will be a binary fraction representing the voltage on the currently-selected input channel as it falls within the range of <math>V_{REFP}</math> to <math>V_{REFN}</math>. Zero in the field indicates that the voltage on the input pin was less than, equal to, or close to that on <math>V_{REFN}</math>, while 0xFFF indicates that the voltage on the input was close to, equal to, or greater than that on <math>V_{REFP}</math>.</p> <p>DATAVALID = 1 indicates that this result has not yet been read.</p>	NA
17:16	THCMPRANGE	<p>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).</p> <p>Threshold Range Comparison result.</p> <p>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).</p> <p>0x1 = Below Range: The last completed conversion on was less than the value programmed into the designated LOW threshold register (THRn_LOW).</p> <p>0x2 = Above Range: The last completed conversion was greater than the value programmed into the designated HIGH threshold register (THRn_HIGH).</p> <p>0x3 = Reserved.</p>	
19:18	THCMPCROSS	<p>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.</p> <p>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.</p> <p>0x1 = Reserved.</p> <p>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.</p> <p>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.</p>	
25:20	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 252. A/D Sequence B Global Data Register (SEQB\_GDAT, address 0x4001 C014) bit description

Bit	Symbol	Description	Reset value
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	<p>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.</p> <p>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).</p>	0
31	DATAVALID	<p>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.</p> <p>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).</p>	0

### 21.6.5 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 253. 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 253. 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	<p>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.</p> <p>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.</p> <p>This bit (in any of the 12 registers) will cause an overrun interrupt request to be asserted if the overrun interrupt is enabled.</p> <p><b>Remark:</b> 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.</p>	NA
31	DATAVALID	<p>This bit is set to 1 when an A/D conversion on this channel completes.</p> <p>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.</p> <p><b>Remark:</b> 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.</p>	NA

### 21.6.6 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, then a threshold crossing has occurred. In this case the MSB of the THCMP\_CROSS status bits will indicate that a threshold crossing has occurred and the LSB will indicate the direction of the crossing. A threshold crossing event will also generate an interrupt request if enabled to do so via the ADCMPINTEN bits associated with each channel in the INTEN register.



**Table 254. 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 255. 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

### 21.6.7 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 256. Compare High Threshold register0 (THR0\_HIGH, address 0x4001 C058) bit description**

Bit	Symbol	Description	Reset value
3:0		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:4	THRHIGH	High threshold value against which A/D results will be compared	0x000
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA



Table 257. 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

### 21.6.8 A/D Channel Threshold Select register

For each channel, this register indicates which pair of threshold registers conversion results should be compared to.

Table 258. 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 258. 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

### 21.6.9 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.

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 259. 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 trigger is disabled.	
		1	Enabled. The sequence A interrupt 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 trigger is disabled.	
		1	Enabled. The sequence B interrupt 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.	
8:7	ADCMPINTEN2		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	

Table 259. A/D Interrupt Enable register (INTEN, address 0x4001 C064 ) bit description

Bit	Symbol	Value	Description	Reset value
		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	
24:23	ADCMPINTEN10		Threshold comparison interrupt enable.	00
		0x0	Disabled.	
		0x1	Outside threshold.	

Table 259. A/D Interrupt Enable register (INTEN, address 0x4001 C064 ) bit description

Bit	Symbol	Value	Description	Reset value
		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

### 21.6.10 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).

Table 260. 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
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

Table 260. A/D Flags register (FLAGS, address 0x4001 C068) bit description

Bit	Symbol	Description	Reset value
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 flag.  If the MODE bit in the SEQA_CTRL register is 0, this flag will mirror the DATAVALID bit in the sequence A global data register (SEQA_GDAT), which is set at the end of every A/D conversion performed as part of sequence A. It will be cleared automatically when the SEQA_GDAT register is read.  If the MODE bit in the SEQA_CTRL register is 1, this flag will be set upon completion of an entire A sequence. In this case it must be cleared by writing a 1 to this SEQA_INT bit.  This interrupt must be enabled in the INTEN register.	0

Table 260. A/D Flags register (FLAGS, address 0x4001 C068) bit description

Bit	Symbol	Description	Reset value
29	SEQB_INT	Sequence A interrupt 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 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	Overflow 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

## 21.7 Functional description

### 21.7.1 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.

### 21.7.2 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.

#### 21.7.2.1 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.

#### 21.7.3 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.

### 21.7.4 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.

#### 21.7.4.1 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.

#### 21.7.4.2 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.

#### 21.7.4.3 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.

### 21.7.5 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.

### 21.7.6 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.

### 22.1 How to read this chapter

The analog comparator is available on all LPC802 parts.

### 22.2 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

### 22.3 Basic configuration

Configure the analog comparator using the following registers:

- In the SYSAHBCLKCTRL register, set bit 19 ([Table 63](#)) to enable the clock to the register interface.
- You can enable or disable the power to the analog comparator through the PDRUNCFG register ([Table 82](#)).
- Clear the analog comparator peripheral reset using the PRESETCTRL register ([Table 64](#)).
- The analog comparator interrupt is connected to interrupt #11 in the NVIC.
- Configure the analog comparator pin functions through the switch matrix. See [Section 22.4](#).

#### 22.3.1 Connect the comparator output to the ADC

The comparator output function (ACMP\_O) can be used to start the ADC conversion, more generally, to create an ADC conversion event without assigning a pin through the switch matrix. To create an ADC event internally connected to the comparator output, select the comparator output as one of the ADC trigger inputs through the ADC trigger select register (see [Table 249 “A/D Conversion Sequence A Control Register \(SEQA\\_CTRL, address 0x4001 C008\) bit description”](#) and [Table 250 “A/D Conversion Sequence B Control Register \(SEQB\\_CTRL, address 0x4001 C00C\) bit description”](#)).

## 22.4 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 8.3.1 “Connect an internal signal to a package pin”](#) to assign the analog comparator output to any pin on the LPC802 package.

**Table 261. Analog comparator pin description**

Function	Type	Pin	Description	SWM register	Reference
ACMP_I1	I	PIO0_0	Comparator input 1	PINENABLE0	<a href="#">Table 97</a>
ACMP_I2	I	PIO0_1	Comparator input 2	PINENABLE0	<a href="#">Table 97</a>
ACMP_I3	I	PIO0_14	Comparator input 3	PINENABLE0	<a href="#">Table 97</a>
ACMP_I4	I	PIO0_16	Comparator input 4	PINENABLE0	<a href="#">Table 97</a>
ACMP_O	O	any	Comparator output	PINASSIGN5	<a href="#">Table 94</a>
VDDCMP	I	PIO0_7	External reference voltage source for 32-stage Voltage Ladder.	PINENABLE0	<a href="#">Table 97</a>

## 22.5 General description

The analog comparator can compare voltage levels on external pins and internal voltages.

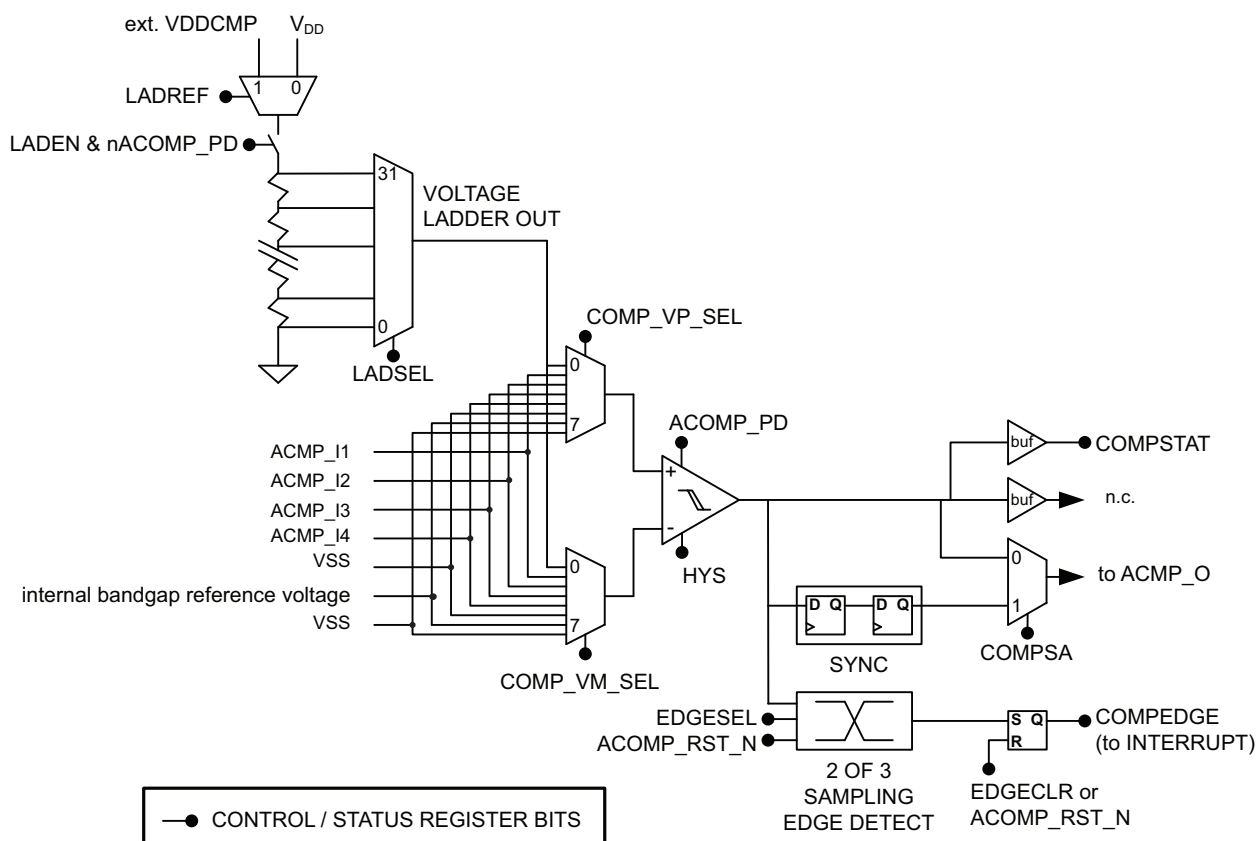
The comparator has seven inputs multiplexed separately to its positive and negative inputs. The multiplexers are controlled by the comparator register CTL (see [Figure 47](#) and [Table 263](#)).

Input 0 of the multiplexer is the programmable voltage ladder output.

Inputs 1 to 4 connect the external inputs ACMP\_I[4:1].

Input 6 of the multiplexer connects the internal reference voltage input.

Input 7 of the multiplexer connects the GND.



aaa-028402

**Fig 47. Comparator block diagram**

### 22.5.1 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.

### 22.5.2 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.

### 22.5.3 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.

### 22.5.4 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 63 “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.

## 22.6 Register description

**Table 262. Register overview: Analog comparator (base address 0x4002 4000)**

Name	Access	Address offset	Description	Reset value	Reference
CTRL	R/W	0x000	Comparator control register	0	<a href="#">Table 263</a>
LAD	R/W	0x004	Voltage ladder register	0	<a href="#">Table 264</a>

### 22.6.1 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 263](#) are reserved and should be written as 0.

**Table 263. 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 263. 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	GND	
		0x6	Band gap. Internal reference voltage.	
		0x7	GND	
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	GND	
		0x6	Band gap. Internal reference voltage.	
		0x7	GND	
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.	0
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	-

## 22.6.2 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 264. 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 <sub>SS</sub> 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



### 23.1 How to read this chapter

---

The CRC engine is available on all LPC802 parts.

### 23.2 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)

### 23.3 Basic configuration

---

Enable the clock to the CRC engine in the SYSAHBCLKCTRL register ([Table 63](#), bit 13).

### 23.4 Pin description

---

The CRC engine has no configurable pins.

### 23.5 General description

---

The Cyclic Redundancy Check (CRC) generator with programmable polynomial settings supports several CRC standards commonly used.

