UM10360

LPC176x/5x User manual

Rev. 4. 1 — 19 December 2016 User manual

Document information

Info	Content
Keywords	LPC1769, LPC1768, LPC1767, LPC1766, LPC1765, LPC1764, LPC1763, LPC1759, LPC1758, LPC1756, LPC1754, LPC1752, LPC1751, ARM, ARM Cortex-M3, 32-bit, USB, Ethernet, CAN, I2S, Microcontroller
Abstract	LPC176x/5x user manual

Revision history

Rev	Date	Description
4.1	20161219	LPC176x/5x user manual
		Modifications: • Added a remark: The LPC175x devices require a standard I^2C connection to the USB ATX to Section 13.3 “Introduction”.
4	20161122	LPC176x/5x user manual
		Modifications: • Updated Table 89 “Pin Mode select register 1 (PINMODE1 - address 0x4002 C044) bit description”. Changed description to Port 0. • Added text: This bit is also cleared by an external reset to Table 8 “Reset Source Identification register (RSID - address 0x400F C180) bit description” and added bits 4 (SYSRESET) and 5 (LOCKUP). • Fixed typo in Table 44 “Power Mode Control register (PCON - address 0x400F C0C0) bit description”; bits 7:5 - Reserved. • Updated Table 74 “Pin description (LPC176x)”. $V_{DD(REG)(3V3)}$ F4 and F10 pins for WLCSP100. • Updated the address for Table 279 “UARTn FIFO Control Register (U0FCR - address 0x4000 C008, U2FCR - 0x4009 8008, U3FCR - 0x4009 C008) bit description”. • Deleted text from Section 19.1 “Basic configuration”: PINMODE_OD registers (open drain) (See Section 8.5). • Updated Figure 124 “Center-aligned PWM waveform without dead time, POLA = 0” and Figure 126 “Center-aligned waveform with dead time, POLA = 0”. • Updated text in Table 513 “RTC Auxiliary control register (RTC_AUX - address 0x4002 405C) bit description”: RTC_OSCF: RTC Oscillator Fail detect flag. Read: This bit is set when the RTC oscillator fails to toggle on the next cycle, and when RTC power is first turned on. An interrupt occurs when this bit is set, if enabled. The RTC_OSCFEN bit in RTC_AUXEN is a 1, and the RTC interrupt is enabled in the NVIC. • Added text to Section 29.3 “Description”: A non-burst mode conversion requires 65 clocks and a burst mode conversion requires 64 clocks. • Updated Table 532 “A/D Control Register (AD0CR - address 0x4003 4000) bit description”: enumerated values for bits 26:24; 010: Set the pin function to EINT0 in PINSEL4 register.and 011: Set the pin function to CAP0.1 in PINSEL3 register. • Added text: on-chip local RAM to Section 32.3.2.7 “RAM used by ISP command handler”. • Added text to the reset value in Table 524 “Watchdog Mode register (WDMOD, address 0x4000 0000) bit description”: (Only after POR).
3.1	20140402	LPC176x/5x user manual
		Modifications: • Added LPC1768UK. • Table 73 “Pin description (LPC175x)” and Table 74 “Pin description (LPC176x)”: Changed RX_MCLK and TX_MCLK type from INPUT to OUTPUT.

Revision history *...continued*

Rev	Date	Description
3	20131220	LPC176x/5x user manual
		Modifications: • Part ID for part LPC1763 added. • Changed title to “LPC176x/5x User manual”. • Updated numbering for CAN interfaces: CAN1 uses SCC = 0, CAN2 uses SCC = 1. See Section 16.13 “ID look-up table RAM” and Section 16.15 “Configuration and search algorithm”. • Updated Serial Wire Output description (Table 610). • Clarified burst mode information for ADGINTEN (Table 532 and Table 534). • Condition CCLK > 18 MHz for USB operation is not applicable for this USB peripheral and was removed (see Section 4.7.1, Section 11.13, and Section 13.11). • Description of CAN interrupt request updated. One common CAN interrupt is triggered. See Section 16.8.3. • Condition on minimum frequency of CAP input clock added in Section 21.5.1. • Description of RIMASK register corrected. See Table 434. • Condition for maximum allowable STCLK frequency added. See Section 23.4. • Delete statement “All PWM related Match registers are configured for toggle on match.” in Figure 121. • Description of INXCNT register updated. See Section 26.6.3.6. • Reset value of the RTC_AUX register corrected. See Table 508. • DAC power-down mode removed in Section 30.2. • Added: The DAC output is disabled in deep-sleep, power-down, or deep power-down modes. See Table 538. • Boot loader SRAM use explained. See Section 33.5. • SYSRESETREQ supported. See Table 660. • Figure 19 “Ethernet packet fields” corrected. • Bit description in the SPI test control register corrected. Bit 0 indicates test mode. All other bits are reserved. See Section 17.7.5 “SPI Test Control Register (SPTCR - 0x4002 0010)”. • Figure 118 “RI timer block diagram” updated.
2	20100819	LPC176x/5x user manual revision.
		Modifications: • UART0/1/2/3: FIFOLVL register removed. • ADC: reset value of the ADCTRM register changed to 0xF00 (Table 500). • Timer0/1/2/3: Description of DMA operation updated. • USB Device: Corrected error in the USBCmdCode register (0x01 = write, 0x02 = read) (Table 184). • Clocking and power control: add bit 15 (PCGPIO) to PCONP register (Table 46). • Part LPC1763 added. • Update register bit description of USBIntStat register in Host and Device mode (Table 155 and Table 221). • Motor control PWM: update description of match and limit registers. • GPIO: update register bit description of the FIOPIN register (Table 73). • Numerous editorial updates throughout the user manual.

Revision history *...continued*

Rev	Date	Description
1	20100104	LPC176x/5x user manual revision.

Contact information

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

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

1.1 Introduction

The LPC176x/5x is an ARM Cortex-M3 based microcontroller for embedded applications requiring a high level of integration and low power dissipation. The ARM Cortex-M3 is a next generation core that offers system enhancements such as modernized debug features and a higher level of support block integration.

High speed versions (LPC1769 and LPC1759) operate at up to a 120 MHz CPU frequency. Other versions operate at up to an 100 MHz CPU frequency. The ARM Cortex-M3 CPU incorporates a 3-stage pipeline and uses a Harvard architecture with separate local instruction and data buses as well as a third bus for peripherals. The ARM Cortex-M3 CPU also includes an internal prefetch unit that supports speculative branches.

The peripheral complement of the LPC176x/5x includes up to 512 kB of flash memory, up to 64 kB of data memory, Ethernet MAC, a USB interface that can be configured as either Host, Device, or OTG, 8 channel general purpose DMA controller, 4 UARTs, 2 CAN channels, 2 SSP controllers, SPI interface, 3 I^2C interfaces, 2-input plus 2-output I^2S interface, 8 channel 12-bit ADC, 10-bit DAC, motor control PWM, Quadrature Encoder interface, 4 general purpose timers, 6-output general purpose PWM, ultra-low power RTC with separate battery supply, and up to 70 general purpose I/O pins.

1.2 Features

Refer to Section 1.4.1 for details of features on specific part numbers.

- ARM Cortex-M3 processor, running at frequencies of up to 120 MHz on high speed versions (LPC1769 and LPC1759), up to 100 MHz on other versions. A Memory Protection Unit (MPU) supporting eight regions is included.
- ARM Cortex-M3 built-in Nested Vectored Interrupt Controller (NVIC).
- Up to 512 kB on-chip flash program memory with In-System Programming (ISP) and In-Application Programming (IAP) capabilities. The combination of an enhanced flash memory accelerator and location of the flash memory on the CPU local code/data bus provides high code performance from flash.
- Up to 64 kB on-chip SRAM includes:
  - Up to 32 kB of SRAM on the CPU with local code/data bus for high-performance CPU access.
  - Up to two 16 kB SRAM blocks with separate access paths for higher throughput. These SRAM blocks may be used for Ethernet, USB, and DMA memory, as well as for general purpose instruction and data storage.
- Eight channel General Purpose DMA controller (GPDMA) on the AHB multilayer matrix that can be used with the SSP, I^2S, UART, the Analog-to-Digital and Digital-to-Analog converter peripherals, timer match signals, GPIO, and for memory-to-memory transfers.
- Multilayer AHB matrix interconnect provides a separate bus for each AHB master. AHB masters include the CPU, General Purpose DMA controller, Ethernet MAC, and the USB interface. This interconnect provides communication with no arbitration delays unless two masters attempt to access the same slave at the same time.
- Split APB bus allows for higher throughput with fewer stalls between the CPU and DMA. A single level of write buffering allows the CPU to continue without waiting for completion of APB writes if the APB was not already busy.
- Serial interfaces:
  - Ethernet MAC with RMII interface and dedicated DMA controller.
  - USB 2.0 full-speed controller that can be configured for either device, Host, or OTG operation with an on-chip PHY for device and Host functions and a dedicated DMA controller.
  - Four UARTs with fractional baud rate generation, internal FIFO, IrDA, and DMA support. One UART has modem control I/O and RS-485/EIA-485 support.
  - Two-channel CAN controller.
  - Two SSP controllers with FIFO and multi-protocol capabilities. The SSP interfaces can be used with the GPDMA controller.
  - SPI controller with synchronous, serial, full duplex communication and programmable data length. SPI is included as a legacy peripheral and can be used instead of SSP0.
  - Three enhanced I^2C-bus interfaces, one with an open-drain output supporting the full I^2C specification and Fast mode plus with data rates of 1Mbit/s, two with standard port pins. Enhancements include multiple address recognition and monitor mode.

  - I^2S (Inter-IC Sound) interface for digital audio input or output, with fractional rate control. The I^2S interface can be used with the GPDMA. The I^2S interface supports 3-wire data transmit and receive or 4-wire combined transmit and receive connections, as well as master clock output.