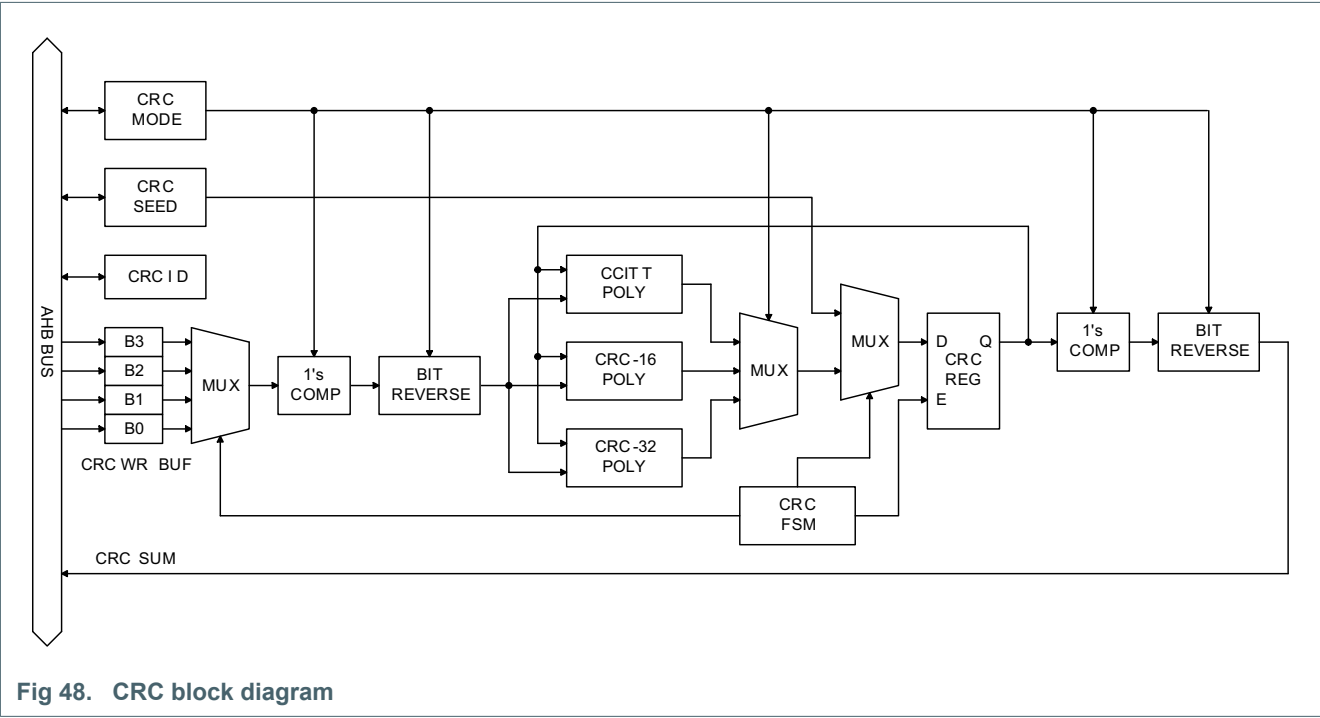


Fig 48. CRC block diagram

## 23.6 Register description

Table 265. 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	<a href="#">Table 266</a>
SEED	R/W	0x004	CRC seed register	0x0000 FFFF	<a href="#">Table 267</a>
SUM	RO	0x008	CRC checksum register	0x0000 FFFF	<a href="#">Table 268</a>
WR_DATA	WO	0x008	CRC data register	-	<a href="#">Table 269</a>

### 23.6.1 CRC mode register

Table 266. 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

### 23.6.2 CRC seed register

Table 267. 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.  <b>Remark:</b> A write access to this register will overrule the CRC calculation in progress.	0x0000 FFFF

### 23.6.3 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 268. 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

### 23.6.4 CRC data register

This register is a Write-only register containing the data block for which the CRC sum will be calculated.

Table 269. 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.	-

## 23.7 Functional description

The following sections describe the register settings for each supported CRC standard:

### 23.7.1 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

### 23.7.2 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

### 23.7.3 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

### 24.1 How to read this chapter

The ROM-based 32-bit integer division routines are available on all parts.

### 24.2 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.

### 24.3 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 49](#) shows the pointer structure used to call the Integer divider API.

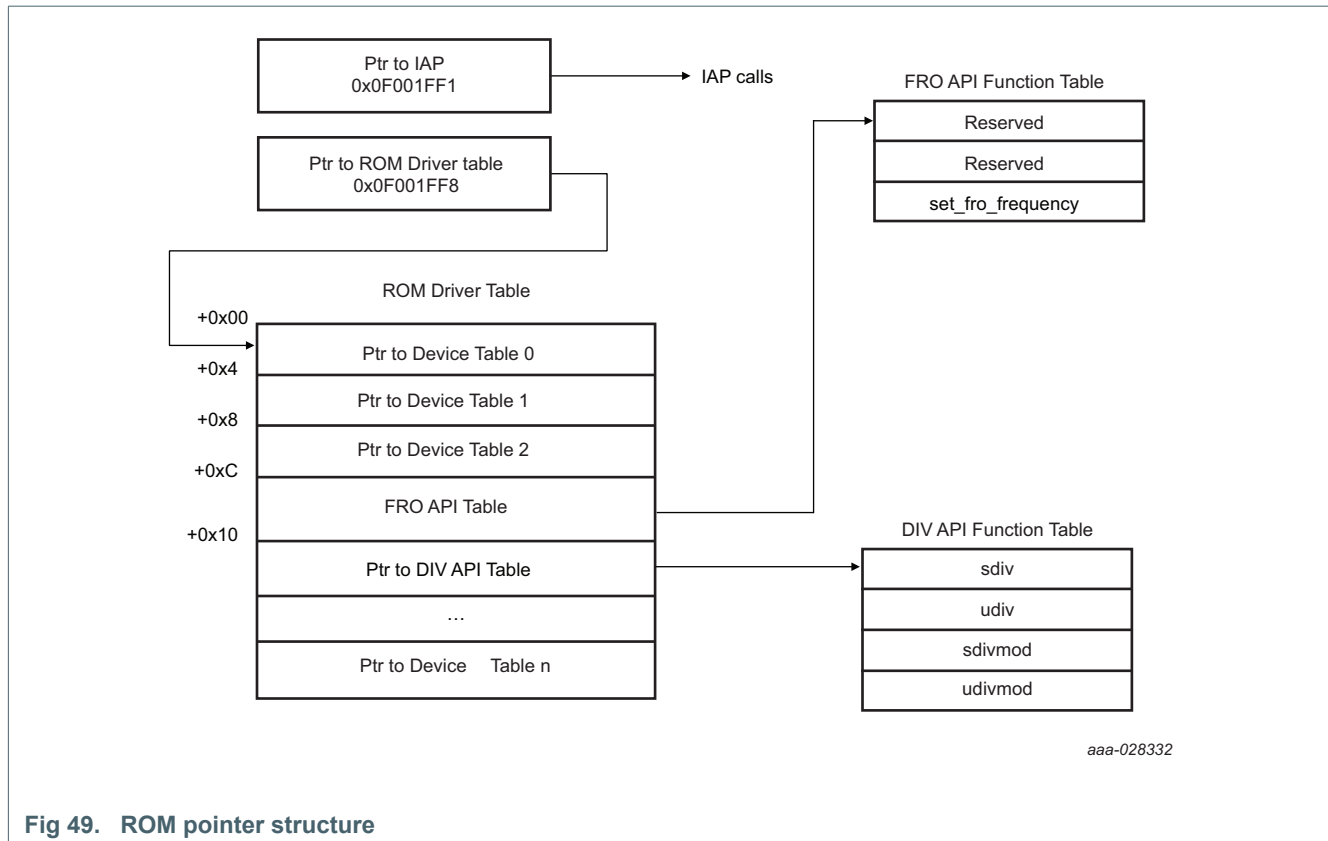


Fig 49. ROM pointer structure

## 24.4 API description

The integer division routines perform arithmetic integer division operations and can be called in the application code through simple API calls.

**Table 270. Divide API calls**

API call	Description	Reference
<code>int(*sdiv)(int numerator, int denominator);</code>	Signed integer division	<a href="#">Table 271</a>
<code>unsigned(*udiv) (int numerator, int denominator);</code>	Unsigned integer division	<a href="#">Table 272</a>
<code>sdiv_t (*sdivmod)(int numerator, int denominator);</code>	Signed integer division with remainder	<a href="#">Table 273</a>
<code>udiv_t (*udivmod)(unsigned numerator, unsigned denominator);</code>	Unsigned integer division with remainder	<a href="#">Table 274</a>

The following function prototypes are used:

```
typedef struct {
    int quot;        /*!< Quotient */
    int rem;         /*!< Remainder */
} IDIV_RETURN_T;

typedef struct {
    unsigned quot; /*!< Quotient */
    unsigned rem; /*!< Remainder */
} UIDIV_RETURN_T;

typedef struct {
    int (*sdiv)(int numerator, int denominator); /*!< Signed integer division */
    unsigned (*udiv)(unsigned numerator, unsigned denominator); /*!< Unsigned
integer division */
    IDIV_RETURN_T (*sdivmod)(int numerator, int denominator); /*!< Signed integer
division with remainder */
    UIDIV_RETURN_T (*udivmod)(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 3.4.2 “ROM-based APIs”](#) must be included in the code.

### 24.4.1 DIV signed integer division

**Table 271. sdiv**

Routine	sdiv
Prototype	<code>int(*sdiv)(int32_t numerator, int32_t denominator);</code>
Input parameter	numerator: Numerator signed integer. denominator: Denominator signed integer.
Return	Signed division result without remainder.
Description	Signed integer division

## 24.4.2 DIV unsigned integer division

Table 272. uidiv

Routine	uidiv
Prototype	<code>int(*uidiv)(int32_t numerator, int32_t denominator);</code>
Input parameter	numerator: Numerator signed integer. denominator: Denominator signed integer.
Return	Unsigned division result without remainder.
Description	Unsigned integer division

## 24.4.3 DIV signed integer division with remainder

Table 273. sidivmod

Routine	sidivmod
Prototype	<code>IDIV_RETURN_T (*sidivmod) (int32_t numerator, int32_t denominator);</code>
Input parameter	numerator: Numerator signed integer. denominator: Denominator signed integer.
Return	Signed division result remainder.
Description	Signed integer division with remainder

## 24.4.4 DIV unsigned integer division with remainder

Table 274. uidivmod

Routine	uidivmod
Prototype	<code>UIDIV_RETURN_T(*uidiv)(uint32_t numerator, uint32_t denominator);</code>
Input parameter	numerator: Numerator unsigned integer. denominator: Denominator unsigned integer.
Return	Unsigned division result with remainder.
Description	Unsigned integer division



## 24.5 Functional description

---

### 24.5.1 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) */
```