- Other peripherals:
  - 70 (100 pin package) or 52 (80-pin package) General Purpose I/O (GPIO) pins with configurable pull-up/down resistors, open drain mode, and repeater mode. All GPIOs are located on an AHB bus for fast access, and support Cortex-M3 bit-banding. GPIOs can be accessed by the General Purpose DMA Controller. Any pin of ports 0 and 2 can be used to generate an interrupt.
  - 12-bit Analog-to-Digital Converter (ADC) with input multiplexing among eight pins, conversion rates up to 200 kHz, and multiple result registers. The 12-bit ADC can be used with the GPDMA controller.
  - 10-bit Digital-to-Analog Converter (DAC) with dedicated conversion timer and DMA support.
  - Four general purpose timers/counters, with a total of eight capture inputs and ten compare outputs. Each timer block has an external count input. Specific timer events can be selected to generate DMA requests.
  - One motor control PWM with support for three-phase motor control.
  - Quadrature encoder interface that can monitor one external quadrature encoder.
  - One standard PWM/timer block with external count input.
  - Real-Time Clock (RTC) with a separate power domain. The RTC is clocked by a dedicated RTC oscillator. The RTC block includes 20 bytes of battery-powered backup registers, allowing system status to be stored when the rest of the chip is powered off. Battery power can be supplied from a standard 3 V Lithium button cell. The RTC will continue working when the battery voltage drops to as low as 2.1 V. An RTC interrupt can wake up the CPU from any reduced power mode.
  - Watchdog Timer (WDT). The WDT can be clocked from the internal RC oscillator, the RTC oscillator, or the APB clock.
  - Cortex-M3 system tick timer, including an external clock input option.
  - Repetitive interrupt timer provides programmable and repeating timed interrupts.
- Standard JTAG test/debug interface as well as Serial Wire Debug and Serial Wire Trace Port options.
- Emulation trace module supports real-time trace.
- Four reduced power modes: Sleep, Deep-sleep, Power-down, and Deep power-down.
- Single 3.3 V power supply (2.4 V to 3.6 V). Temperature range of -40 °C to 85 °C.
- Four external interrupt inputs configurable as edge/level sensitive. All pins on PORT0 and PORT2 can be used as edge sensitive interrupt sources.
- Non-maskable Interrupt (NMI) input.
- Clock output function that can reflect the main oscillator clock, IRC clock, RTC clock, CPU clock, or the USB clock.
- The Wake-up Interrupt Controller (WIC) allows the CPU to automatically wake up from any priority interrupt that can occur while the clocks are stopped in deep sleep, Power-down, and Deep power-down modes.

- Processor wake-up from Power-down mode via any interrupt able to operate during Power-down mode (includes external interrupts, RTC interrupt, USB activity, Ethernet wake-up interrupt, CAN bus activity, PORT0/2 pin interrupt, and NMI).
- Each peripheral has its own clock divider for further power savings.
- Brownout detect with separate threshold for interrupt and forced reset.
- On-chip Power-On Reset (POR).
- On-chip crystal oscillator with an operating range of 1 MHz to 25 MHz.
- 4 MHz internal RC oscillator trimmed to 1% accuracy that can optionally be used as a system clock.
- An on-chip PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. May be run from the main oscillator, the internal RC oscillator, or the RTC oscillator.
- A second, dedicated PLL may be used for the USB interface in order to allow added flexibility for the Main PLL settings.
- Versatile pin function selection feature allows many possibilities for using on-chip peripheral functions.
- Available as LQFP100 (14 mm × 14 mm × 1.4 mm), TFBGA100[1] (9 mm × 9 mm × 0.7 mm), WLCSP100 (5.074 × 5.074 × 0.6 mm) package, and 80-pin LQFP (12 x 12 x 1.4 mm) packages

1.3 Applications

- eMetering
- Lighting
- Industrial networking
- Alarm systems
- White goods
- Motor control

1. LPC1768/65 only.

1.4 Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
LPC1769FBD100	LQFP100	plastic low profile quad flat package; 100 leads; body 14 × 14 × 1.4 mm	SOT407-1
LPC1768FBD100			
LPC1767FBD100			
LPC1766FBD100			
LPC1765FBD100			
LPC1764FBD100			
LPC1763FBD100			
LPC1768FET100	TFBGA100	plastic thin fine-pitch ball grid array package; 100 balls; body 9 x 9 x 0.7 mm	SOT926-1
LPC1768UK	WLCSP100	wafer level chip-scale package; 100 balls; 5.074 × 5.074 × 0.6 mm	-
LPC1759FBD80	LQFP80	plastic low profile quad flat package; 80 leads; body 12 × 12 × 1.4 mm	SOT315-1
LPC1758FBD80			
LPC1756FBD80			
LPC1754FBD80			
LPC1752FBD80			
LPC1751FBD80			

1.4.1 Part options summary

Table 2. Ordering options for LPC176x/5x parts

Type number	Max. CPU speed	Flash	Total SRAM	Ethernet	USB	CAN	I^2S	DAC	Package
LPC1769FBD100	120 MHz	512 kB	64 kB	yes	Device/Host/OTG	2	yes	yes	100 pin
LPC1768FBD100	100 MHz	512 kB	64 kB	yes	Device/Host/OTG	2	yes	yes	100 pin
LPC1768FET100	100 MHz	512 kB	64 kB	yes	Device/Host/OTG	2	yes	yes	100 pin
LPC1768UK	100 MHz	512 kB	64 kB	yes	Device/Host/OTG	2	yes	yes	100 pin
LPC1767FBD100	100 MHz	512 kB	64 kB	yes	no	no	yes	yes	100 pin
LPC1766FBD100	100 MHz	256 kB	64 kB	yes	Device/Host/OTG	2	yes	yes	100 pin
LPC1765FBD100	100 MHz	256 kB	64 kB	no	Device/Host/OTG	2	yes	yes	100 pin
LPC1764FBD100	100 MHz	128 kB	32 kB	yes	Device	2	no	no	100 pin
LPC1763FBD100	100 MHz	256 kB	64 kB	no	no	no	yes	yes	100 pin
LPC1759FBD80	120 MHz	512 kB	64 kB	no	Device/Host/OTG	2	yes	yes	80 pin
LPC1758FBD80	100 MHz	512 kB	64 kB	yes	Device/Host/OTG	2	yes	yes	80 pin
LPC1756FBD80	100 MHz	256 kB	32 kB	no	Device/Host/OTG	2	yes	yes	80 pin
LPC1754FBD80	100 MHz	128 kB	32 kB	no	Device/Host/OTG	1	no	yes	80 pin
LPC1752FBD80	100 MHz	64 kB	16 kB	no	Device	1	no	no	80 pin
LPC1751FBD80	100 MHz	32 kB	8 kB	no	Device	1	no	no	80 pin

1.5 Simplified block diagram

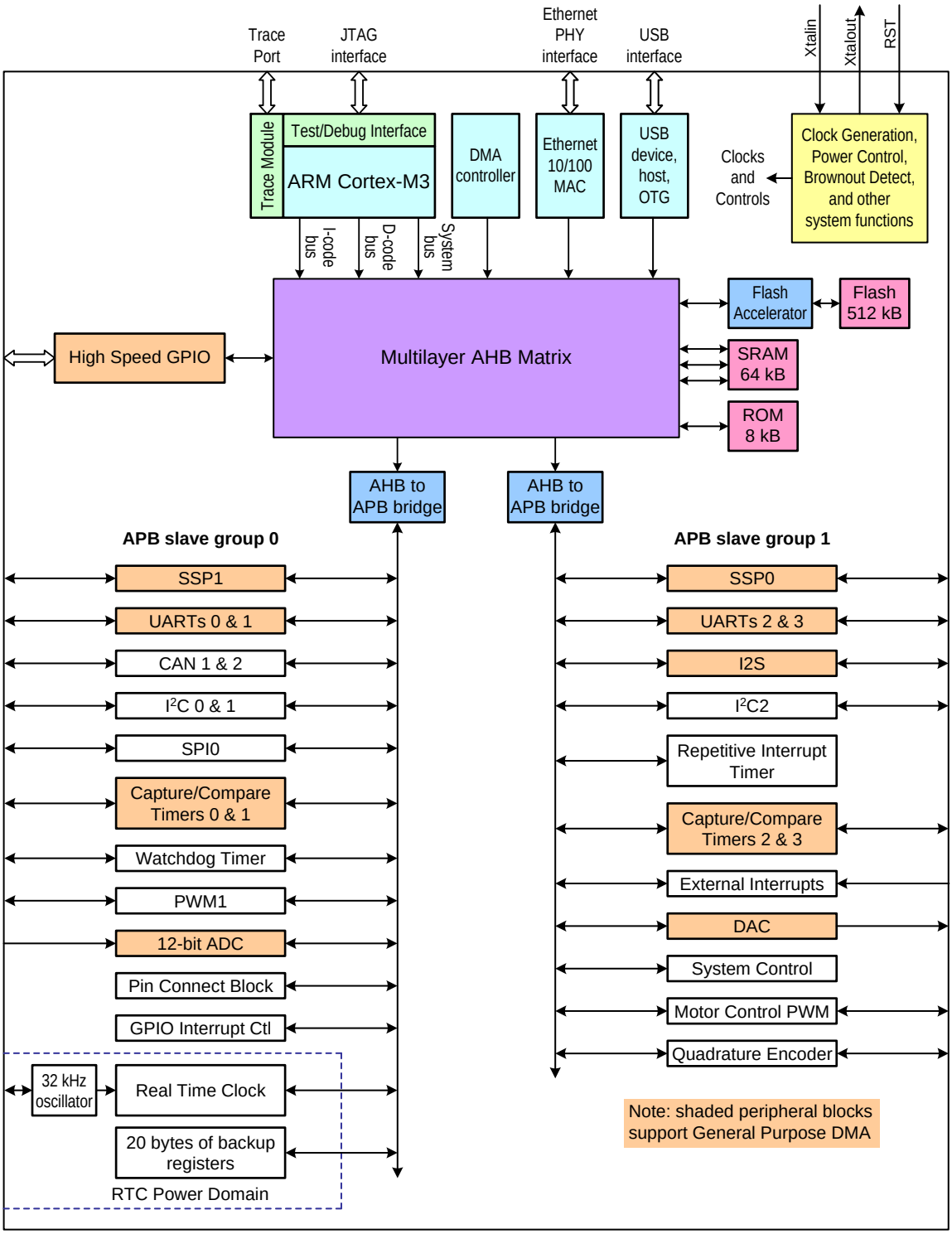

Fig 1. LPC1768 simplified block diagram

1.6 Architectural overview

The ARM Cortex-M3 includes three AHB-Lite buses, one system bus and the I-code and D-code buses which are faster and are used similarly to TCM interfaces: one bus dedicated for instruction fetch (I-code) and one bus for data access (D-code). The use of two core buses allows for simultaneous operations if concurrent operations target different devices.

The LPC176x/5x uses a multi-layer AHB matrix to connect the Cortex-M3 buses and other bus masters to peripherals in a flexible manner that optimizes performance by allowing peripherals on different slaves ports of the matrix to be accessed simultaneously by different bus masters. Details of the multilayer matrix connections are shown in Figure 2.

APB peripherals are connected to the CPU via two APB busses using separate slave ports from the multilayer AHB matrix. This allows for better performance by reducing collisions between the CPU and the DMA controller. The APB bus bridges are configured to buffer writes so that the CPU or DMA controller can write to APB devices without always waiting for APB write completion.

1.7 ARM Cortex-M3 processor

The ARM Cortex-M3 is a general purpose 32-bit microprocessor, which offers high performance and very low power consumption. The Cortex-M3 offers many new features, including a Thumb-2 instruction set, low interrupt latency, hardware divide, interruptible/continuable multiple load and store instructions, automatic state save and restore for interrupts, tightly integrated interrupt controller with Wake-up Interrupt Controller, and multiple core buses capable of simultaneous accesses.

Pipeline techniques are employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

The ARM Cortex-M3 processor is described in detail in the Cortex-M3 User Guide that is appended to this manual.

1.7.1 Cortex-M3 Configuration Options

The LPC176x/5x uses the r2p0 version of the Cortex-M3 CPU, which includes a number of configurable options, as noted below.

System options:

- The Nested Vectored Interrupt Controller (NVIC) is included. The NVIC includes the SYSTICK timer.
- The Wake-up Interrupt Controller (WIC) is included. The WIC allows more powerful options for waking up the CPU from reduced power modes.
- A Memory Protection Unit (MPU) is included.
- A ROM Table in included. The ROM Table provides addresses of debug components to external debug systems.

Debug related options:

- A JTAG debug interface is included.
- Serial Wire Debug is included. Serial Wire Debug allows debug operations using only 2 wires, simple trace functions can be added with a third wire.
- The Embedded Trace Macrocell (ETM) is included. The ETM provides instruction trace capabilities.
- The Data Watchpoint and Trace (DWT) unit is included. The DWT allows data address or data value matches to be trace information or trigger other events. The DWT includes 4 comparators and counters for certain internal events.
- An Instrumentation Trace Macrocell (ITM) is included. Software can write to the ITM in order to send messages to the trace port.
- The Trace Port Interface Unit (TPIU) is included. The TPIU encodes and provides trace information to the outside world. This can be on the Serial Wire Viewer pin or the 4-bit parallel trace port.
- A Flash Patch and Breakpoint (FPB) is included. The FPB can generate hardware breakpoints and remap specific addresses in code space to SRAM as a temporary method of altering non-volatile code. The FPB include 2 literal comparators and 6 instruction comparators.

1.8 On-chip flash memory system

The LPC176x/5x contains up to 512 kB of on-chip flash memory. A flash memory accelerator maximizes performance for use with the two fast AHB-Lite buses. This memory may be used for both code and data storage. Programming of the flash memory may be accomplished in several ways. It may be programmed In System via the serial port. The application program may also erase and/or program the flash while the application is running, allowing a great degree of flexibility for data storage field firmware upgrades, etc.

1.9 On-chip Static RAM

The LPC176x/5x contains up to 64 kB of on-chip static RAM memory. Up to 32 kB of SRAM, accessible by the CPU and all three DMA controllers are on a higher-speed bus. Devices containing more than 32 kB SRAM have two additional 16 kB SRAM blocks, each situated on separate slave ports on the AHB multilayer matrix.

This architecture allows the possibility for CPU and DMA accesses to be separated in such a way that there are few or no delays for the bus masters.

1.10 Block diagram

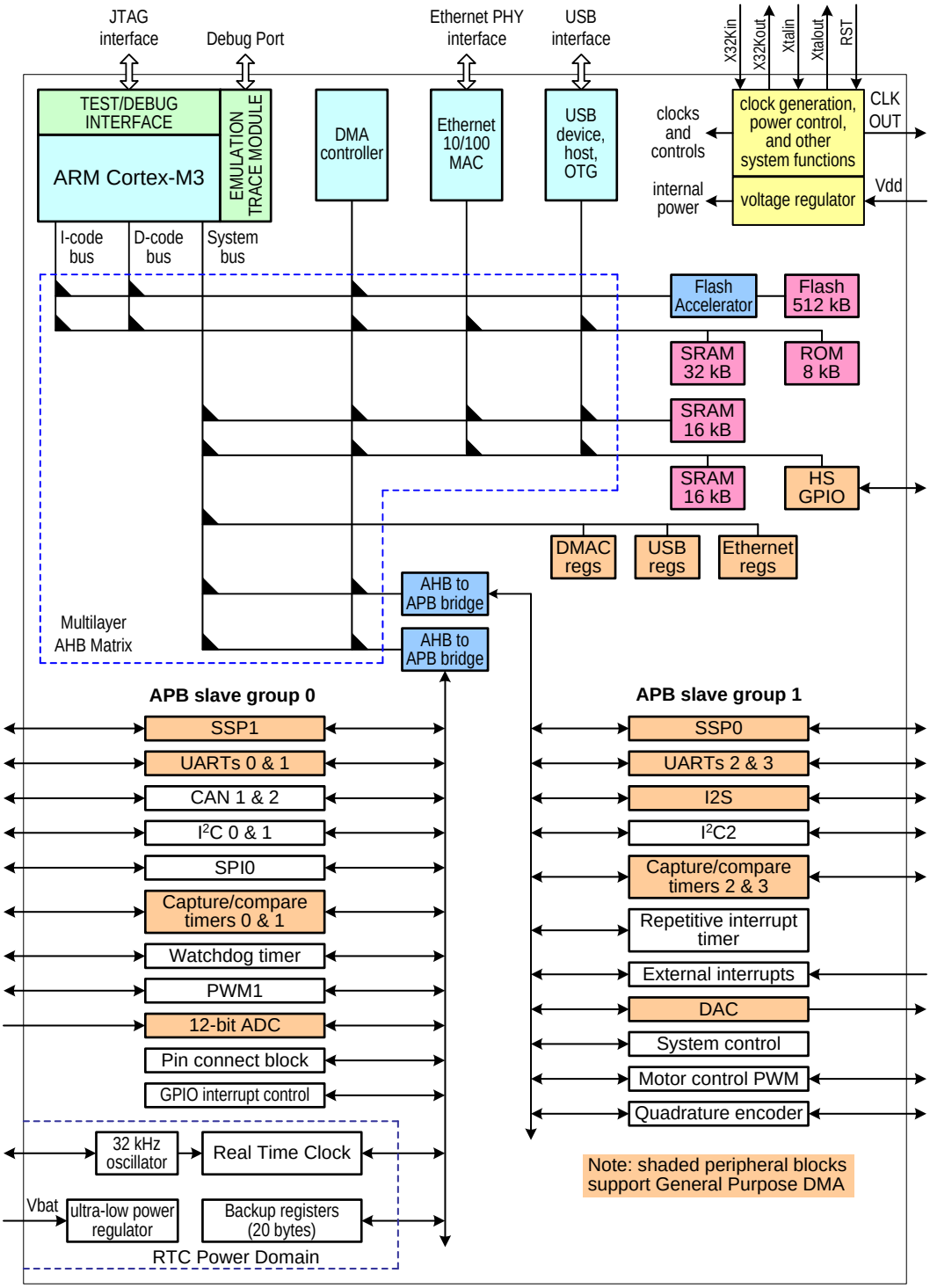

Fig 2. LPC1768 block diagram, CPU and buses

2.1 Memory map and peripheral addressing

The ARM Cortex-M3 processor has a single 4 GB address space. The following table shows how this space is used on the LPC176x/5x.

Table 3. LPC176x/5x memory usage and details

Address range	General Use	Address range details and description	
0x0000 0000 to 0x1FFF FFFF	On-chip non-volatile memory	0x0000 0000 - 0x0007 FFFF	For devices with 512 kB of flash memory.
		0x0000 0000 - 0x0003 FFFF	For devices with 256 kB of flash memory.
		0x0000 0000 - 0x0001 FFFF	For devices with 128 kB of flash memory.
		0x0000 0000 - 0x0000 FFFF	For devices with 64 kB of flash memory.
		0x0000 0000 - 0x0000 7FFF	For devices with 32 kB of flash memory.
	On-chip SRAM	0x1000 0000 - 0x1000 7FFF	For devices with 32 kB of local SRAM.
		0x1000 0000 - 0x1000 3FFF	For devices with 16 kB of local SRAM.
		0x1000 0000 - 0x1000 1FFF	For devices with 8 kB of local SRAM.
	Boot ROM	0x1FFF 0000 - 0x1FFF 1FFF	8 kB Boot ROM with flash services.
0x2000 0000 to 0x3FFF FFFF	On-chip SRAM (typically used for peripheral data)	0x2007 C000 - 0x2007 FFFF	AHB SRAM - bank 0 (16 kB), present on devices with 32 kB or 64 kB of total SRAM.
		0x2008 0000 - 0x2008 3FFF	AHB SRAM - bank 1 (16 kB), present on devices with 64 kB of total SRAM.
	GPIO	0x2009 C000 - 0x2009 FFFF	GPIO.
0x4000 0000 to 0x5FFF FFFF	APB Peripherals	0x4000 0000 - 0x4007 FFFF	APB0 Peripherals, up to 32 peripheral blocks, 16 kB each.
		0x4008 0000 - 0x400F FFFF	APB1 Peripherals, up to 32 peripheral blocks, 16 kB each.
	AHB peripherals	0x5000 0000 - 0x501F FFFF	DMA Controller, Ethernet interface, and USB interface.
0xE000 0000 to 0xE00F FFFF	Cortex-M3 Private Peripheral Bus	0xE000 0000 - 0xE00F FFFF	Cortex-M3 related functions, includes the NVIC and System Tick Timer.

2.2 Memory maps

The LPC176x/5x incorporates several distinct memory regions, shown in the following figures. Figure 3 shows the overall map of the entire address space from the user program viewpoint following reset. The interrupt vector area supports address remapping, which is described later in this section.