### 24.5.2 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) */
```

### 25.1 How to read this chapter

The debug functionality is identical for all LPC802 parts.

### 25.2 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.

### 25.3 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 is available.

### 25.4 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 8.3.2](#) to enable the analog comparator inputs and the reference voltage input.

**Table 275. SWD pin description**

Function	Type	Pin	Description	SWM register	Reference
SWCLK	I/O	SWCLK/PIO0_3/ TCLK	<b>Serial Wire Clock.</b> This pin is the clock for SWD debug logic when in the Serial Wire Debug mode (SWD). This pin is pulled up internally.	PINENABLE0	<a href="#">Table 97</a>
SWDIO	I/O	SWDIO/PIO0_2/ TMS	<b>Serial wire debug data input/output.</b> The SWDIO pin is used by an external debug tool to communicate with and control the LPC802. This pin is pulled up internally.	PINENABLE0	<a href="#">Table 97</a>

The boundary scan mode and the pins needed are selected by hardware (see [Section 25.5.3](#)). There is no access to the boundary scan pins through the switch matrix.

Table 276. JTAG boundary scan pin description

Function	Pin name	Type	Description
TCK	SWCLK/PIO0_3/ TCK	I	<b>JTAG Test Clock.</b> This pin is the clock for JTAG boundary scan when the $\overline{\text{RESET}}$ pin is LOW.
TMS	SWDIO/PIO0_2/ TMS	I	<b>JTAG Test Mode Select.</b> 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	<b>JTAG Test Data In.</b> 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	<b>JTAG Test Data Output.</b> 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/ $\overline{\text{WAKEUP/TRST}}$	I	<b>JTAG Test Reset.</b> 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.

## 25.5 Functional description

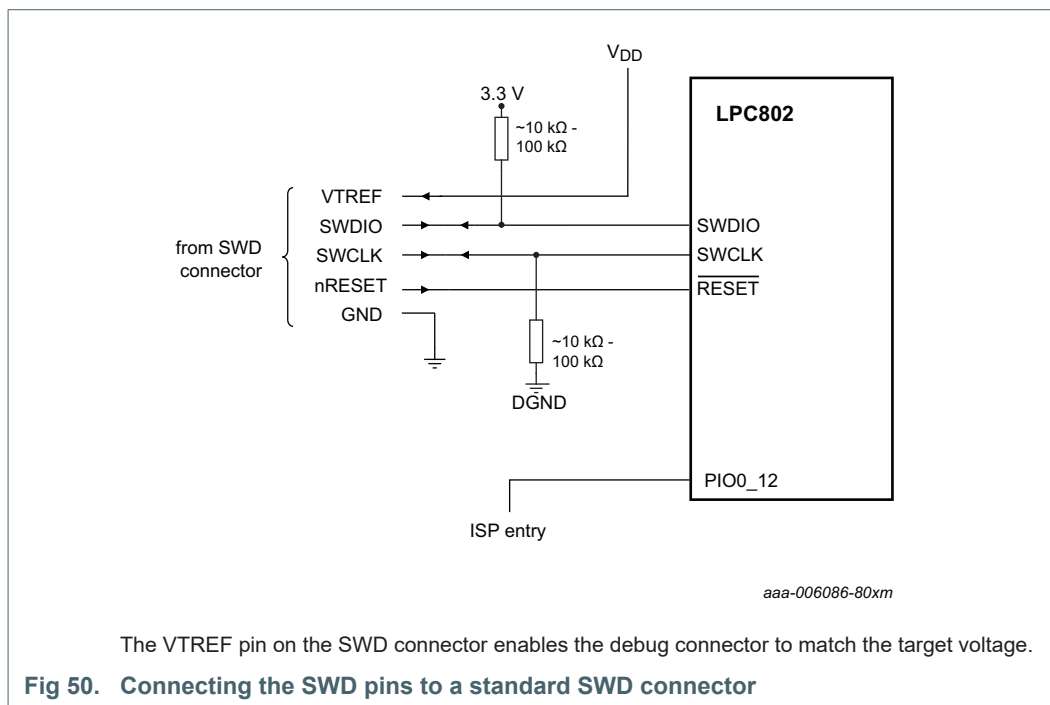
### 25.5.1 Debug limitations

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.

### 25.5.2 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 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.



### 25.5.3 Boundary scan

The  $\overline{\text{RESET}}$  pin selects between the JTAG boundary scan ( $\overline{\text{RESET}} = \text{LOW}$ ) and the Arm SWD debug ( $\overline{\text{RESET}} = \text{HIGH}$ ). 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 1 ms.
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  $\overline{\text{RESET}} = \text{HIGH}$  places the test TAP in Run-Test Idle mode.

## 26.1 Abbreviations

Table 277. 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

## 26.2 References

- [1] **LPC802** — [LPC802 Data sheet](#)
- [2] **ES\_LPC802** — [LPC802 Errata sheet](#)
- [3] **DDI0484B\_cortex\_m0p\_r0p0\_trm** — Arm Cortex-M0+ Technical Reference Manual
- [4] **DDI0486A** — Arm technical reference manual
- [5] **ARMv6-M Architecture Reference Manual**

## 26.4 Tables

Table 1.	Ordering information	7	Table 45.	Interrupt Priority Register 1 (IPR1, address 0xE000 E404) bit description	45
Table 2.	Ordering options	7	Table 46.	Interrupt Priority Register 2 (IPR2, address 0xE000 E408) bit description	45
Table 3.	Pin location in ISP mode	12	Table 47.	Interrupt Priority Register 3 (IPR3, address 0xE000 E40C) bit description	45
Table 4.	API calls	14	Table 48.	Interrupt Priority Register 4 (IPR4, address 0xE000 E410) bit description	46
Table 5.	LPC802 flash configuration	17	Table 49.	Interrupt Priority Register 5 (IPR5, address 0xE000 E414) bit description	46
Table 6.	USART ISP command limitations in CRP modes	18	Table 50.	Interrupt Priority Register 6 (IPR6, address 0xE000 E418) bit description	46
Table 7.	USART ISP command summary	21	Table 51.	Interrupt Priority Register 7 (IPR7, address 0xE000 E41C) bit description	47
Table 8.	ISP commands allowed for different CRP levels	21	Table 52.	SYSCON pin description	49
Table 9.	USART ISP Unlock command	22	Table 53.	Clocking diagram signal name descriptions	50
Table 10.	USART ISP Set Baud Rate command	22	Table 54.	Register overview: System configuration (base address 0x4004 8000)	53
Table 11.	USART ISP Echo command	23	Table 55.	System memory remap register (SYSMEMREMAP, address 0x4004 8000) bit description	56
Table 12.	USART ISP Write to RAM command	23	Table 56.	System reset status register (SYSRSTSTAT, address 0x4004 8038) bit description	56
Table 13.	USART ISP Read Memory command	23	Table 57.	Main clock source select register (MAINCLKSEL, address 0x4004 8050) bit description	57
Table 14.	USART ISP Prepare sectors for write operation command	24	Table 58.	Main clock source update enable register (MAINCLKUEN, address 0x4004 8054) bit description	57
Table 15.	USART ISP Copy command	25	Table 59.	System clock divider register (SYSAHBCLKDIV, address 0x4004 8058) bit description	57
Table 16.	USART ISP Go command	25	Table 60.	ADC clock source select register (ADCCLKSEL, address 0x4004 8064) bit description	58
Table 17.	USART ISP Erase sector command	26	Table 61.	ADC clock divider register (ADCCCLKDIV, address 0x4004 8068) bit description	58
Table 18.	USART ISP Erase page command	26	Table 62.	WDT and Wake Timer clock enable control register (LPOSCCLKEN, address 0x4004 807C) bit description	58
Table 19.	USART ISP Blank check sector command	27	Table 63.	System clock control 0 register (SYSAHBCLKCTRL0, address 0x4004 8080) bit description	59
Table 20.	USART ISP Read Part Identification command	27	Table 64.	Peripheral reset control 0 register (PRESETCTRL0, address 0x4004 8088) bit description	61
Table 21.	LPC802 device identification numbers	27	Table 65.	Peripheral reset control 1 register (PRESETCTRL1, address 0x4004 808C) bit description	62
Table 22.	USART ISP Read Boot Code version number command	27	Table 66.	Peripheral clock source select registers	63
Table 23.	USART ISP Compare command	28	Table 67.	Fractional generator 0 divider value register (FRG0DIV, address 0x4004 80D0) bit description	64
Table 24.	USART ReadUID command	28	Table 68.	Fractional generator 0 multiplier value register (FRG0MULT, address 0x4004 80D4) bit description	64
Table 25.	USART ISP Read CRC checksum command	29	Table 69.	FRG0 clock source select register (FRG0CLKSEL, address 0x4004 80D8) bit description	65
Table 26.	ISP/IAP Error codes	30			
Table 27.	IAP Command Summary	32			
Table 28.	IAP Prepare sector(s) for write operation command	33			
Table 29.	IAP Copy RAM to flash command	33			
Table 30.	IAP Erase Sector(s) command	34			
Table 31.	IAP Blank check sector(s) command	34			
Table 32.	IAP Read Part Identification command	34			
Table 33.	IAP Read Boot Code version number command	35			
Table 34.	IAP Compare command	35			
Table 35.	Reinvoke ISP	35			
Table 36.	IAP ReadUID command	36			
Table 37.	IAP Erase page command	36			
Table 38.	Connection of interrupt sources to the NVIC	37			
Table 39.	Register overview: NVIC (base address 0xE000 E000)	40			
Table 40.	Interrupt Set Enable Register 0 register (ISER0, address 0xE000 E100) bit description	41			
Table 41.	Interrupt clear enable register 0 (ICER0, address 0xE000 E180)	42			
Table 42.	Interrupt set pending register 0 register (ISPR0, address 0xE000 E200) bit description	43			
Table 43.	Interrupt clear pending register 0 register (ICPR0, address 0xE000 E280) bit description	44			
Table 44.	Interrupt Priority Register 0 (IPR0, address 0xE000 E400) bit description	44			

Table 70. CLKOUT clock source select register (CLKOUTSEL, address 0x4004 80F0) bit description . . . . .	65	0x4000 C01C) bit description . . . . .	85
Table 71. CLKOUT clock divider registers (CLKOUTDIV, address 0x4004 80F4) bit description . . . . .	65	Table 97. Pin enable register 0 (PINENABLE0, address 0x4000 C1C0) bit description . . . . .	86
Table 72. POR captured PIO status register 0 (PIOPORCAP0, address 0x4004 8100) bit description . . . . .	66	Table 98. Pinout summary . . . . .	88
Table 73. BOD control register (BODCTRL, address 0x4004 8150) bit description . . . . .	66	Table 99. Register overview: I/O configuration (base address 0x4004 4000) . . . . .	90
Table 74. System tick timer calibration register (SYSTCKCAL, address 0x4004 8154) bit description . . . . .	66	Table 100. I/O configuration registers ordered by pin name . . . . .	91
Table 75. IRQ latency register (IRQLATENCY, address 0x4004 8170) bit description . . . . .	67	Table 101. PIO0_17 register (PIO0_17, address 0x4004 4000) bit description . . . . .	91
Table 76. NMI source selection register (NMISRC, address 0x4004 8174) bit description . . . . .	67	Table 102. PIO0_13 register (PIO0_13, address 0x4004 4004) bit description . . . . .	92
Table 77. Pin interrupt select registers (PINTSEL[0:7], address 0x4004 8178 (PINTSEL0) to 0x4004 8194 (PINTSEL7)) bit description . . . . .	68	Table 103. PIO0_12 register (PIO0_12, address 0x4004 4008) bit description . . . . .	93
Table 78. Start logic 0 pin wake-up enable register 0 (STARTERP0, address 0x4004 8204) bit description . . . . .	68	Table 104. PIO0_5 register (PIO0_5, address 0x4004 400C) bit description . . . . .	93
Table 79. Start logic 1 interrupt wake-up enable register (STARTERP1, address 0x4004 8214) bit description . . . . .	69	Table 105. PIO0_4 register (PIO0_4, address 0x4004 4010) bit description . . . . .	94
Table 80. Deep-sleep configuration register (PDSLEEPCFG, address 0x4004 8230) bit description . . . . .	70	Table 106. PIO0_3 register (PIO0_3, address 0x4004 4014) bit description . . . . .	94
Table 81. Wake-up configuration register (PDWAKECFG, address 0x4004 8234) bit description . . . . .	71	Table 107. PIO0_2 register (PIO0_2, address 0x4004 4018) bit description . . . . .	95
Table 82. Power configuration register (PDRUNCFG, address 0x4004 8238) bit description . . . . .	72	Table 108. PIO0_11 register (PIO0_11, address 0x4004 401C) bit description . . . . .	96
Table 83. Device ID register (DEVICE_ID, address 0x4004 83F8) bit description . . . . .	72	Table 109. PIO0_10 register (PIO0_10, address 0x4004 4020) bit description . . . . .	96