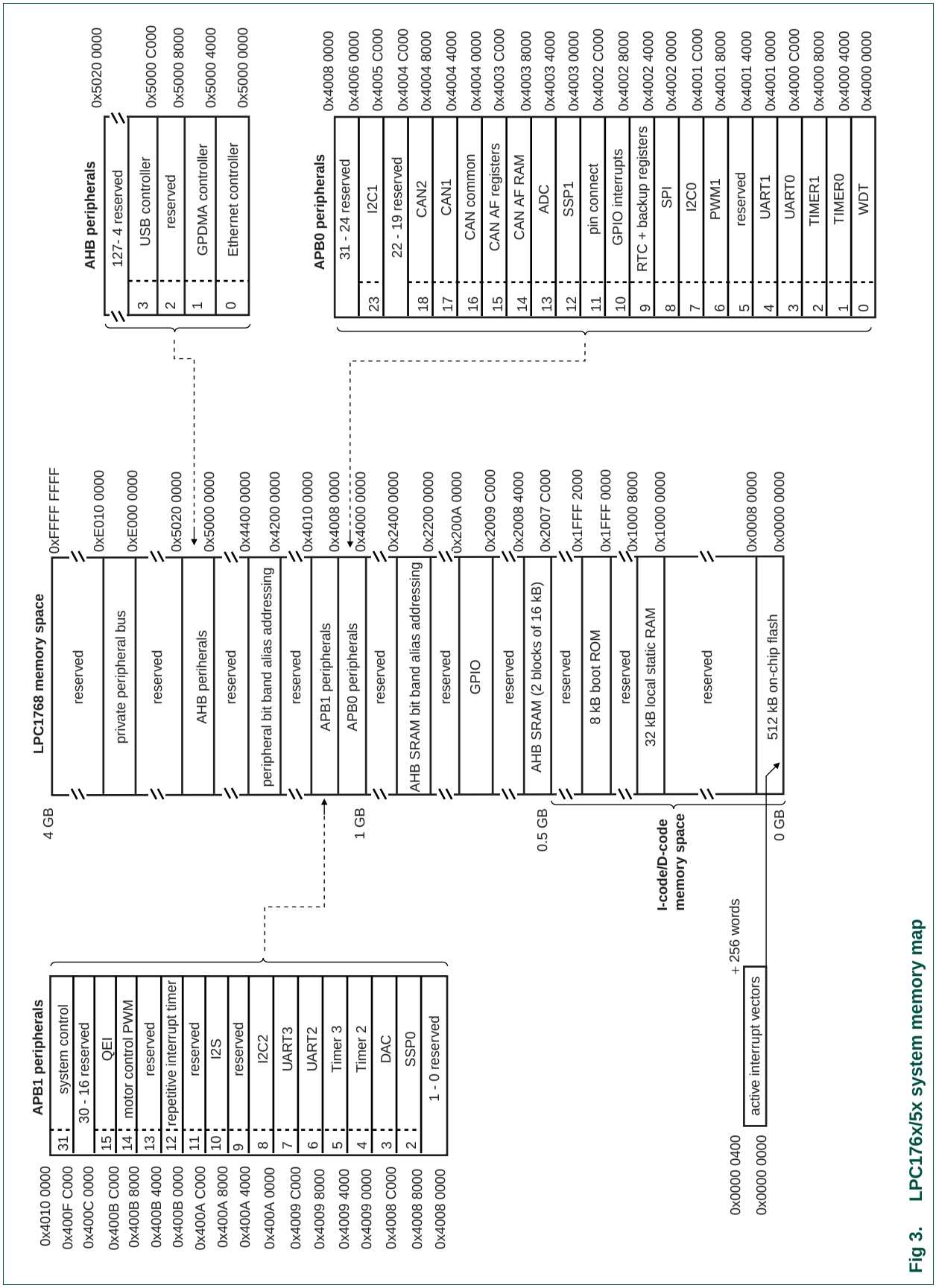

Fig 3. LPC176x/5x system memory map

Figure 3 and Table 4 show different views of the peripheral address space. The AHB peripheral area is 2 megabyte in size, and is divided to allow for up to 128 peripherals. The APB peripheral area is 1 megabyte in size and is divided to allow for up to 64 peripherals. Each peripheral of either type is allocated 16 kilobytes of space. This allows simplifying the address decoding for each peripheral.

All peripheral register addresses are word aligned (to 32-bit boundaries) regardless of their size. This eliminates the need for byte lane mapping hardware that would be required to allow byte (8-bit) or half-word (16-bit) accesses to occur at smaller boundaries. An implication of this is that word and half-word registers must be accessed all at once. For example, it is not possible to read or write the upper byte of a word register separately.

2.3 APB peripheral addresses

The following table shows the APB0/1 address maps. No APB peripheral uses all of the 16 kB space allocated to it. Typically each device's registers are "aliased" or repeated at multiple locations within each 16 kB range.

Table 4. APB0 peripherals and base addresses

APB0 peripheral	Base address	Peripheral name
0	0x4000 0000	Watchdog Timer
1	0x4000 4000	Timer 0
2	0x4000 8000	Timer 1
3	0x4000 C000	UART0
4	0x4001 0000	UART1
5	0x4001 4000	reserved
6	0x4001 8000	PWM1
7	0x4001 C000	I^2C0
8	0x4002 0000	SPI
9	0x4002 4000	RTC
10	0x4002 8000	GPIO interrupts
11	0x4002 C000	Pin Connect Block
12	0x4003 0000	SSP1
13	0x4003 4000	ADC
14	0x4003 8000	CAN Acceptance Filter RAM
15	0x4003 C000	CAN Acceptance Filter Registers
16	0x4004 0000	CAN Common Registers
17	0x4004 4000	CAN Controller 1
18	0x4004 8000	CAN Controller 2
19 to 22	0x4004 C000 to 0x4005 8000	reserved
23	0x4005 C000	I^2C1
24 to 31	0x4006 0000 to 0x4007 C000	reserved

Table 5. APB1 peripherals and base addresses

APB1 peripheral	Base address	Peripheral name
0	0x4008 0000	reserved
1	0x4008 4000	reserved
2	0x4008 8000	SSP0
3	0x4008 C000	DAC
4	0x4009 0000	Timer 2
5	0x4009 4000	Timer 3
6	0x4009 8000	UART2
7	0x4009 C000	UART3
8	0x400A 0000	I^2C2
9	0x400A 4000	reserved
10	0x400A 8000	I^2S
11	0x400A C000	reserved
12	0x400B 0000	Repetitive interrupt timer
13	0x400B 4000	reserved
14	0x400B 8000	Motor control PWM
15	0x400B C000	Quadrature Encoder Interface
16 to 30	0x400C 0000 to 0x400F 8000	reserved
31	0x400F C000	System control

2.4 Memory re-mapping

The Cortex-M3 incorporates a mechanism that allows remapping the interrupt vector table to alternate locations in the memory map. This is controlled via the Vector Table Offset Register contained in the Cortex-M3. Refer to Section 6.4 and Section 34.4.3.5 of the Cortex-M3 User Guide appended to this manual for details of the Vector Table Offset feature.

Boot ROM re-mapping

Following a hardware reset, the Boot ROM is temporarily mapped to address 0. This is normally transparent to the user. However, if execution is halted immediately after reset by a debugger, it should correct the mapping for the user. See Section 33.6.

2.5 AHB arbitration

The Multilayer AHB Matrix arbitrates between several masters. By default, the Cortex-M3 D-code bus has the highest priority, followed by the I-Code bus. All other masters share a lower priority.

2.6 Bus fault exceptions

The LPC176x/5x generates Bus Fault exception if an access is attempted for an address that is in a reserved or unassigned address region. The regions are areas of the memory map that are not implemented for a specific derivative. These include all spaces marked “reserved” in Figure 3.

For these areas, both attempted data access and instruction fetch generate an exception. In addition, a Bus Fault exception is generated for any instruction fetch that maps to an AHB or APB peripheral address.

Within the address space of an existing APB peripheral, an exception is not generated in response to an access to an undefined address. Address decoding within each peripheral is limited to that needed to distinguish defined registers within the peripheral itself. For example, an access to address 0x4000 D000 (an undefined address within the UART0 space) may result in an access to the register defined at address 0x4000 C000. Details of such address aliasing within a peripheral space are not defined in the LPC176x/5x documentation and are not a supported feature.

If software executes a write directly to the flash memory, the flash accelerator will generate a Bus Fault exception. Flash programming must be accomplished by using the specified flash programming interface provided by the Boot Code.

Note that the Cortex-M3 core stores the exception flag along with the associated instruction in the pipeline and processes the exception only if an attempt is made to execute the instruction fetched from the disallowed address. This prevents accidental aborts that could be caused by prefetches that occur when code is executed very near a memory boundary.

3.1 Introduction

The system control block includes several system features and control registers for a number of functions that are not related to specific peripheral devices. These include:

- Reset
- Brown-Out Detection
- External Interrupt Inputs
- Miscellaneous System Controls and Status

Each type of function has its own register(s) if any are required and unneeded bits are defined as reserved in order to allow future expansion. Unrelated functions never share the same register addresses

3.2 Pin description

Table 6 shows pins that are associated with System Control block functions.

Table 6. Pin summary

Pin name	Pin direction	Pin description
EINT0	Input	**External Interrupt Input 0** - An active low/high level or falling/rising edge general purpose interrupt input. This pin may be used to wake up the processor from Sleep, Deep-sleep, or Power-down modes.
EINT1	Input	**External Interrupt Input 1** - See the EINT0 description above.
EINT2	Input	**External Interrupt Input 2** - See the EINT0 description above.
EINT3	Input	**External Interrupt Input 3** - See the EINT0 description above.
$\overline{\text{RESET}}$	Input	**External Reset input** - A LOW on this pin resets the chip, causing I/O ports and peripherals to take on their default states, and the processor to begin execution at address 0x0000 0000.

3.3 Register description

All registers, regardless of size, are on word address boundaries. Details of the registers appear in the description of each function.

Table 7. Summary of system control registers

Name	Description	Access	Reset value	Address
External Interrupts				
EXTINT	External Interrupt Flag Register	R/W	0	0x400F C140
EXTMODE	External Interrupt Mode register	R/W	0	0x400F C148
EXTPOLAR	External Interrupt Polarity Register	R/W	0	0x400F C14C
Reset				
RSID	Reset Source Identification Register	R/W	see Table 8	0x400F C180
Syscon Miscellaneous Registers				
SCS	System Control and Status	R/W	0	0x400F C1A0

3.4 Reset

Reset has 4 sources on the LPC176x/5x: the $\overline{\text{RESET}}$ pin, Watchdog Reset, Power On Reset (POR), and Brown Out Detect (BOD).

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 wake-up timer (see description in Section 4.9 “Wake-up timer” in this chapter), causing reset to remain asserted until the external Reset is de-asserted, the oscillator is running, a fixed number of clocks have passed, and the flash controller has completed its initialization. The reset logic is shown in the following block diagram (see Figure 4).

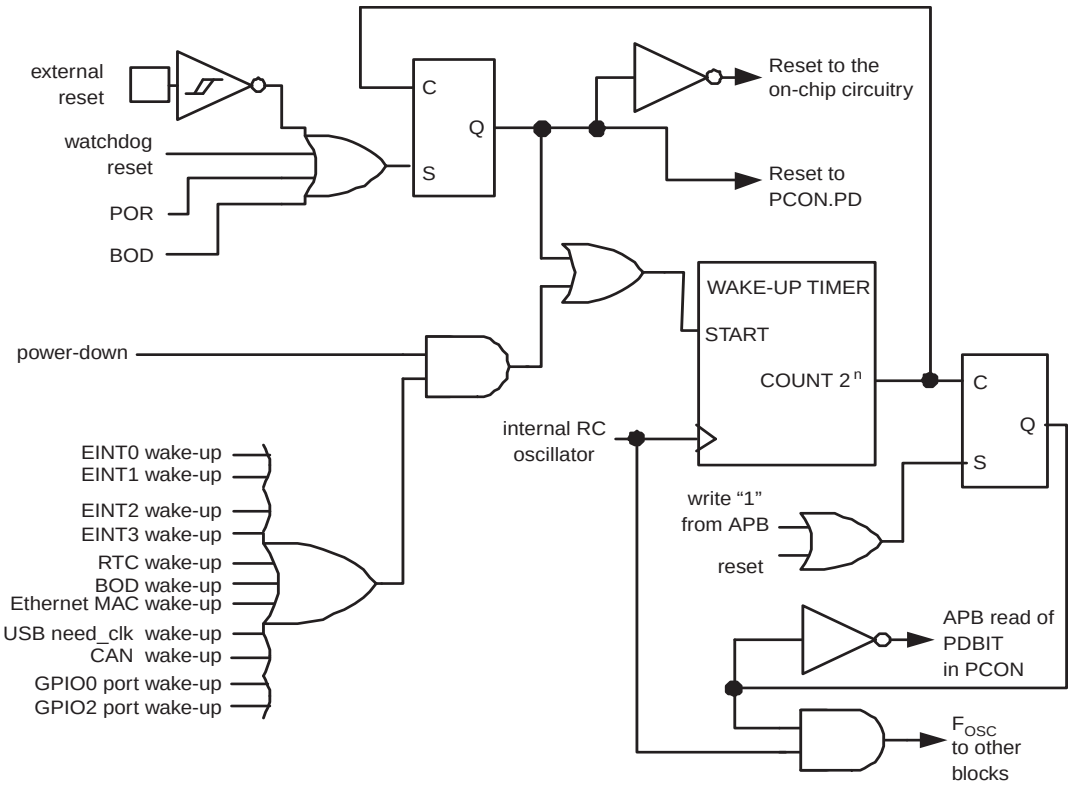

Fig 4. Reset block diagram including the wake-up timer

On the assertion of a reset source external to the Cortex-M3 CPU (POR, BOD reset, External reset, and Watchdog reset), the IRC starts up. After the IRC-start-up time (maximum of 60 μs on power-up) and after the IRC provides a stable clock output, the reset signal is latched and synchronized on the IRC clock. Then the following two sequences start simultaneously:

1. The 2-bit IRC wake-up timer starts counting when the synchronized reset is de-asserted. The boot code in the ROM starts when the 2-bit IRC wake-up timer times out. The boot code performs the boot tasks and may jump to the flash. If the flash is not ready to access, the Flash Accelerator will insert wait cycles until the flash is ready.
2. The flash wake-up timer (9-bit) starts counting when the synchronized reset is de-asserted. The flash wake-up timer generates the 100 μs flash start-up time. Once it times out, the flash initialization sequence is started, which takes about 250 cycles. When it's done, the Flash Accelerator will be granted access to the flash.