Table 84. LPC802 device identification numbers . . . . .	72	Table 110. PIO0_16 register (PIO0_16, address 0x4004 4024) bit description . . . . .	97
Table 85. FRO API call . . . . .	75	Table 111. PIO0_15 register (PIO0_15, address 0x4004 4028) bit description . . . . .	97
Table 86. set_fro_frequency routine . . . . .	75	Table 112. PIO0_1 register (PIO0_1, address 0x4004 402C) bit description . . . . .	98
Table 87. Movable functions (assign to pins PIO0_0 to PIO0_5 and PIO0_7 to PIO0_17 through switch matrix) . . . . .	80	Table 113. PIO0_9 register (PIO0_9, address 0x4004 4034) bit description . . . . .	98
Table 88. Register overview: Switch matrix (base address 0x4000 C000) . . . . .	82	Table 114. PIO0_8 register (PIO0_8, address 0x4004 4038) bit description . . . . .	99
Table 89. Pin assign register 0 (PINASSIGN0, address 0x4000 C000) bit description . . . . .	83	Table 115. PIO0_7 register (PIO0_7, address 0x4004 403C) bit description . . . . .	99
Table 90. Pin assign register 1 (PINASSIGN1, address 0x4000 C004) bit description . . . . .	83	Table 116. PIO0_0 register (PIO0_0, address 0x4004 4044) bit description . . . . .	100
Table 91. Pin assign register 2 (PINASSIGN2, address 0x4000 C008) bit description . . . . .	83	Table 117. PIO0_14 register (PIO0_14, address 0x4004 4048) bit description . . . . .	101
Table 92. Pin assign register 3 (PINASSIGN3, address 0x4000 C00C) bit description . . . . .	84	Table 118. GPIO pins available . . . . .	102
Table 93. Pin assign register 4 (PINASSIGN4, address 0x4000 C010) bit description . . . . .	84	Table 119. Register overview: GPIO port (base address 0xA000 0000) . . . . .	103
Table 94. Pin assign register 5 (PINASSIGN5, address 0x4000 C014) bit description . . . . .	85	Table 120. GPIO port byte pin registers (B[0:17], addresses 0xA000 0000 (B0) to 0xA000 0011 (B17)) bit description . . . . .	103
Table 95. Pin assign register 6 (PINASSIGN6, address 0x4000 C018) bit description . . . . .	85	Table 121. GPIO port word pin registers (W[0:17], addresses 0xA000 1000 (W0) to 0xA000 1044 (W17)) bit description . . . . .	104
Table 96. Pin assign register 7 (PINASSIGN7, address 0x4000 C01C) bit description . . . . .	85	Table 122. GPIO direction port register (DIR0), address 0xA000 2000 bit description . . . . .	104
		Table 123. GPIO mask port register (MASK0), address 0xA000 2080 bit description . . . . .	104
		Table 124. GPIO port pin register (PIN0), address 0xA000 2100 bit description . . . . .	105
		Table 125. GPIO masked port pin register (MPIN0), address	



0xA000 2180 bit description . . . . .	105	Table 151. Power control register (PCON, address 0x4002 0000) bit description . . . . .	138
Table 126. GPIO port set register (SET0), address 0xA000 2200 bit description. . . . .	105	Table 152. General purpose registers 0 to 4 (GPREG[0:4], address 0x4002 0004 (GPREG0) to 0x4002 0014 (GPREG4)) bit description . . . . .	138
Table 127. GPIO port clear register (CLR0), address 0xA000 2280 bit description. . . . .	106	Table 153. WUSRCREG, address 0x4002 0020 bit description . . . . .	139
Table 128. GPIO port toggle register (NOT0), address 0xA000 2300 bit description . . . . .	106	Table 154. WUENAREG, address 0x4002 0024 bit description . . . . .	139
Table 129. GPIO port direction set register (DIRSET0), address 0xA000 2380 bit description . . . . .	106	Table 155. Peripheral configuration in reduced power modes 140	
Table 130. GPIO port direction clear register (DIRCLR0), 0xA000 2400 bit description . . . . .	106	Table 156. USART pin description . . . . .	149
Table 131. GPIO port direction toggle register (DIRNOT0), address 0xA000 2480 bit description . . . . .	107	Table 157. Register overview: USART (base address 0x4006 4000 (USART0) and 0x4006 8000 (USART1)) . . . . .	151
Table 132. Pin interrupt/pattern match engine pin description 110		Table 158. USART Configuration register (CFG, address 0x4006 4000 (USART0), 0x4006 8000 (USART1)) bit description . . . . .	152
Table 133. Register overview: Pin interrupts and pattern match engine (base address: 0xA000 4000) . 115		Table 159. USART Control register (CTL, address 0x4006 4004 (USART0), 0x4006 8004 (USART1)) bit description . . . . .	154
Table 134. Pin interrupt mode register (ISEL, address 0xA000 4000) bit description . . . . .	115	Table 160. USART Status register (STAT, address 0x4006 4008 (USART0), 0x4006 8008 (USART1)) bit description . . . . .	156
Table 135. Pin interrupt level or rising edge interrupt enable register (IENR, address 0xA000 4004) bit description . . . . .	116	Table 161. USART Interrupt Enable read and set register (INTENSET, address 0x4006 400C (USART0), 0x4006 800C (USART1)) bit description . . . 157	
Table 136. Pin interrupt level or rising edge interrupt set register (SIENR, address 0xA000 4008) bit description . . . . .	116	Table 162. USART Interrupt Enable clear register (INTENCLR, address 0x4006 4010 (USART0), 0x4006 8010 (USART1)) bit description . . . . 158	
Table 137. Pin interrupt level or rising edge interrupt clear register (CIENR, address 0xA000 400C) bit description . . . . .	117	Table 163. USART Receiver Data register (RXDAT, address 0x4006 4014 (USART0), 0x4006 8014 (USART1)) bit description . . . . .	158
Table 138. Pin interrupt active level or falling edge interrupt enable register (IENF, address 0xA000 4010) bit description . . . . .	117	Table 164. USART Receiver Data with Status register (RXDATSTAT, address 0x4006 4018 (USART0), 0x4006 8018 (USART1)) bit description. . . . 159	
Table 139. Pin interrupt active level or falling edge interrupt set register (SIENF, address 0xA000 4014) bit description . . . . .	118	Table 165. USART Transmitter Data Register (TXDAT, address 0x4006 401C (USART0), 0x4006 801C (USART1)) bit description . . . . .	159
Table 140. Pin interrupt active level or falling edge interrupt clear register (CIENF, address 0xA000 4018) bit description . . . . .	118	Table 166. USART Baud Rate Generator register (BRG, address 0x4006 4020 (USART0), 0x4006 8020 (USART1)) bit description . . . . .	160
Table 141. Pin interrupt rising edge register (RISE, address 0xA000 401C) bit description . . . . .	118	Table 167. USART Interrupt Status register (INTSTAT, address 0x4006 4024 (USART0), 0x4006 8024 (USART1)) bit description . . . . .	160
Table 142. Pin interrupt falling edge register (FALL, address 0xA000 4020) bit description . . . . .	119	Table 168. USART Oversample selection register (OSR, address 0x4006 4028 (USART0), 0x4006 4028 (USART1)) bit description . . . . .	161
Table 143. Pin interrupt status register (IST, address 0xA000 4024) bit description . . . . .	119	Table 169. USART Address register (ADDR, address 0x4006 402C (USART0), 0x4006 802C (USART1)) bit description . . . . .	161
Table 144. Pattern match interrupt control register (PMCTRL, address 0xA000 4028) bit description . . . . .	120	Table 170. SPI Pin Description . . . . .	167
Table 145. Pattern match bit-slice source register (PMSRC, address 0xA000 402C) bit description . . . . .	120	Table 171. Register overview: SPI (base address 0x4005 8000 (SPI) . . . . .	168
Table 146. Pattern match bit slice configuration register (PMCFG, address 0xA000 4030) bit description 125		Table 172. SPI Configuration register (CFG, addresses 0x4005 8000 (SPI) bit description . . . . .	169
Table 147. Pin interrupt registers for edge- and level-sensitive pins . . . . .	130	Table 173. SPI Delay register (DLY, addresses 0x4005 8004 (SPI) bit description . . . . .	171
Table 148. System control register (SCR, address 0xE000 ED10) bit description . . . . .	135		
Table 149. Wake-up sources for reduced power modes . 137			
Table 150. Register overview: PMU (base address 0x4002 0000) . . . . .	137		



Table 174. SPI Status register (STAT, addresses 0x4005 8008 (SPI) bit description . . . . .	172	Table 203. Settings for 400 KHz clock rate . . . . .	208
Table 175. SPI Interrupt Enable read and Set register (INTENSET, addresses 0x4005 800C (SPI) bit description . . . . .	173	Table 204. Timer/Counter pin description . . . . .	213
Table 176. SPI Interrupt Enable clear register (INTENCLR, addresses 0x4005 8010 (SPI) bit description . . . . .	174	Table 205. Register overview: CTIMER (register base addresses 0x4003 8000) . . . . .	214
Table 177. SPI Receiver Data register (RXDAT, addresses 0x4005 8014 (SPI) bit description . . . . .	175	Table 206. Interrupt Register (IR, offset 0x000) bit description . . . . .	216
Table 178. SPI Transmitter Data and Control register (TXDATCTL, addresses 0x4005 8018 (SPI) bit description . . . . .	176	Table 207. Timer Control Register (TCR, offset 0x004) bit description . . . . .	216
Table 179. SPI Transmitter Data Register (TXDAT, addresses 0x4005 801C (SPI) bit description . . . . .	177	Table 208. Timer counter registers (TC, offset 0x08) bit description . . . . .	216
Table 180. SPI Transmitter Control register (TXCTL, addresses 0x4005 8020 (SPI) bit description . . . . .	178	Table 209. Timer prescale registers (PR, offset 0x00C) bit description . . . . .	217
Table 181. SPI Divider register (DIV, addresses 0x4005 8024 (SPI) bit description . . . . .	178	Table 210. Timer prescale counter registers (PC, offset 0x010) bit description . . . . .	217
Table 182. SPI Interrupt Status register (INTSTAT, addresses 0x4005 8028 (SPI) bit description . . . . .	178	Table 211. Match Control Register (MCR, offset 0x014) bit description . . . . .	217
Table 183. SPI mode summary . . . . .	179	Table 212. Timer match registers (MR[0:3], offset [0x018:0x024]) bit description . . . . .	218
Table 184. I2C-bus pin description . . . . .	191	Table 213. Capture Control Register (CCR, offset 0x028) bit description . . . . .	219
Table 185. Register overview: I2C (base address 0x4005 0000 (I2C0) . . . . .	193	Table 214. Timer capture registers (CR[0:2], offsets [0x02C:0x034]) bit description . . . . .	219
Table 186. I2C Configuration register (CFG, address 0x4005 0000 (I2C0) bit description . . . . .	193	Table 215. Timer external match registers (EMR, offset 0x03C) bit description . . . . .	220
Table 187. I2C Status register (STAT, address 0x4005 0004 (I2C0) bit description . . . . .	195	Table 216. Count Control Register (CTCR, offset 0x070) bit description . . . . .	221
Table 188. Master function state codes (MSTSTATE) . . . . .	198	Table 217. PWM Control Register (PWMC, offset 0x074) bit description . . . . .	222
Table 189. Slave function state codes (SLVSTATE) . . . . .	198	Table 218. Timer match shadow registers (MSR[0:3], offset [0x078:0x084]) bit description . . . . .	223
Table 190. Interrupt Enable Set and read register (INTENSET, address 0x4005 0008 (I2C0) bit description . . . . .	199	Table 219. Register overview: Watchdog timer (base address 0x4000 0000) . . . . .	230
Table 191. Interrupt Enable Clear register (INTENCLR, address 0x4005 000C (I2C0) bit description . . . . .	200	Table 220. Watchdog mode register (MOD, 0x4000 0000) bit description . . . . .	230
Table 192. Time-out value register (TIMEOUT, address 0x4005 0010 (I2C0) bit description . . . . .	201	Table 221. Watchdog operating modes selection . . . . .	232
Table 193. I2C Clock Divider register (CLKDIV, address 0x4005 0014 (I2C0) bit description . . . . .	201	Table 222. Watchdog Timer Constant register (TC, 0x4000 0004) bit description . . . . .	232
Table 194. I2C Interrupt Status register (INTSTAT, address 0x4005 0018 (I2C0) bit description . . . . .	202	Table 223. Watchdog Feed register (FEED, 0x4000 0008) bit description . . . . .	233
Table 195. Master Control register (MSTCTL, address 0x4005 0020 (I2C0) bit description . . . . .	203	Table 224. Watchdog Timer Value register (TV, 0x4000 000C) bit description . . . . .	233
Table 196. Master Time register (MSTTIME, address 0x4005 0024 (I2C0) bit description . . . . .	203	Table 225. Watchdog Timer Warning Interrupt register (WARNINT, 0x4000 0014) bit description . . . . .	233
Table 197. Master Data register (MSTDAT, address 0x4005 0028 (I2C0) bit description . . . . .	204	Table 226. Watchdog Timer Window register (WINDOW, 0x4000 0018) bit description . . . . .	234
Table 198. Slave Control register (SLVCTL, address 0x4005 0040 (I2C0) bit description . . . . .	204	Table 227. Register overview: WKT (base address 0x4000 8000) . . . . .	237
Table 199. Slave Data register (SLVDAT, address 0x4005 0044 (I2C0) bit description . . . . .	205	Table 228. Control register (CTRL, address 0x4000 8000) bit description . . . . .	237
Table 200. Slave Address registers (SLVADR[0:3], address 0x4005 0048 (SLVADR0) to 0x4005 0054 (SLVADR3) (I2C0) bit description . . . . .	205	Table 229. Counter register (COUNT, address 0x4000 800C) bit description . . . . .	238
Table 201. Slave address Qualifier 0 register (SLVQUAL0, address 0x4005 0058 (I2C0) bit description . . . . .	206	Table 230. Register overview: MRT (base address 0x4000 4000) . . . . .	242
Table 202. Monitor data register (MONRXDAT, address 0x4005 0080 (I2C0) bit description . . . . .	206	Table 231. Time interval register (INTVAL[0:3], address 0x4000 4000 (INTVAL0) to 0x4000 4030 (INTVAL3)) bit description . . . . .	242
		Table 232. Timer register (TIMER[0:3], address 0x4000 4004 (TIMER0) to 0x4000 4034 (TIMER3)) bit	

description .....	243	(CHAN_THRSEL, addresses 0x4001 C060) bit description .....	271
Table 233. Control register (CTRL[0:3], address 0x4000 4008 (CTRL0) to 0x4000 4038 (CTRL3)) bit description .....	243	Table 259. A/D Interrupt Enable register (INTEN, address 0x4001 C064 ) bit description .....	273
Table 234. Status register (STAT[0:3], address 0x4000 400C (STAT0) to 0x4000 403C (STAT3)) bit description 244		Table 260. A/D Flags register (FLAGS, address 0x4001 C068) bit description .....	275
Table 235. Module Configuration register (MODCFG, offset 0xF0) bit description .....	245	Table 261. Analog comparator pin description .....	282
Table 236. Idle channel register (IDLE_CH, address 0x4000 40F4) bit description .....	245	Table 262. Register overview: Analog comparator (base address 0x4002 4000) .....	284
Table 237. Global interrupt flag register (IRQ_FLAG, address 0x4000 40F8) bit description .....	246	Table 263. Comparator control register (CTRL, address 0x4002 4000) bit description .....	284
Table 238. Register overview: SysTick timer (base address 0xE000 E000) .....	248	Table 264. Voltage ladder register (LAD, address 0x4002 4004) bit description .....	286
Table 239. SysTick Timer Control and status register (SYST_CSR, 0xE000 E010) bit description ..	249	Table 265. Register overview: CRC engine (base address 0x5000 0000) .....	289
Table 240. System Timer Reload value register (SYST_RVR, 0xE000 E014) bit description ..	249	Table 266. CRC mode register (MODE, address 0x5000 0000) bit description .....	289
Table 241. System Timer Current value register (SYST_CVR, 0xE000 E018) bit description ..	249	Table 267. CRC seed register (SEED, address 0x5000 0004) bit description .....	290
Table 242. System Timer Calibration value register (SYST_CALIB, 0xE000 E01C) bit description 250		Table 268. CRC checksum register (SUM, address 0x5000 0008) bit description .....	290
Table 243. Pinout summary .....	251	Table 269. CRC data register (WR_DATA, address 0x5000 0008) bit description .....	290
Table 244. ADC hardware trigger inputs .....	254	Table 270. Divide API calls .....	293
Table 245. ADC supply and reference voltage pins. ....	255	Table 271. sidiv .....	293
Table 246. ADC pin description .....	255	Table 272. uidiv .....	294
Table 247. Register overview: ADC (base address 0x4001 C000 ) .....	257	Table 273. sidivmod .....	294
Table 248. A/D Control Register (CTRL, addresses 0x4001 C000) bit description .....	258	Table 274. uidivmod .....	294
Table 249. A/D Conversion Sequence A Control Register (SEQA_CTRL, address 0x4001 C008) bit description .....	260	Table 275. SWD pin description .....	296
Table 250. A/D Conversion Sequence B Control Register (SEQB_CTRL, address 0x4001 C00C) bit description .....	262	Table 276. JTAG boundary scan pin description .....	297
Table 251. A/D Sequence A Global Data Register (SEQA_GDAT, address 0x4001 C010) bit description .....	265	Table 277. Abbreviations .....	299
Table 252. A/D Sequence B Global Data Register (SEQB_GDAT, address 0x4001 C014) bit description .....	266		
Table 253. A/D Data Registers (DAT[0:11], address 0x4001 C020 (DAT0) to 0x4001 C04C (DAT11)) bit description .....	268		
Table 254. A/D Compare Low Threshold register 0 (THR0_LOW, address 0x4001 C050) bit description .....	270		
Table 255. A/D Compare Low Threshold register 1 (THR1_LOW, address 0x4001 C054) bit description .....	270		
Table 256. Compare High Threshold register0 (THR0_HIGH, address 0x4001 C058) bit description .....	270		
Table 257. Compare High Threshold register 1 (THR1_HIGH, address 0x4001 C05C) bit description .....	271		
Table 258. A/D Channel Threshold Select register			

## 26.5 Figures

Fig 1.	LPC802 block diagram . . . . .	9	Fig 47.	Comparator block diagram . . . . .	283
Fig 2.	LPC802 Memory mapping . . . . .	11	Fig 48.	CRC block diagram . . . . .	288
Fig 3.	Boot ROM structure . . . . .	13	Fig 49.	ROM pointer structure . . . . .	292
Fig 4.	Boot process flowchart. . . . .	15	Fig 50.	Connecting the SWD pins to a standard SWD connector. . . . .	298
Fig 5.	IAP parameter passing . . . . .	32			
Fig 6.	LPC802 clock generation. . . . .	50			
Fig 7.	LPC802 clock generation (continued) . . . . .	51			
Fig 8.	LPC802 WKT clocking . . . . .	52			
Fig 9.	LPC802 FRO subsystem . . . . .	52			
Fig 10.	ROM pointer structure . . . . .	74			
Fig 11.	Example: Connect function U0_RXD and U0_TXD to pins 4 and 14 . . . . .	77			
Fig 12.	Functional diagram of the switch matrix. . . . .	79			
Fig 13.	Pin configuration . . . . .	89			
Fig 14.	Pin interrupt connections . . . . .	111			
Fig 15.	Pattern match engine connections. . . . .	112			
Fig 16.	Pattern match bit slice with detect logic. . . . .	113			
Fig 17.	Pattern match engine examples: sticky edge detect 132				
Fig 18.	Pattern match engine examples: Windowed non-sticky edge detect evaluates as true. . . . .	133			
Fig 19.	Pattern match engine examples: Windowed non-sticky edge detect evaluates as false . . . . .	133			
Fig 20.	USART clocking. . . . .	147			
Fig 21.	USART block diagram . . . . .	150			
Fig 22.	Hardware flow control using RTS and CTS . . . . .	163			
Fig 23.	SPI clocking . . . . .	165			
Fig 24.	SPI block diagram . . . . .	168			
Fig 25.	Basic SPI operating modes . . . . .	179			
Fig 26.	Pre_delay and Post_delay . . . . .	180			
Fig 27.	Frame_delay . . . . .	181			
Fig 28.	Transfer_delay . . . . .	182			
Fig 29.	Examples of data stalls . . . . .	185			
Fig 30.	I2C clocking . . . . .	186			
Fig 31.	I2C block diagram . . . . .	191			
Fig 32.	32-bit counter/timer block diagram. . . . .	212			
Fig 33.	A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled . . . . .	224			
Fig 34.	A timer cycle in which PR=2, MRx=6, and both interrupt and stop on match are enabled . . . . .	224			
Fig 35.	Sample PWM waveforms with a PWM cycle length of 100 (selected by MR3) and MAT3:0 enabled as PWM outputs by the PWCON register. . . . .	225			
Fig 36.	USART clocking. . . . .	227			
Fig 37.	Windowed Watchdog timer block diagram. . . . .	228			
Fig 38.	Early watchdog feed with windowed mode enabled 234				
Fig 39.	Correct watchdog feed with windowed mode enabled . . . . .	234			
Fig 40.	Watchdog warning interrupt. . . . .	234			
Fig 41.	WKT clocking . . . . .	236			
Fig 42.	MRT clocking . . . . .	239			
Fig 43.	MRT block diagram . . . . .	240			
Fig 44.	System tick timer block diagram . . . . .	247			
Fig 45.	ADC clocking . . . . .	252			
Fig 46.	ADC block diagram . . . . .	256			

## 26.6 Contents

### Chapter 1: LPC802 Introductory information

<b>1.1</b>	<b>Introduction</b> .....	<b>4</b>	<b>1.4</b>	<b>General description</b> .....	<b>8</b>
<b>1.2</b>	<b>Features</b> .....	<b>4</b>	<b>1.4.1</b>	Arm Cortex-M0+ core configuration .....	8