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.

Figure 5 shows an example of the relationship between the $\overline{\text{RESET}}$, the IRC, and the processor status when the LPC176x/5x starts up after reset. See Section 4.3.2 "Main oscillator" for start-up of the main oscillator if selected by the user code.

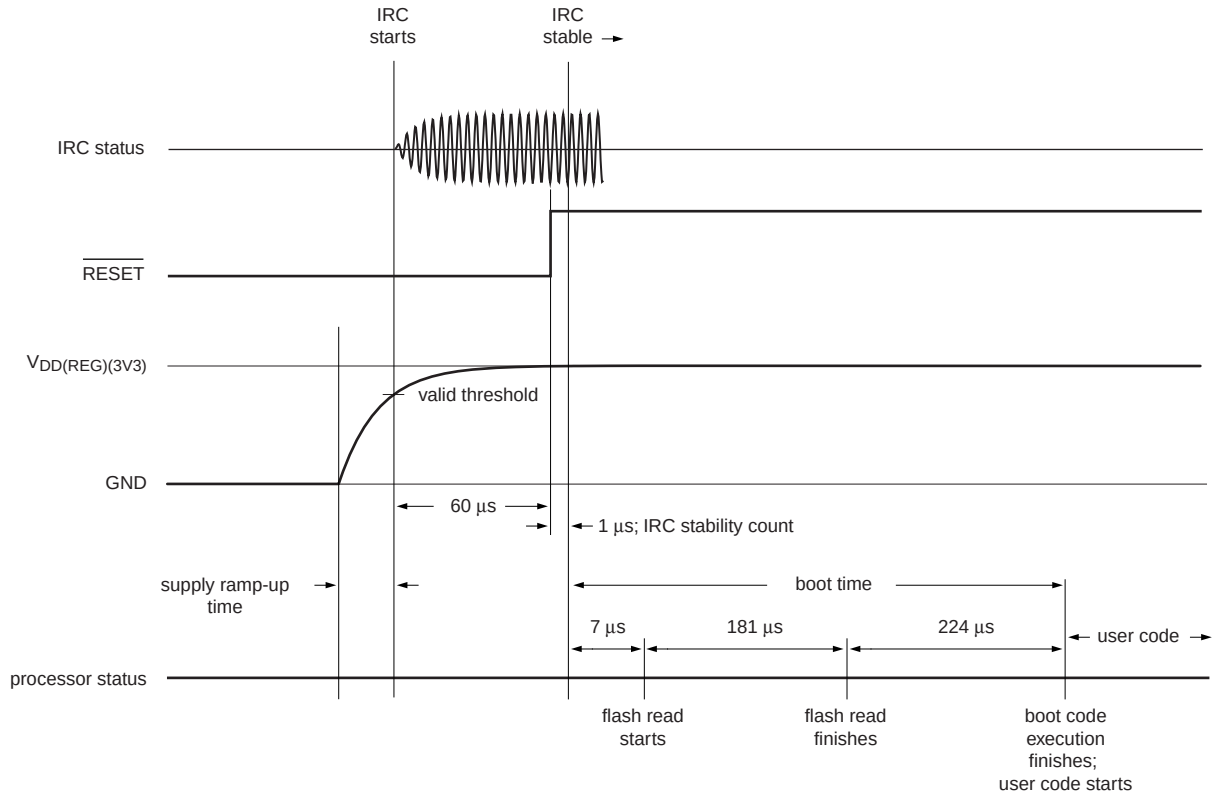

Fig 5. Example of start-up after reset

3.4.1 Reset Source Identification Register (RSID - 0x400F C180)

This register contains one bit for each source of Reset. Writing a 1 to any of these bits clears the corresponding read-side bit to 0. The interactions among the four sources are described below.

Table 8. Reset Source Identification register (RSID - address 0x400F C180) bit description

Bit	Symbol	Description	Reset value
0	POR	Assertion of the POR signal sets this bit, and clears all of the other bits in this register. But if another Reset signal (e.g., External Reset) remains asserted after the POR signal is negated, then its bit is set. This bit is not affected by any of the other sources of Reset.	See text
1	EXTR	Assertion of the $\overline{\text{RESET}}$ signal sets this bit. This bit is cleared only by software or POR.	See text
2	WDTR	This bit is set when the Watchdog Timer times out and the WDTRESET bit in the Watchdog Mode Register is 1. This bit is cleared only by software or POR. This bit is also cleared by an external reset.	See text
3	BODR	This bit is set when the $V_{DD(REG)(3V3)}$ voltage reaches a level below the BOD reset trip level (typically 1.85 V under nominal room temperature conditions). If the $V_{DD(REG)(3V3)}$ voltage dips from the normal operating range to below the BOD reset trip level and recovers, the BODR bit will be set to 1. If the $V_{DD(REG)(3V3)}$ voltage dips from the normal operating range to below the BOD reset trip level and continues to decline to the level at which POR is asserted (nominally 1 V), the BODR bit is cleared. If the $V_{DD(REG)(3V3)}$ voltage rises continuously from below 1 V to a level above the BOD reset trip level, the BODR will be set to 1. This bit is cleared only by software or POR. **Note:** Only in the case where a reset occurs and the POR = 0, the BODR bit indicates if the $V_{DD(REG)(3V3)}$ voltage was below the BOD reset trip level or not.	See text
4	SYSRESET	This bit is set if the processor has been reset due to a system reset request. Setting the SYSRESETREQ bit in the Cortex-M3 AIRCR register causes a chip reset in the LPC178x/177x. This bit is cleared only by software or POR.	See text
5	LOCKUP	This bit is set if the processor has been reset due to a “lockup”. The lockup state causes a chip reset in the LPC178x/177x. This bit is cleared only by software or POR.	See text
31:6	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

3.5 Brown-out detection

The LPC176x/5x includes a Brown-Out Detector (BOD) that provides 2-stage monitoring of the voltage on the $V_{DD(REG)(3V3)}$ pins. If this voltage falls below the BOD interrupt trip level (typically 2.2 V under nominal room temperature conditions), the BOD asserts an interrupt signal to the NVIC. This signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC in order to cause a CPU interrupt; if not, software can monitor the signal by reading the Raw Interrupt Status Register.

The second stage of low-voltage detection asserts Reset to inactivate the LPC176x/5x when the voltage on the $V_{DD(REG)(3V3)}$ pins falls below the BOD reset trip level (typically 1.85 V under nominal room temperature conditions). This Reset prevents alteration of the flash as operation of the various elements of the chip would otherwise become unreliable due to low voltage. The BOD circuit maintains this reset down below 1 V, at which point the Power-On Reset circuitry maintains the overall Reset.

Both the BOD reset interrupt level and the BOD reset trip level thresholds include some hysteresis. In normal operation, this hysteresis allows the BOD reset interrupt level detection to reliably interrupt, or a regularly-executed event loop to sense the condition.

But when Brown-Out Detection is enabled to bring the LPC176x/5x out of Power-down mode (which is itself not a guaranteed operation -- see Section 4.8.7 “Power Mode Control register (PCON - 0x400F C0C0)”), the supply voltage may recover from a transient before the wake-up timer has completed its delay. In this case, the net result of the transient BOD is that the part wakes up and continues operation after the instructions that set Power-down mode, without any interrupt occurring and with the BOD bit in the RSID being 0. Since all other wake-up conditions have latching flags (see Section 3.6.2 “External Interrupt flag register (EXTINT - 0x400F C140)” and Section 27.6.2), a wake-up of this type, without any apparent cause, can be assumed to be a Brown-Out that has gone away.

3.6 External interrupt inputs

TheLPC176x/5x includes four External Interrupt Inputs as selectable pin functions. The logic of an individual external interrupt is represented in Figure 6. In addition, external interrupts have the ability to wake up the CPU from Power-down mode. Refer to Section 4.8.8 “Wake-up from Reduced Power Modes” for details.

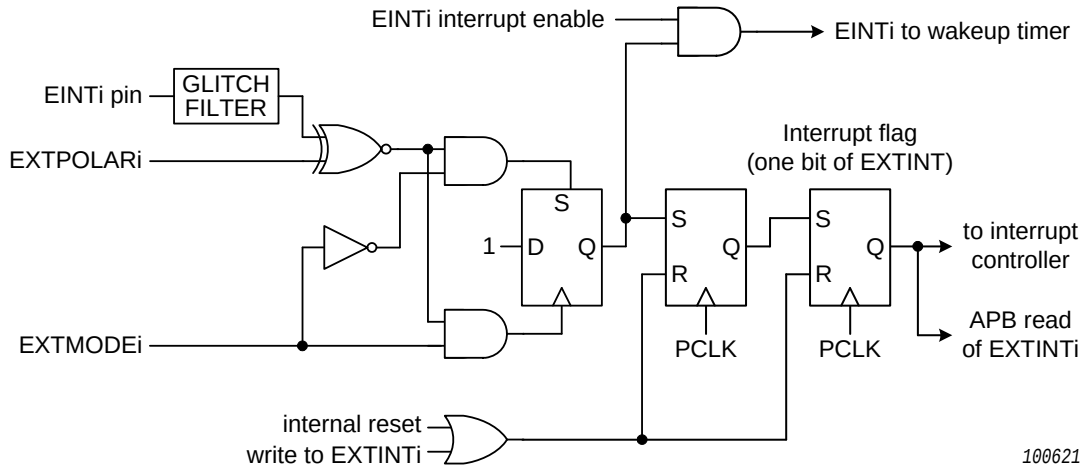

Fig 6. External interrupt logic

3.6.1 Register description

The external interrupt function has four registers associated with it. The EXTINT register contains the interrupt flags. The EXTMODE and EXTPOLAR registers specify the level and edge sensitivity parameters.