<b>1.3</b>	<b>Ordering options</b> .....	<b>7</b>	<b>1.5</b>	<b>Block diagram</b> .....	<b>9</b>

### Chapter 2: LPC802 memory mapping

<b>2.1</b>	<b>How to read this chapter</b> .....	<b>10</b>	<b>2.2.1</b>	Memory mapping .....	11
<b>2.2</b>	<b>General description</b> .....	<b>10</b>			

### Chapter 3: LPC802 Boot Process

<b>3.1</b>	<b>How to read this chapter</b> .....	<b>12</b>	<b>3.4.1</b>	Bootloader .....	12
<b>3.2</b>	<b>Features</b> .....	<b>12</b>	<b>3.4.2</b>	ROM-based APIs .....	13
<b>3.3</b>	<b>Pin description</b> .....	<b>12</b>	<b>3.5</b>	<b>Functional description</b> .....	<b>14</b>
<b>3.4</b>	<b>General description</b> .....	<b>12</b>	<b>3.5.1</b>	Memory map after any reset .....	14
			<b>3.5.2</b>	Boot process .....	15

### Chapter 4: LPC802 ISP and IAP

<b>4.1</b>	<b>How to read this chapter</b> .....	<b>16</b>	<b>4.5.6</b>	Prepare sectors for write operation .....	24
<b>4.2</b>	<b>Features</b> .....	<b>16</b>	<b>4.5.7</b>	Copy RAM to flash .....	24
<b>4.3</b>	<b>General description</b> .....	<b>16</b>	<b>4.5.8</b>	Go .....	25
4.3.1	Boot loader .....	16	<b>4.5.9</b>	Erase sectors .....	26
4.3.2	Memory map after any reset .....	16	<b>4.5.10</b>	Erase pages .....	26
4.3.3	Flash content protection mechanism .....	16	<b>4.5.11</b>	Blank check sectors .....	26
4.3.4	Criteria for Valid User Code .....	17	<b>4.5.12</b>	Read Part Identification number .....	27
4.3.5	Flash partitions .....	17	<b>4.5.13</b>	Read Boot code version number .....	27
4.3.6	Code Read Protection (CRP) .....	18	<b>4.5.14</b>	Compare .....	28
4.3.6.1	ISP entry protection .....	19	<b>4.5.15</b>	ReadUID .....	28
4.3.6.2	ISP entry configuration and detection .....	19	<b>4.5.16</b>	Read CRC checksum .....	28
4.3.7	ISP interrupt and SRAM use .....	19	<b>4.5.17</b>	ISP/IAP Error codes .....	30
4.3.7.1	Interrupts during IAP .....	19	<b>4.6</b>	<b>IAP commands</b> .....	<b>31</b>
4.3.7.2	RAM used by ISP command handlers .....	19	<b>4.6.1</b>	Prepare sector(s) for write operation .....	33
4.3.7.3	RAM used by IAP command handlers .....	19	<b>4.6.2</b>	Copy RAM to flash .....	33
<b>4.4</b>	<b>USART ISP communication protocol</b> .....	<b>20</b>	<b>4.6.3</b>	Erase Sector(s) .....	34
4.4.1	USART ISP initialization .....	20	<b>4.6.4</b>	Blank check sector(s) .....	34
4.4.2	USART ISP command format .....	20	<b>4.6.5</b>	Read Part Identification number .....	34
4.4.3	USART ISP response format .....	20	<b>4.6.6</b>	Read Boot code version number .....	35
4.4.4	USART ISP data format .....	20	<b>4.6.7</b>	Compare <address1> <address2> <no of bytes> .....	35
<b>4.5</b>	<b>USART ISP commands</b> .....	<b>21</b>	<b>4.6.8</b>	Reinvoke ISP .....	35
4.5.1	Unlock .....	22	<b>4.6.9</b>	ReadUID .....	36
4.5.2	Set Baud Rate .....	22	<b>4.6.10</b>	Erase page .....	36
4.5.3	Echo .....	23	<b>4.6.11</b>	IAP Error Codes .....	36
4.5.4	Write to RAM .....	23			
4.5.5	Read Memory .....	23			

### Chapter 5: LPC802 Nested Vectored Interrupt Controller (NVIC)

<b>5.1</b>	<b>How to read this chapter</b> .....	<b>37</b>	<b>5.3.3</b>	Vector table offset .....	39
<b>5.2</b>	<b>Features</b> .....	<b>37</b>	<b>5.4</b>	<b>Register description</b> .....	<b>40</b>
<b>5.3</b>	<b>General description</b> .....	<b>37</b>	<b>5.4.1</b>	.... Interrupt Set Enable Register 0 register .....	41
5.3.1	Interrupt sources .....	37	<b>5.4.2</b>	Interrupt clear enable register 0 .....	41
5.3.2	Non-Maskable Interrupt (NMI) .....	39	<b>5.4.3</b>	.... Interrupt Set Pending Register 0 register .....	42

5.4.4	Interrupt Clear Pending Register 0 register	43	5.4.9	Interrupt Priority Register 4	46
5.4.5	Interrupt Priority Register 0	44	5.4.10	Interrupt Priority Register 5	46
5.4.6	Interrupt Priority Register 1	45	5.4.11	Interrupt Priority Register 6	46
5.4.7	Interrupt Priority Register 2	45	5.4.12	Interrupt Priority Register 7	47
5.4.8	Interrupt Priority Register 3	45			

## Chapter 6: LPC802 System configuration (SYSCON)

<b>6.1</b>	<b>How to read this chapter</b>	<b>48</b>	6.6.11	Peripheral reset control 1 register	62
<b>6.2</b>	<b>Features</b>	<b>48</b>	6.6.12	Peripheral clock source select registers	63
<b>6.3</b>	<b>Basic configuration</b>	<b>48</b>	6.6.13	Fractional generator 0 divider value register	63
6.3.1	Set up the FRO	48	6.6.14	Fractional generator 0 multiplier value register	64
6.3.2	Configure the main clock and system clock	49	6.6.15	FRG0 clock source select register	65
<b>6.4</b>	<b>Pin description</b>	<b>49</b>	6.6.16	CLKOUT clock source select register	65
<b>6.5</b>	<b>General description</b>	<b>50</b>	6.6.17	CLKOUT clock divider register	65
6.5.1	Clock generation	50	6.6.18	POR captured PIO0 status register 0	66
6.5.2	Power control of analog components	52	6.6.19	BOD control register	66
6.5.3	Configuration of reduced power-modes	53	6.6.20	System tick counter calibration register	66
6.5.4	Reset and interrupt control	53	6.6.21	IRQ latency register	67
<b>6.6</b>	<b>Register description</b>	<b>53</b>	6.6.22	NMI source selection register	67
6.6.1	System memory remap register	55	6.6.23	Pin interrupt select registers	68
6.6.2	System reset status register	56	6.6.24	Start logic 0 pin wake-up enable register	68
6.6.3	Main clock source select register	56	6.6.25	Start logic 1 interrupt wake-up enable register	69
6.6.4	Main clock source update enable register	57	6.6.26	Deep-sleep mode configuration register	70
6.6.5	System clock divider register	57	6.6.27	Wake-up configuration register	71
6.6.6	ADC clock source select register	57	6.6.28	Power configuration register	71
6.6.7	ADC clock divider register	58	6.6.29	Device ID register	72
6.6.8	WDT and Wake Timer clock enable control register	58	<b>6.7</b>	<b>Functional description</b>	<b>73</b>
6.6.9	System clock control 0 register	58	6.7.1	Reset	73
6.6.10	Peripheral reset control 0 register	61	6.7.2	Brown-out detection	73

## Chapter 7: LPC802 FRO API ROM routine

<b>7.1</b>	<b>How to read this chapter</b>	<b>74</b>	<b>7.4</b>	<b>API description</b>	<b>75</b>
<b>7.2</b>	<b>Features</b>	<b>74</b>	7.4.1	set_fro_frequency	75
<b>7.3</b>	<b>General description</b>	<b>74</b>	7.4.1.1	Param0: frequency	75

## Chapter 8: LPC802 Switch matrix (SWM)

<b>8.1</b>	<b>How to read this chapter</b>	<b>76</b>	<b>8.5</b>	<b>Register description</b>	<b>82</b>
<b>8.2</b>	<b>Features</b>	<b>76</b>	8.5.1	Pin assign register 0	83
<b>8.3</b>	<b>Basic configuration</b>	<b>76</b>	8.5.2	Pin assign register 1	83
8.3.1	Connect an internal signal to a package pin	77	8.5.3	Pin assign register 2	83
8.3.2	Enable an analog input or other special function	78	8.5.4	Pin assign register 3	84
8.3.3	Changing the pin function assignment	78	8.5.5	Pin assign register 4	84
<b>8.4</b>	<b>General description</b>	<b>79</b>	8.5.6	Pin assign register 5	85
8.4.1	Movable functions	80	8.5.7	Pin assign register 6	85
8.4.2	Switch matrix register interface	81	8.5.8	Pin assign register 7	85
8.4.3	Level Shifter function	82	8.5.9	PINENABLE 0	86

## Chapter 9: LPC802 I/O configuration (IOCON)

<b>9.1</b>	<b>How to read this chapter</b>	<b>88</b>	<b>9.4</b>	<b>General description</b>	<b>89</b>
<b>9.2</b>	<b>Features</b>	<b>88</b>	9.4.1	Pin configuration	89
<b>9.3</b>	<b>Basic configuration</b>	<b>88</b>	9.4.2	Pin function	89



9.4.3	Pin mode . . . . .	89	9.5.8	PIO0_11 register . . . . .	96
9.4.4	Open-drain mode . . . . .	90	9.5.9	PIO0_10 register . . . . .	96
9.4.5	Analog mode . . . . .	90	9.5.10	PIO0_16 register . . . . .	97
<b>9.5</b>	<b>Register description . . . . .</b>	<b>90</b>	9.5.11	PIO0_15 register . . . . .	97
9.5.1	PIO0_17 register . . . . .	91	9.5.12	PIO0_1 register . . . . .	98
9.5.2	PIO0_13 register . . . . .	92	9.5.13	PIO0_9 register . . . . .	98
9.5.3	PIO0_12 register . . . . .	93	9.5.14	PIO0_8 register . . . . .	99
9.5.4	PIO0_5 register . . . . .	93	9.5.15	PIO0_7 register . . . . .	99
9.5.5	PIO0_4 register . . . . .	94	9.5.16	PIO0_0 register . . . . .	100
9.5.6	PIO0_3 register . . . . .	94	9.5.17	PIO0_14 register . . . . .	101
9.5.7	PIO0_2 register . . . . .	95			

## Chapter 10: LPC802 General Purpose I/O (GPIO)

<b>10.1</b>	<b>How to read this chapter . . . . .</b>	<b>102</b>	10.5.8	GPIO port clear registers . . . . .	105
<b>10.2</b>	<b>Basic configuration . . . . .</b>	<b>102</b>	10.5.9	GPIO port toggle registers . . . . .	106
<b>10.3</b>	<b>Features . . . . .</b>	<b>102</b>	10.5.10	GPIO port direction set registers . . . . .	106
<b>10.4</b>	<b>General description . . . . .</b>	<b>102</b>	10.5.11	GPIO port direction clear registers . . . . .	106
<b>10.5</b>	<b>Register description . . . . .</b>	<b>102</b>	10.5.12	GPIO port direction toggle registers . . . . .	107
10.5.1	GPIO port byte pin registers . . . . .	103	<b>10.6</b>	<b>Functional description . . . . .</b>	<b>107</b>
10.5.2	GPIO port word pin registers . . . . .	103	10.6.1	Reading pin state . . . . .	107
10.5.3	GPIO port direction registers . . . . .	104	10.6.2	GPIO output . . . . .	107
10.5.4	GPIO port mask registers . . . . .	104	10.6.3	Masked I/O . . . . .	108
10.5.5	GPIO port pin registers . . . . .	104	10.6.4	GPIO direction . . . . .	108
10.5.6	GPIO masked port pin registers . . . . .	105	10.6.5	Recommended practices . . . . .	108
10.5.7	GPIO port set registers . . . . .	105			

## Chapter 11: LPC802 Pin interrupts/pattern match engine

<b>11.1</b>	<b>How to read this chapter . . . . .</b>	<b>109</b>	11.6.4	Pin interrupt level or rising edge interrupt clear register . . . . .	116
<b>11.2</b>	<b>Features . . . . .</b>	<b>109</b>	11.6.5	Pin interrupt active level or falling edge interrupt enable register . . . . .	117
<b>11.3</b>	<b>Basic configuration . . . . .</b>	<b>109</b>	11.6.6	Pin interrupt active level or falling edge interrupt set register . . . . .	117
11.3.1	Configure pins as pin interrupts or as inputs to the pattern match engine . . . . .	110	11.6.7	Pin interrupt active level or falling edge interrupt clear register . . . . .	118
<b>11.4</b>	<b>Pin description . . . . .</b>	<b>110</b>	11.6.8	Pin interrupt rising edge register . . . . .	118
<b>11.5</b>	<b>General description . . . . .</b>	<b>110</b>	11.6.9	Pin interrupt falling edge register . . . . .	119
11.5.1	Pin interrupts . . . . .	111	11.6.10	Pin interrupt status register . . . . .	119
11.5.2	Pattern match engine . . . . .	111	11.6.11	Pattern Match Interrupt Control Register . . . . .	119
11.5.2.1	Inputs and outputs of the pattern match engine . . . . .	113	11.6.12	Pattern Match Interrupt Bit-Slice Source register . . . . .	120
11.5.2.2	Boolean expressions . . . . .	114	11.6.13	Pattern Match Interrupt Bit Slice Configuration register . . . . .	124
<b>11.6</b>	<b>Register description . . . . .</b>	<b>115</b>	<b>11.7</b>	<b>Functional description . . . . .</b>	<b>130</b>
11.6.1	Pin interrupt mode register . . . . .	115	11.7.1	Pin interrupts . . . . .	130
11.6.2	Pin interrupt level or rising edge interrupt enable register . . . . .	116	11.7.2	Pattern Match engine example . . . . .	131
11.6.3	Pin interrupt level or rising edge interrupt set register . . . . .	116	11.7.3	Pattern match engine edge detect examples . . . . .	132

## Chapter 12: LPC802 Reduced power modes and power management

<b>12.1</b>	<b>How to read this chapter . . . . .</b>	<b>134</b>	<b>12.4</b>	<b>Pin description . . . . .</b>	<b>135</b>
<b>12.2</b>	<b>Features . . . . .</b>	<b>134</b>	<b>12.5</b>	<b>General description . . . . .</b>	<b>136</b>
<b>12.3</b>	<b>Basic configuration . . . . .</b>	<b>134</b>	12.5.1	Wake-up process . . . . .	136
12.3.1	Low power modes in the Arm Cortex-M0+ core . . . . .	134	<b>12.6</b>	<b>Register description . . . . .</b>	<b>137</b>
12.3.1.1	System control register . . . . .	134	12.6.1	Power control register . . . . .	138
			12.6.2	General purpose registers 0 to 4 . . . . .	138

12.6.3	Deep power-down wake-up source status register	139	12.7.5.1	Power configuration in deep-sleep mode . . .	142
12.6.4	Deep power-down wake-up enable register .	139	12.7.5.2	Programming deep-sleep mode . . . . .	142
<b>12.7</b>	<b>Functional description . . . . .</b>	<b>140</b>	12.7.5.3	Wake-up from deep-sleep mode . . . . .	142
12.7.1	Power management . . . . .	140	12.7.6	Power-down mode . . . . .	143
12.7.2	Reduced power modes and WWDT lock features	140	12.7.6.1	Power configuration in power-down mode . .	143
12.7.3	Active mode . . . . .	140	12.7.6.2	Programming power-down mode . . . . .	144
12.7.3.1	Power configuration in Active mode . . . . .	141	12.7.6.3	Wake-up from power-down mode . . . . .	144
12.7.4	Sleep mode . . . . .	141	12.7.7	Deep power-down mode . . . . .	144
12.7.4.1	Power configuration in sleep mode . . . . .	141	12.7.7.1	Power configuration in deep power-down mode .	145
12.7.4.2	Programming sleep mode . . . . .	141	12.7.7.2	Programming deep power-down mode using the	
12.7.4.3	Wake-up from sleep mode . . . . .	142		WAKEUP pin: . . . . .	145
12.7.5	Deep-sleep mode . . . . .	142	12.7.7.3	Wake-up from deep power-down mode using the	
				WAKEUP pin: . . . . .	145

## Chapter 13: LPC802 USART0/1

<b>13.1</b>	<b>How to read this chapter . . . . .</b>	<b>146</b>	13.6.7	USART Receiver Data with Status register .	159
<b>13.2</b>	<b>Features . . . . .</b>	<b>146</b>	13.6.8	USART Transmitter Data Register . . . . .	159
<b>13.3</b>	<b>Basic configuration . . . . .</b>	<b>146</b>	13.6.9	USART Baud Rate Generator register . . . . .	159
13.3.1	Configure the USART clock and baud rate .	147	13.6.10	USART Interrupt Status register . . . . .	160
13.3.2	Configure the USART for wake-up . . . . .	148	13.6.11	USART Oversample selection register . . . .	161
13.3.2.1	Wake-up from sleep mode . . . . .	148	13.6.12	USART Address register . . . . .	161
13.3.2.2	Wake-up from deep-sleep or power-down mode .	148	<b>13.7</b>	<b>Functional description . . . . .</b>	<b>162</b>
<b>13.4</b>	<b>Pin description . . . . .</b>	<b>149</b>	13.7.1	Clocking and baud rates . . . . .	162
<b>13.5</b>	<b>General description . . . . .</b>	<b>149</b>	13.7.1.1	Fractional Rate Generator (FRG) . . . . .	162
<b>13.6</b>	<b>Register description . . . . .</b>	<b>151</b>	13.7.1.2	Baud Rate Generator (BRG) . . . . .	162
13.6.1	USART Configuration register . . . . .	152	13.7.1.3	Baud rate calculations . . . . .	162
13.6.2	USART Control register . . . . .	154	13.7.2	Synchronous mode . . . . .	162
13.6.3	USART Status register . . . . .	155	13.7.3	Flow control . . . . .	162
13.6.4	USART Interrupt Enable read and set register .	157	13.7.3.1	Hardware flow control . . . . .	163
13.6.5	USART Interrupt Enable Clear register . . . .	158	13.7.3.2	Software flow control . . . . .	163
13.6.6	USART Receiver Data register . . . . .	158	13.7.4	Autobaud function . . . . .	163
			13.7.5	RS-485 support . . . . .	164
			13.7.6	Oversampling . . . . .	164

## Chapter 14: LPC802 SPI

<b>14.1</b>	<b>How to read this chapter . . . . .</b>	<b>165</b>	14.6.10	SPI Divider register . . . . .	178
<b>14.2</b>	<b>Features . . . . .</b>	<b>165</b>	14.6.11	SPI Interrupt Status register . . . . .	178
<b>14.3</b>	<b>Basic configuration . . . . .</b>	<b>165</b>	<b>14.7</b>	<b>Functional description . . . . .</b>	<b>179</b>
14.3.1	Configure the SPI for wake-up . . . . .	166	14.7.1	Operating modes: clock and phase selection	179
14.3.1.1	Wake-up from sleep mode . . . . .	166	14.7.2	Frame delays . . . . .	180
14.3.1.2	Wake up from deep-sleep or power-down mode .	166	14.7.2.1	Pre_delay and Post_delay . . . . .	180
<b>14.4</b>	<b>Pin description . . . . .</b>	<b>166</b>	14.7.2.2	Frame_delay . . . . .	181
<b>14.5</b>	<b>General description . . . . .</b>	<b>168</b>	14.7.2.3	Transfer_delay . . . . .	182
<b>14.6</b>	<b>Register description . . . . .</b>	<b>168</b>	14.7.3	Clocking and data rates . . . . .	183
14.6.1	SPI Configuration register . . . . .	169	14.7.3.1	Data rate calculations . . . . .	183
14.6.2	SPI Delay register . . . . .	170	14.7.4	Slave select . . . . .	183
14.6.3	SPI Status register . . . . .	172	14.7.5	Data lengths greater than 16 bits . . . . .	183
14.6.4	SPI Interrupt Enable read and Set register .	173	14.7.6	Data stalls . . . . .	184
14.6.5	SPI Interrupt Enable Clear register . . . . .	174			
14.6.6	SPI Receiver Data register . . . . .	174			
14.6.7	SPI Transmitter Data and Control register .	176			
14.6.8	SPI Transmitter Data Register . . . . .	177			
14.6.9	SPI Transmitter Control register . . . . .	177			

**Chapter 15: LPC802 I2C**

<b>15.1</b>	<b>How to read this chapter</b> . . . . .	<b>186</b>	15.6.4	Interrupt Enable Clear register . . . . .	200
<b>15.2</b>	<b>Features</b> . . . . .	<b>186</b>	15.6.5	Time-out value register . . . . .	200
<b>15.3</b>	<b>Basic configuration</b> . . . . .	<b>186</b>	15.6.6	Clock Divider register . . . . .	201
15.3.1	I <sup>2</sup> C transmit/receive in master mode . . . . .	187	15.6.7	Interrupt Status register . . . . .	202
15.3.1.1	Master write to slave . . . . .	188	15.6.8	Master Control register . . . . .	202
15.3.1.2	Master read from slave . . . . .	188	15.6.9	Master Time . . . . .	203
15.3.2	I <sup>2</sup> C receive/transmit in slave mode . . . . .	188	15.6.10	Master Data register . . . . .	204
15.3.2.1	Slave read from master . . . . .	189	15.6.11	Slave Control register . . . . .	204
15.3.2.2	Slave write to master . . . . .	189	15.6.12	Slave Data register . . . . .	205
15.3.3	Configure the I <sup>2</sup> C for wake-up . . . . .	190	15.6.13	Slave Address registers . . . . .	205
15.3.3.1	Wake-up from sleep mode . . . . .	190	15.6.14	Slave address Qualifier 0 register . . . . .	205
15.3.3.2	Wake-up from deep-sleep and power-down modes . . . . .	190	15.6.15	Monitor data register . . . . .	206
<b>15.4</b>	<b>Pin description</b> . . . . .	<b>191</b>	<b>15.7</b>	<b>Functional description</b> . . . . .	<b>208</b>
<b>15.5</b>	<b>General description</b> . . . . .	<b>191</b>	15.7.1	Bus rates and timing considerations . . . . .	208
<b>15.6</b>	<b>Register description</b> . . . . .	<b>191</b>	15.7.1.1	Rate calculations . . . . .	208
15.6.1	I <sup>2</sup> C Configuration register . . . . .	193	15.7.2	Time-out . . . . .	208
15.6.2	I <sup>2</sup> C Status register . . . . .	195	15.7.3	Ten-bit addressing . . . . .	209
15.6.3	Interrupt Enable Set and read register . . . . .	199	15.7.4	Clocking and power considerations . . . . .	209
			15.7.5	Interrupt handling . . . . .	209

**Chapter 16: LPC802 Standard counter/timer (CTIMER)**

<b>16.1</b>	<b>How to read this chapter</b> . . . . .	<b>210</b>	16.6.3	Timer Counter registers . . . . .	216
<b>16.2</b>	<b>Features</b> . . . . .	<b>210</b>	16.6.4	Prescale register . . . . .	217
<b>16.3</b>	<b>Basic configuration</b> . . . . .	<b>210</b>	16.6.5	Prescale Counter register . . . . .	217
<b>16.4</b>	<b>General description</b> . . . . .	<b>211</b>	16.6.6	Match Control Register . . . . .	217
16.4.1	Capture inputs . . . . .	211	16.6.7	Match Registers . . . . .	218
16.4.2	Match outputs . . . . .	211	16.6.8	Capture Control Register . . . . .	218
16.4.3	Applications . . . . .	211	16.6.9	Capture Registers . . . . .	219
16.4.4	Architecture . . . . .	211	16.6.10	External Match Register . . . . .	219
<b>16.5</b>	<b>Pin description</b> . . . . .	<b>213</b>	16.6.11	Count Control Register . . . . .	221
16.5.1	Multiple CAP and MAT pins . . . . .	213	16.6.12	PWM Control Register . . . . .	222
<b>16.6</b>	<b>Register description</b> . . . . .	<b>214</b>	16.6.13	Match Shadow Registers . . . . .	223
16.6.1	Interrupt Register . . . . .	216	<b>16.7</b>	<b>Functional description</b> . . . . .	<b>224</b>
16.6.2	Timer Control Register . . . . .	216	16.7.1	Rules for single edge controlled PWM outputs . . . . .	224

**Chapter 17: LPC802 Windowed Watchdog Timer (WWDT)**

<b>17.1</b>	<b>How to read this chapter</b> . . . . .	<b>226</b>	17.5.3.2	Changing the WWDT reload value . . . . .	229
<b>17.2</b>	<b>Features</b> . . . . .	<b>226</b>	<b>17.6</b>	<b>Register description</b> . . . . .	<b>230</b>
<b>17.3</b>	<b>Basic configuration</b> . . . . .	<b>226</b>	17.6.1	Watchdog mode register . . . . .	230
<b>17.4</b>	<b>Pin description</b> . . . . .	<b>227</b>	17.6.2	Watchdog Timer Constant register . . . . .	232