Table 9. External Interrupt registers

Name	Description	Access	Reset value[1]	Address
EXTINT	The External Interrupt Flag Register contains interrupt flags for EINT0, EINT1, EINT2 and EINT3. See Table 10.	R/W	0x00	0x400F C140
EXTMODE	The External Interrupt Mode Register controls whether each pin is edge- or level-sensitive. See Table 11.	R/W	0x00	0x400F C148
EXTPOLAR	The External Interrupt Polarity Register controls which level or edge on each pin will cause an interrupt. See Table 12.	R/W	0x00	0x400F C14C

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

3.6.2 External Interrupt flag register (EXTINT - 0x400F C140)

When a pin is selected for its external interrupt function, the level or edge on that pin (selected by its bits in the EXTPOLAR and EXTMODE registers) will set its interrupt flag in this register. This asserts the corresponding interrupt request to the NVIC, which will cause an interrupt if interrupts from the pin are enabled.

Writing ones to bits EINT0 through EINT3 in EXTINT register clears the corresponding bits. In level-sensitive mode the interrupt is cleared only when the pin is in its inactive state.

Once a bit from EINT0 to EINT3 is set and an appropriate code starts to execute (handling wake-up and/or external interrupt), this bit in EXTINT register must be cleared. Otherwise event that was just triggered by activity on the EINT pin will not be recognized in future.

Important: whenever a change of external interrupt operating mode (i.e. active level/edge) is performed (including the initialization of an external interrupt), the corresponding bit in the EXTINT register must be cleared! For details see Section 3.6.3 “External Interrupt Mode register (EXTMODE - 0x400F C148)” and Section 3.6.4 “External Interrupt Polarity register (EXTPOLAR - 0x400F C14C)”.

For example, if a system wakes up from Power-down using low level on external interrupt 0 pin, its post wake-up code must reset EINT0 bit in order to allow future entry into the Power-down mode. If EINT0 bit is left set to 1, subsequent attempt(s) to invoke Power-down mode will fail. The same goes for external interrupt handling.

More details on Power-down mode will be discussed in the following chapters.

Table 10. External Interrupt Flag register (EXTINT - address 0x400F C140) bit description

Bit	Symbol	Description	Reset value
0	EINT0	In level-sensitive mode, this bit is set if the EINT0 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT0 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
1	EINT1	In level-sensitive mode, this bit is set if the EINT1 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT1 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
2	EINT2	In level-sensitive mode, this bit is set if the EINT2 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT2 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
3	EINT3	In level-sensitive mode, this bit is set if the EINT3 function is selected for its pin, and the pin is in its active state. In edge-sensitive mode, this bit is set if the EINT3 function is selected for its pin, and the selected edge occurs on the pin. This bit is cleared by writing a one to it, except in level sensitive mode when the pin is in its active state.[1]	0
31:4	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

[1] Example: e.g. if the EINTx is selected to be low level sensitive and low level is present on corresponding pin, this bit can not be cleared; this bit can be cleared only when signal on the pin becomes high.

3.6.3 External Interrupt Mode register (EXTMODE - 0x400F C148)

The bits in this register select whether each EINT pin is level- or edge-sensitive. Only pins that are selected for the EINT function (see Section 8.5) and enabled in the appropriate NVIC register) can cause interrupts from the External Interrupt function (though of course pins selected for other functions may cause interrupts from those functions).

Note: Software should only change a bit in this register when its interrupt is disabled in the NVIC (state readable in the ISERn/ICERn registers), and should write the corresponding 1 to EXTINT before enabling (initializing) or re-enabling the interrupt. An extraneous interrupt(s) could be set by changing the mode and not having the EXTINT cleared.

Table 11. External Interrupt Mode register (EXTMODE - address 0x400F C148) bit description

Bit	Symbol	Value	Description	Reset value
0	EXTMODE0	0	Level-sensitivity is selected for $\overline{\text{EINT0}}$.	0
		1	$\overline{\text{EINT0}}$ is edge sensitive.	
1	EXTMODE1	0	Level-sensitivity is selected for $\overline{\text{EINT1}}$.	0
		1	$\overline{\text{EINT1}}$ is edge sensitive.	
2	EXTMODE2	0	Level-sensitivity is selected for $\overline{\text{EINT2}}$.	0
		1	$\overline{\text{EINT2}}$ is edge sensitive.	
3	EXTMODE3	0	Level-sensitivity is selected for $\overline{\text{EINT3}}$.	0
		1	$\overline{\text{EINT3}}$ is edge sensitive.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

3.6.4 External Interrupt Polarity register (EXTPOLAR - 0x400F C14C)

In level-sensitive mode, the bits in this register select whether the corresponding pin is high- or low-active. In edge-sensitive mode, they select whether the pin is rising- or falling-edge sensitive. Only pins that are selected for the EINT function Only pins that are selected for the EINT function (see Section 8.5) and enabled in the appropriate NVIC register) can cause interrupts from the External Interrupt function (though of course pins selected for other functions may cause interrupts from those functions).

Note: Software should only change a bit in this register when its interrupt is disabled in the NVIC (state readable in the ISERn/ICERn registers), and should write the corresponding 1 to EXTINT before enabling (initializing) or re-enabling the interrupt. An extraneous interrupt(s) could be set by changing the polarity and not having the EXTINT cleared.

Table 12. External Interrupt Polarity register (EXTPOLAR - address 0x400F C14C) bit description

Bit	Symbol	Value	Description	Reset value
0	EXTPOLAR0	0	$\overline{\text{EINT0}}$ is low-active or falling-edge sensitive (depending on EXTMODE0).	0
		1	$\overline{\text{EINT0}}$ is high-active or rising-edge sensitive (depending on EXTMODE0).	
1	EXTPOLAR1	0	$\overline{\text{EINT1}}$ is low-active or falling-edge sensitive (depending on EXTMODE1).	0
		1	$\overline{\text{EINT1}}$ is high-active or rising-edge sensitive (depending on EXTMODE1).	
2	EXTPOLAR2	0	$\overline{\text{EINT2}}$ is low-active or falling-edge sensitive (depending on EXTMODE2).	0
		1	$\overline{\text{EINT2}}$ is high-active or rising-edge sensitive (depending on EXTMODE2).	

Table 12. External Interrupt Polarity register (EXTPOLAR - address 0x400F C14C) bit description

Bit	Symbol	Value	Description	Reset value
3	EXTPOLAR3	0	$\overline{\text{EINT3}}$ is low-active or falling-edge sensitive (depending on EXTMODE3).	0
		1	$\overline{\text{EINT3}}$ is high-active or rising-edge sensitive (depending on EXTMODE3).	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

3.7 Other system controls and status flags

Some aspects of controlling LPC176x/5x operation that do not fit into peripheral or other registers are grouped here.

3.7.1 System Controls and Status register (SCS - 0x400F C1A0)

The SCS register contains several control/status bits related to the main oscillator. Since chip operation always begins using the Internal RC Oscillator, and the main oscillator may not be used at all in some applications, it will only be started by software request. This is accomplished by setting the OSCEN bit in the SCS register, as described in Table 3-13. The main oscillator provides a status flag (the OSCSTAT bit in the SCS register) so that software can determine when the oscillator is running and stable. At that point, software can control switching to the main oscillator as a clock source. Prior to starting the main oscillator, a frequency range must be selected by configuring the OSCRANGE bit in the SCS register.

Table 13. System Controls and Status register (SCS - address 0x400F C1A0) bit description

Bit	Symbol	Value	Description	Access	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
4	OSCRANGE		Main oscillator range select.	R/W	0
		0	The frequency range of the main oscillator is 1 MHz to 20 MHz.		
		1	The frequency range of the main oscillator is 15 MHz to 25 MHz.		
5	OSCEN		Main oscillator enable.	R/W	0
		0	The main oscillator is disabled.		
		1	The main oscillator is enabled, and will start up if the correct external circuitry is connected to the XTAL1 and XTAL2 pins.		
6	OSCSTAT		Main oscillator status.	RO	0
		0	The main oscillator is not ready to be used as a clock source.		
		1	The main oscillator is ready to be used as a clock source. The main oscillator must be enabled via the OSCEN bit.		
31:7	-	-	Reserved. User software should not write ones to reserved bits. The value read from a reserved bit is not defined.	-	NA

4.1 Summary of clocking and power control functions

This section describes the generation of the various clocks needed by the LPC176x/5x and options of clock source selection, as well as power control and wake-up from reduced power modes. Functions described in the following subsections include:

- Oscillators
- Clock source selection
- PLLs
- Clock dividers
- APB dividers
- Power control
- Wake-up timer
- External clock output

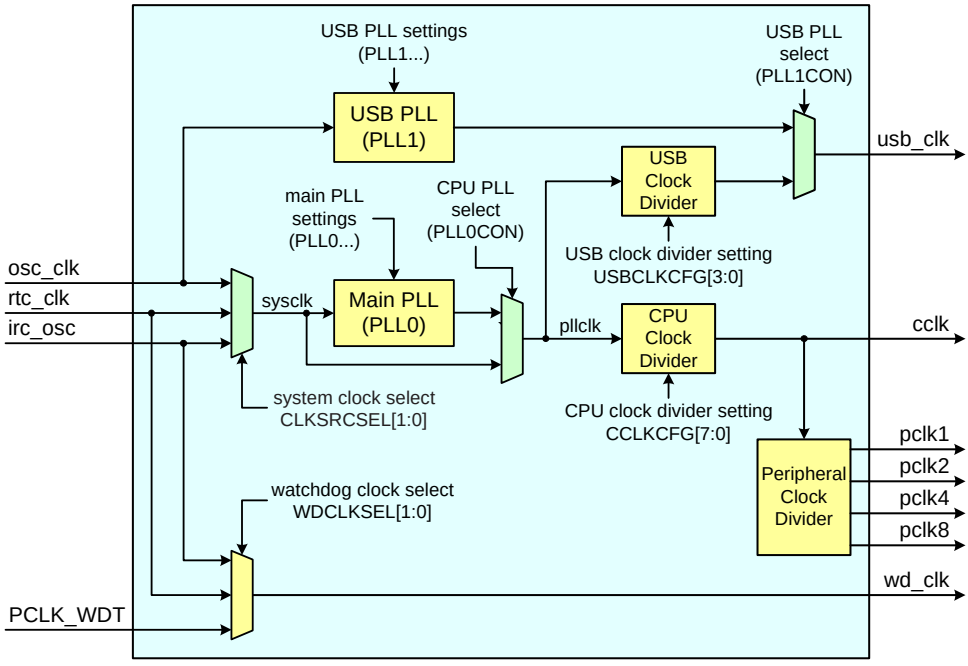

Fig 7. Clock generation for the LPC176x/5x

4.2 Register description

All registers, regardless of size, are on word address boundaries. Details of the registers appear in the description of each function.