<b>17.5</b>	<b>General description</b> . . . . .	<b>227</b>	17.6.3	Watchdog Feed register . . . . .	232
17.5.1	Block diagram . . . . .	228	17.6.4	Watchdog Timer Value register . . . . .	233
17.5.2	Clocking and power control . . . . .	228	17.6.5	Watchdog Timer Warning Interrupt register . . . . .	233
17.5.3	Using the WWDT lock features . . . . .	229	17.6.6	Watchdog Timer Window register . . . . .	233
17.5.3.1	Disabling the WWDT clock source . . . . .	229	<b>17.7</b>	<b>Functional description</b> . . . . .	<b>234</b>

**Chapter 18: LPC802 Self-wake-up timer (WKT)**

<b>18.1</b>	<b>How to read this chapter</b> . . . . .	<b>235</b>	<b>18.5</b>	<b>General description</b> . . . . .	<b>236</b>
<b>18.2</b>	<b>Features</b> . . . . .	<b>235</b>	18.5.1	WKT clock sources . . . . .	236
<b>18.3</b>	<b>Basic configuration</b> . . . . .	<b>235</b>	<b>18.6</b>	<b>Register description</b> . . . . .	<b>237</b>
<b>18.4</b>	<b>Pin description</b> . . . . .	<b>236</b>	18.6.1	Control register . . . . .	237



18.6.2	Count register . . . . .	238
--------	--------------------------	-----

## Chapter 19: LPC802 Multi-Rate Timer (MRT)

<b>19.1</b>	<b>How to read this chapter . . . . .</b>	<b>239</b>	<b>19.6</b>	<b>Register description . . . . .</b>	<b>242</b>
<b>19.2</b>	<b>Features . . . . .</b>	<b>239</b>	19.6.1	Time interval register . . . . .	242
<b>19.3</b>	<b>Basic configuration . . . . .</b>	<b>239</b>	19.6.2	Timer register . . . . .	243
<b>19.4</b>	<b>Pin description . . . . .</b>	<b>239</b>	19.6.3	Control register . . . . .	243
<b>19.5</b>	<b>General description . . . . .</b>	<b>239</b>	19.6.4	Status register . . . . .	243
19.5.1	Repeat interrupt mode . . . . .	240	19.6.5	Module Configuration register . . . . .	245
19.5.2	One-shot interrupt mode . . . . .	241	19.6.6	Idle channel register . . . . .	245
19.5.3	One-shot bus stall mode . . . . .	241	19.6.7	Global interrupt flag register . . . . .	245

## Chapter 20: LPC802 System tick timer (SysTick)

<b>20.1</b>	<b>How to read this chapter . . . . .</b>	<b>247</b>	20.6.1	System Timer Control and status register . .	248
<b>20.2</b>	<b>Features . . . . .</b>	<b>247</b>	20.6.2	System Timer Reload value register . . . . .	249
<b>20.3</b>	<b>Basic configuration . . . . .</b>	<b>247</b>	20.6.3	System Timer Current value register . . . . .	249
<b>20.4</b>	<b>Pin description . . . . .</b>	<b>247</b>	20.6.4	System Timer Calibration value register . . .	250
<b>20.5</b>	<b>General description . . . . .</b>	<b>247</b>	<b>20.7</b>	<b>Functional description . . . . .</b>	<b>250</b>
<b>20.6</b>	<b>Register description . . . . .</b>	<b>248</b>	20.7.1	Example timer calculation . . . . .	250
				Example (system clock = 20 MHz) . . . . .	250

## Chapter 21: LPC802 12-bit Analog-to-Digital Converter (ADC)

<b>21.1</b>	<b>How to read this chapter . . . . .</b>	<b>251</b>	21.6.5	A/D Channel Data Registers 0 to 11 . . . . .	267
<b>21.2</b>	<b>Features . . . . .</b>	<b>251</b>	21.6.6	A/D Compare Low Threshold Registers 0 and 1 .	269
<b>21.3</b>	<b>Basic configuration . . . . .</b>	<b>251</b>	21.6.7	A/D Compare High Threshold Registers 0 and 1	270
21.3.1	Perform a single ADC conversion using a software trigger . . . . .	252	21.6.8	A/D Channel Threshold Select register . . . .	271
21.3.2	Perform a sequence of conversions triggered by an external pin . . . . .	253	21.6.9	A/D Interrupt Enable Register . . . . .	272
21.3.3	ADC hardware trigger inputs . . . . .	253	21.6.10	A/D Flag register . . . . .	275
21.3.4	Sample Time . . . . .	254	<b>21.7</b>	<b>Functional description . . . . .</b>	<b>277</b>
<b>21.4</b>	<b>Pin description . . . . .</b>	<b>254</b>	21.7.1	Conversion Sequences . . . . .	277
21.4.1	ADC versus digital receiver . . . . .	255	21.7.2	Hardware-triggered conversion . . . . .	277
<b>21.5</b>	<b>General description . . . . .</b>	<b>256</b>	21.7.2.1	Avoiding spurious hardware triggers . . . . .	278
<b>21.6</b>	<b>Register description . . . . .</b>	<b>257</b>	21.7.3	Software-triggered conversion . . . . .	278
21.6.1	ADC Control Register . . . . .	258	21.7.4	Interrupts . . . . .	279
21.6.2	A/D Conversion Sequence A Control Register . .	259	21.7.4.1	Conversion-Complete or Sequence-Complete interrupts . . . . .	279
21.6.3	A/D Conversion Sequence B Control Register . .	262	21.7.4.2	Threshold-Compare Out-of-Range Interrupt . .	279
21.6.4	A/D Global Data Register A and B . . . . .	264	21.7.4.3	Data Overrun Interrupt . . . . .	279
			21.7.5	Optional operating modes . . . . .	280
			21.7.6	Hardware Trigger Source Selection . . . . .	280

## Chapter 22: LPC802 Analog comparator

<b>22.1</b>	<b>How to read this chapter . . . . .</b>	<b>281</b>	22.5.1	Reference voltages . . . . .	283
<b>22.2</b>	<b>Features . . . . .</b>	<b>281</b>	22.5.2	Settling times . . . . .	283
<b>22.3</b>	<b>Basic configuration . . . . .</b>	<b>281</b>	22.5.3	Interrupts . . . . .	283
22.3.1	Connect the comparator output to the ADC . .	281	22.5.4	Comparator outputs . . . . .	284
<b>22.4</b>	<b>Pin description . . . . .</b>	<b>282</b>	<b>22.6</b>	<b>Register description . . . . .</b>	<b>284</b>
<b>22.5</b>	<b>General description . . . . .</b>	<b>282</b>	22.6.1	Comparator control register . . . . .	284
			22.6.2	Voltage ladder register . . . . .	286

## Chapter 23: LPC802 CRC engine

<b>23.1</b>	<b>How to read this chapter . . . . .</b>	<b>287</b>	<b>23.2</b>	<b>Features . . . . .</b>	<b>287</b>
-------------	---	------------	-------------	---------------------------	------------

<b>23.3</b>	<b>Basic configuration</b> . . . . .	<b>287</b>	<b>23.6.3</b>	CRC checksum register . . . . .	290
<b>23.4</b>	<b>Pin description</b> . . . . .	<b>287</b>	<b>23.6.4</b>	CRC data register . . . . .	290
<b>23.5</b>	<b>General description</b> . . . . .	<b>287</b>	<b>23.7</b>	<b>Functional description</b> . . . . .	<b>290</b>
<b>23.6</b>	<b>Register description</b> . . . . .	<b>289</b>	<b>23.7.1</b>	CRC-CCITT set-up . . . . .	290
23.6.1	CRC mode register . . . . .	289	<b>23.7.2</b>	CRC-16 set-up . . . . .	291
23.6.2	CRC seed register . . . . .	290	<b>23.7.3</b>	CRC-32 set-up . . . . .	291

## Chapter 24: LPC802 ROM API integer divide routines

<b>24.1</b>	<b>How to read this chapter</b> . . . . .	<b>292</b>	<b>24.4.3</b>	DIV signed integer division with remainder .	294
<b>24.2</b>	<b>Features</b> . . . . .	<b>292</b>	<b>24.4.4</b>	DIV unsigned integer division with remainder	294
<b>24.3</b>	<b>General description</b> . . . . .	<b>292</b>	<b>24.5</b>	<b>Functional description</b> . . . . .	<b>295</b>
<b>24.4</b>	<b>API description</b> . . . . .	<b>293</b>	<b>24.5.1</b>	Signed division . . . . .	295
24.4.1	DIV signed integer division . . . . .	293	<b>24.5.2</b>	Unsigned division with remainder . . . . .	295
24.4.2	DIV unsigned integer division . . . . .	294			

## Chapter 25: LPC802 Serial Wire Debug (SWD)

<b>25.1</b>	<b>How to read this chapter</b> . . . . .	<b>296</b>	<b>25.5</b>	<b>Functional description</b> . . . . .	<b>297</b>
<b>25.2</b>	<b>Features</b> . . . . .	<b>296</b>	<b>25.5.1</b>	Debug limitations . . . . .	297
<b>25.3</b>	<b>General description</b> . . . . .	<b>296</b>	<b>25.5.2</b>	Debug connections for SWD . . . . .	297
<b>25.4</b>	<b>Pin description</b> . . . . .	<b>296</b>	<b>25.5.3</b>	Boundary scan . . . . .	298

## Chapter 26: Supplementary information

<b>26.1</b>	<b>Abbreviations</b> . . . . .	<b>299</b>	<b>26.3.3</b>	Trademarks . . . . .	300
<b>26.2</b>	<b>References</b> . . . . .	<b>299</b>	<b>26.4</b>	<b>Tables</b> . . . . .	<b>301</b>
<b>26.3</b>	<b>Legal information</b> . . . . .	<b>300</b>	<b>26.5</b>	<b>Figures</b> . . . . .	<b>306</b>
26.3.1	Definitions . . . . .	300	<b>26.6</b>	<b>Contents</b> . . . . .	<b>307</b>
26.3.2	Disclaimers . . . . .	300			

## **How To Reach Us**

### **Home Page:**

[nxp.com](http://nxp.com)

### **Web Support:**

[nxp.com/support](http://nxp.com/support)

**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.

**Security** — Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at [PSIRT@nxp.com](mailto:PSIRT@nxp.com)) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

**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.

**Non-automotive qualified products** — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

**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

*Table continues on the next page...*

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.

**Limiting values** — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

**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.

**Evaluation products** — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out of the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, GREENCHIP, VIBES, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, Freescale, the Freescale logo, Altivec, the Energy Efficient Solutions logo, Kinetis, Layerscape, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, SafeAssure, the SafeAssure logo, Symphony, VortiQa, Vybrid, Airfast, CoreNet, Flexis, Platform in a Package, QUICC Engine, Tower, TurboLink, EdgeLock, eIQ, and Immersive3D are trademarks of NXP B.V. All other product or service names are the property of their respective owners. AMBA, Arm, Arm7, Arm7TDMI, Arm9, Arm11, Artisan, big.LITTLE, Cordio, CoreLink, CoreSight, Cortex, DesignStart, DynamIQ, Jazelle, Keil, Mali, Mbed, Mbed Enabled, NEON, POP, RealView, SecurCore, Socrates, Thumb, TrustZone, ULINK, ULINK2, ULINK-ME, ULINK-PLUS, ULINKpro, µVision, Versatile are trademarks or registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. The related technology may be protected by any or all of patents, copyrights, designs and trade secrets. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Synopsys and Designware are registered trademarks of Synopsys, Inc.



© NXP B.V. 2017-2021.

All rights reserved.

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

For sales office addresses, please send an email to: [salesaddresses@nxp.com](mailto:salesaddresses@nxp.com)

Date of release: 03/22/2021

Document identifier: UM11045