Table 14. Summary of system control registers

Name	Description	Access	Reset value	Address
Clock source selection				
CLKSRCSEL	Clock Source Select Register	R/W	0	0x400F C10C
Phase Locked Loop (PLL0, Main PLL)				
PLL0CON	PLL0 Control Register	R/W	0	0x400F C080
PLL0CFG	PLL0 Configuration Register	R/W	0	0x400F C084
PLL0STAT	PLL0 Status Register	RO	0	0x400F C088
PLL0FEED	PLL0 Feed Register	WO	NA	0x400F C08C
Phase Locked Loop (PLL1, USB PLL)				
PLL1CON	PLL1 Control Register	R/W	0	0x400F C0A0
PLL1CFG	PLL1 Configuration Register	R/W	0	0x400F C0A4
PLL1STAT	PLL1 Status Register	RO	0	0x400F C0A8
PLL1FEED	PLL1 Feed Register	WO	NA	0x400F C0AC
Clock dividers				
CCLKCFG	CPU Clock Configuration Register	R/W	0	0x400F C104
USBCLKCFG	USB Clock Configuration Register	R/W	0	0x400F C108
PCLKSEL0	Peripheral Clock Selection register 0.	R/W	0	0x400F C1A8
PCLKSEL1	Peripheral Clock Selection register 1.	R/W	0	0x400F C1AC
Power control				
PCON	Power Control Register	R/W	0	0x400F C0C0
PCONP	Power Control for Peripherals Register	R/W	0x03BE	0x400F C0C4
Utility				
CLKOUTCFG	Clock Output Configuration Register	R/W	0	0x400F C1C8

4.3 Oscillators

The LPC176x/5x includes three independent oscillators. These are the Main Oscillator, the Internal RC Oscillator, and the RTC oscillator. Each oscillator can be used for more than one purpose as required in a particular application. This can be seen in Figure 7.

Following Reset, the LPC176x/5x will operate from the Internal RC Oscillator until switched by software. This allows systems to operate without any external crystal, and allows the boot loader code to operate at a known frequency.

4.3.1 Internal RC oscillator

The Internal RC Oscillator (IRC) may be used as the clock source for the watchdog timer, and/or as the clock that drives PLL0 and subsequently the CPU. The precision of the IRC does not allow for use of the USB interface, which requires a much more precise time base in order to comply with the USB specification. Also, the IRC should not be used with the CAN1/2 block if the CAN baud rate is higher than 100 kbit/s.The nominal IRC frequency is 4 MHz.

Upon power-up or any chip reset, the LPC176x/5x uses the IRC as the clock source. Software may later switch to one of the other available clock sources.

4.3.2 Main oscillator

The main oscillator can be used as the clock source for the CPU, with or without using PLL0. The main oscillator operates at frequencies of 1 MHz to 25 MHz. This frequency can be boosted to a higher frequency, up to the maximum CPU operating frequency, by the Main PLL (PLL0). The oscillator output is called OSC_CLK. The clock selected as the PLL0 input is PLLCLKIN and the ARM processor clock frequency is referred to as CCLK for purposes of rate equations, etc. elsewhere in this document. The frequencies of PLLCLKIN and CCLK are the same value unless the PLL0 is active and connected. Refer to Section 4.5 “PLL0 (Phase Locked Loop 0)” for details.

The on-board oscillator in the LPC176x/5x can operate in one of two modes: slave mode and oscillation mode.

In slave mode the input clock signal should be coupled by means of a capacitor of 100 pF (C_C in Figure 8, drawing a), with an amplitude between 200 mVrms and 1000 mVrms. This corresponds to a square wave signal with a signal swing of between 280 mV and 1.4 V. The XTAL2 pin in this configuration can be left unconnected.

External components and models used in oscillation mode are shown in Figure 8, drawings b and c, and in Table 15 and Table 16. Since the feedback resistance is integrated on chip, only a crystal and the capacitances C_{X1} and C_{X2} need to be connected externally in case of fundamental mode oscillation (the fundamental frequency is represented by L, C_L and R_S). Capacitance C_P in Figure 8, drawing c, represents the parallel package capacitance and should not be larger than 7 pF. Parameters F_C, C_L, R_S and C_P are supplied by the crystal manufacturer.

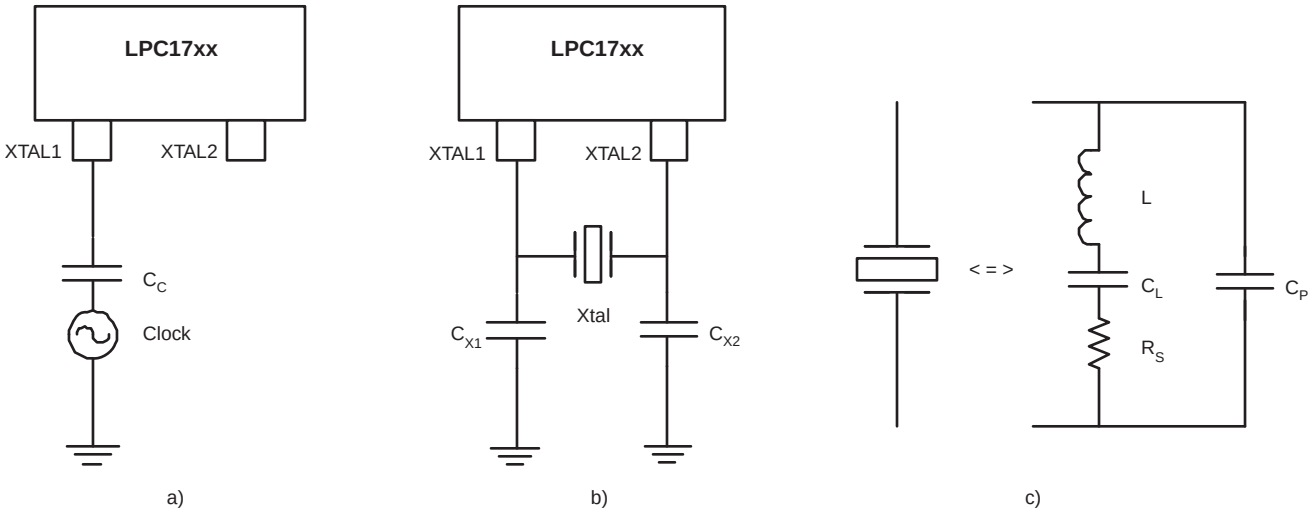

Fig 8. Oscillator modes and models: a) slave mode of operation, b) oscillation mode of operation, c) external crystal model used for $C_{X1}/_{X2}$ evaluation

Table 15. Recommended values for $C_{X1/X2}$ in oscillation mode (crystal and external components parameters) low frequency mode (OSCRANGE = 0, see Table 13)

Fundamental oscillation frequency F_{OSC}	Crystal load capacitance C_L	Maximum crystal series resistance R_S	External load capacitors C_{X1}, C_{X2}
1 MHz to 5 MHz	10 pF	< 300 Ω	18 pF, 18 pF
	20 pF	< 300 Ω	39 pF, 39 pF
	30 pF	< 300 Ω	57 pF, 57 pF
5 MHz to 10 MHz	10 pF	< 300 Ω	18 pF, 18 pF
	20 pF	< 200 Ω	39 pF, 39 pF
	30 pF	< 100 Ω	57 pF, 57 pF
10 MHz to 15 MHz	10 pF	< 160 Ω	18 pF, 18 pF
	20 pF	< 60 Ω	39 pF, 39 pF
15 MHz to 20 MHz	10 pF	< 80 Ω	18 pF, 18 pF

Table 16. Recommended values for $C_{X1/X2}$ in oscillation mode (crystal and external components parameters) high frequency mode (OSCRANGE = 1, see Table 13)

Fundamental oscillation frequency F_{OSC}	Crystal load capacitance C_L	Maximum crystal series resistance R_S	External load capacitors C_{X1}, C_{X2}
15 MHz to 20 MHz	10 pF	< 180 Ω	18 pF, 18 pF
	20 pF	< 100 Ω	39 pF, 39 pF
20 MHz to 25 MHz	10 pF	< 160 Ω	18 pF, 18 pF
	20 pF	< 80 Ω	39 pF, 39 pF

Since chip operation always begins using the Internal RC Oscillator, and the main oscillator may not be used at all in some applications, it will only be started by software request. This is accomplished by setting the OSCEN bit in the SCS register, as described in Table 13. The main oscillator provides a status flag (the OSCSTAT bit in the SCS register) so that software can determine when the oscillator is running and stable. At that

point, software can control switching to the main oscillator as a clock source. Prior to starting the main oscillator, a frequency range must be selected by configuring the OSCRANGE bit in the SCS register.

4.3.3 RTC oscillator

The RTC oscillator provides a 1 Hz clock to the RTC and a 32 kHz clock output that can be used as the clock source for PLL0 and CPU and/or the watchdog timer.

Remark: The RTC oscillator must not be used as a clock source when the PLL0 output is selected to drive the USB controller. In this case select the main oscillator as clock source for PLL0 (see also Table 17).

4.4 Clock source selection multiplexer

Several clock sources may be chosen to drive PLL0 and ultimately the CPU and on-chip peripheral devices. The clock sources available are the main oscillator, the RTC oscillator, and the Internal RC oscillator.

The clock source selection can only be changed safely when PLL0 is not connected. For a detailed description of how to change the clock source in a system using PLL0 see Section 4.5.13 “PLL0 setup sequence”.

Note the following restrictions regarding the choice of clock sources:

- Only the main oscillator must be used (via PLL0) as the clock source for the USB subsystem. The IRC or RTC oscillators do not provide the proper tolerances for this use.
- The IRC oscillator should not be used (via PLL0) as the clock source for the CAN controllers if the CAN baud rate is higher than 100 kbit/s.

4.4.1 Clock Source Select register (CLKSRCSEL - 0x400F C10C)

The CLKSRCSEL register contains the bits that select the clock source for PLL0.

Table 17. Clock Source Select register (CLKSRCSEL - address 0x400F C10C) bit description

Bit	Symbol	Value	Description	Reset value
1:0	CLKSRC		Selects the clock source for PLL0 as follows:	0
		00	Selects the Internal RC oscillator as the PLL0 clock source (default).	
		01	Selects the main oscillator as the PLL0 clock source. **Remark:** Select the main oscillator as PLL0 clock source if the PLL0 clock output is used for USB or for CAN with baudrates > 100 kBit/s.	
		10	Selects the RTC oscillator as the PLL0 clock source.	
		11	Reserved, do not use this setting.	
		Warning: Improper setting of this value, or an incorrect sequence of changing this value may result in incorrect operation of the device.		
31:2	-	0	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.5 PLL0 (Phase Locked Loop 0)

PLL0 accepts an input clock frequency in the range of 32 kHz to 50 MHz. The clock source is selected in the CLKSRCSEL register (see Section 4.4). The input frequency is multiplied up to a high frequency, then divided down to provide the actual clock used by the CPU, peripherals, and optionally the USB subsystem. Note that the USB subsystem has its own dedicated PLL (see Section 4.6). PLL0 can produce a clock up to the maximum allowed for the CPU, which is 120 MHz on high speed versions (LPC1769 and LPC1759), and 100 MHz on other versions.

4.5.1 PLL0 operation

The PLL input, in the range of 32 kHZ to 50 MHz, may initially be divided down by a value "N", which may be in the range of 1 to 256. This input division provides a greater number of possibilities in providing a wide range of output frequencies from the same input frequency.

Following the PLL input divider is the PLL multiplier. This can multiply the input divider output through the use of a Current Controlled Oscillator (CCO) by a value "M", in the range of 6 through 512, plus additional values listed in Table 21. The resulting frequency must be in the range of 275 MHz to 550 MHz. The multiplier works by dividing the CCO output by the value of M, then using a phase-frequency detector to compare the divided CCO output to the multiplier input. The error value is used to adjust the CCO frequency.

There are additional dividers at the output of PLL0 to bring the frequency down to what is needed for the CPU, peripherals, and potentially the USB subsystem. PLL0 output dividers are described in the Clock Dividers section following the PLL0 description. A block diagram of PLL0 is shown in Figure 9

PLL activation is controlled via the PLL0CON register. PLL0 multiplier and divider values are controlled by the PLL0CFG register. These two registers are protected in order to prevent accidental alteration of PLL0 parameters or deactivation of the PLL. Since all chip operations, including the Watchdog Timer, could be dependent on PLL0 if so configured (for example when it is providing the chip clock), accidental changes to the PLL0 setup values could result in unexpected or fatal behavior of the microcontroller. The protection is accomplished by a feed sequence similar to that of the Watchdog Timer. Details are provided in the description of the PLL0FEED register.

PLL0 is turned off and bypassed following a chip Reset and by entering Power-down mode. PLL0 must be configured, enabled, and connected to the system by software.

It is important that the setup procedure described in Section 4.5.13 “PLL0 setup sequence” is followed or PLL0 might not operate at all!

4.5.1.1 PLL0 and startup/boot code interaction

When there is no valid user code (determined by the checksum word) in the user flash or the ISP enable pin (P2.10) is pulled low on startup, the ISP mode will be entered and the boot code will setup the PLL with the IRC. Therefore it can not be assumed that the PLL is disabled when the user opens a debug session to debug the application code. The user startup code must follow the steps described in this chapter to disconnect the PLL.

4.5.2 PLL0 register description

PLL0 is controlled by the registers shown in Table 18. More detailed descriptions follow.

Warning: Improper setting of PLL0 values may result in incorrect operation of the device!

Table 18. PLL0 registers

Name	Description	Access	Reset value[1]	Address
PLL0CON	PLL0 Control Register. Holding register for updating PLL0 control bits. Values written to this register do not take effect until a valid PLL0 feed sequence has taken place.	R/W	0	0x400F C080
PLL0CFG	PLL0 Configuration Register. Holding register for updating PLL0 configuration values. Values written to this register do not take effect until a valid PLL0 feed sequence has taken place.	R/W	0	0x400F C084
PLL0STAT	PLL0 Status Register. Read-back register for PLL0 control and configuration information. If PLL0CON or PLL0CFG have been written to, but a PLL0 feed sequence has not yet occurred, they will not reflect the current PLL0 state. Reading this register provides the actual values controlling the PLL0, as well as the PLL0 status.	RO	0	0x400F C088
PLL0FEED	PLL0 Feed Register. This register enables loading of the PLL0 control and configuration information from the PLL0CON and PLL0CFG registers into the shadow registers that actually affect PLL0 operation.	WO	NA	0x400F C08C

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

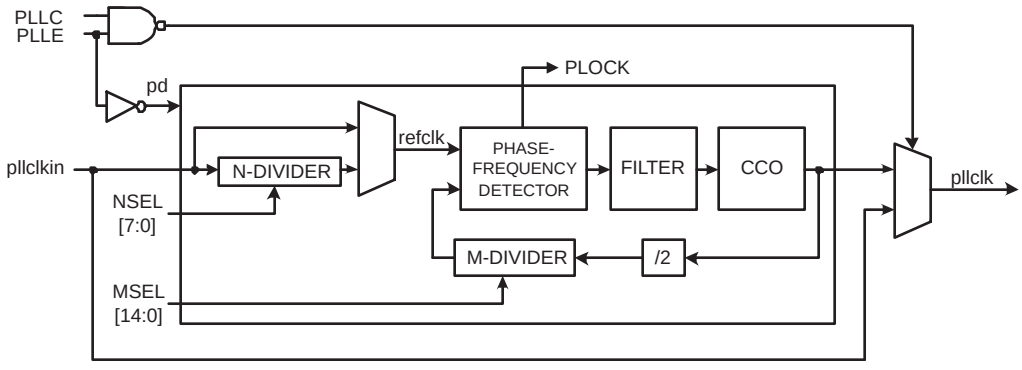

Fig 9. PLL0 block diagram

4.5.3 PLL0 Control register (PLL0CON - 0x400F C080)

The PLL0CON register contains the bits that enable and connect PLL0. Enabling PLL0 allows it to attempt to lock to the current settings of the multiplier and divider values. Connecting PLL0 causes the processor and most chip functions to run from the PLL0 output clock. Changes to the PLL0CON register do not take effect until a correct PLL0

feed sequence has been given (see Section 4.5.8 “PLL0 Feed register (PLL0FEED - 0x400F C08C)”).

Table 19. PLL Control register (PLL0CON - address 0x400F C080) bit description

Bit	Symbol	Description	Reset value
0	PLLE0	PLL0 Enable. When one, and after a valid PLL0 feed, this bit will activate PLL0 and allow it to lock to the requested frequency. See PLL0STAT register, Table 22.	0
1	PLLC0	PLL0 Connect. Setting PLLC0 to one after PLL0 has been enabled and locked, then followed by a valid PLL0 feed sequence causes PLL0 to become the clock source for the CPU, AHB peripherals, and used to derive the clocks for APB peripherals. The PLL0 output may potentially be used to clock the USB subsystem if the frequency is 48 MHz. See PLL0STAT register, Table 22.	0
31:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

PLL0 must be set up, enabled, and Lock established before it may be used as a clock source. When switching from the oscillator clock to the PLL0 output or vice versa, internal circuitry synchronizes the operation in order to ensure that glitches are not generated. Hardware does not insure that PLL0 is locked before it is connected or automatically disconnect PLL0 if lock is lost during operation. In the event of loss of lock on PLL0, it is likely that the oscillator clock has become unstable and disconnecting PLL0 will not remedy the situation.

4.5.4 PLL0 Configuration register (PLL0CFG - 0x400F C084)

The PLL0CFG register contains PLL0 multiplier and divider values. Changes to the PLL0CFG register do not take effect until a correct PLL feed sequence has been given (see Section 4.5.8 “PLL0 Feed register (PLL0FEED - 0x400F C08C)”). Calculations for the PLL frequency, and multiplier and divider values are found in the Section 4.5.10 “PLL0 frequency calculation”.

Table 20. PLL0 Configuration register (PLL0CFG - address 0x400F C084) bit description

Bit	Symbol	Description	Reset value
14:0	MSEL0	PLL0 Multiplier value. Supplies the value "M" in PLL0 frequency calculations. The value stored here is M - 1. Supported values for M are 6 through 512 and those listed in Table 21. **Note:** Not all values of M are needed, and therefore some are not supported by hardware. For details on selecting values for MSEL0 see Section 4.5.10 “PLL0 frequency calculation”.	0
15	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
23:16	NSEL0	PLL0 Pre-Divider value. Supplies the value "N" in PLL0 frequency calculations. The value stored here is N - 1. Supported values for N are 1 through 32. **Note:** For details on selecting the right value for NSEL0 see Section 4.5.10 “PLL0 frequency calculation”.	0
31:24	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 21. Multiplier values for PLL0 with a 32 kHz input

Multiplier (M)	Pre-divide (N)	F_{CCO}	Multiplier (M)	Pre-divide (N)	F_{CCO}
4272	1	279.9698	12085	2	396.0013
4395	1	288.0307	12207	2	399.9990
4578	1	300.0238	12817	2	419.9875
4725	1	309.6576	12817	3	279.9916
4807	1	315.0316	13184	2	432.0133
5127	1	336.0031	13184	3	288.0089
5188	1	340.0008	13672	2	448.0041
5400	1	353.8944	13733	2	450.0029
5493	1	359.9892	13733	3	300.0020
5859	1	383.9754	13916	2	455.9995
6042	1	395.9685	14099	2	461.9960
6075	1	398.1312	14420	3	315.0097
6104	1	400.0317	14648	2	479.9857
6409	1	420.0202	15381	2	504.0046
6592	1	432.0133	15381	3	336.0031
6750	1	442.3680	15564	3	340.0008
6836	1	448.0041	15625	2	512.0000
6866	1	449.9702	15869	2	519.9954
6958	1	455.9995	16113	2	527.9908
7050	1	462.0288	16479	3	359.9892
7324	1	479.9857	17578	3	383.9973
7425	1	486.6048	18127	3	395.9904
7690	1	503.9718	18311	3	400.0099
7813	1	512.0328	19226	3	419.9984
7935	1	520.0282	19775	3	431.9915
8057	1	528.0236	20508	3	448.0041
8100	1	530.8416	20599	3	449.9920
8545	2	280.0026	20874	3	455.9995
8789	2	287.9980	21149	3	462.0070
9155	2	299.9910	21973	3	480.0075
9613	2	314.9988	23071	3	503.9937
10254	2	336.0031	23438	3	512.0109
10376	2	340.0008	23804	3	520.0063
10986	2	359.9892	24170	3	528.0017
11719	2	384.0082			

4.5.5 PLL0 Status register (PLL0STAT - 0x400F C088)

The read-only PLL0STAT register provides the actual PLL0 parameters that are in effect at the time it is read, as well as PLL0 status. PLL0STAT may disagree with values found in PLL0CON and PLL0CFG because changes to those registers do not take effect until a

proper PLL0 feed has occurred (see Section 4.5.8 “PLL0 Feed register (PLL0FEED - 0x400F C08C)”).

Table 22. PLL Status register (PLL0STAT - address 0x400F C088) bit description

Bit	Symbol	Description	Reset value
14:0	MSEL0	Read-back for the PLL0 Multiplier value. This is the value currently used by PLL0, and is one less than the actual multiplier.	0
15	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
23:16	NSEL0	Read-back for the PLL0 Pre-Divider value. This is the value currently used by PLL0, and is one less than the actual divider.	0
24	PLLE0_STAT	Read-back for the PLL0 Enable bit. This bit reflects the state of the PLEC0 bit in PLL0CON (see Table 19) after a valid PLL0 feed. When one, PLL0 is currently enabled. When zero, PLL0 is turned off. This bit is automatically cleared when Power-down mode is entered.	0
25	PLLC0_STAT	Read-back for the PLL0 Connect bit. This bit reflects the state of the PLLC0 bit in PLL0CON (see Table 19) after a valid PLL0 feed. When PLLC0 and PLLE0 are both one, PLL0 is connected as the clock source for the CPU. When either PLLC0 or PLLE0 is zero, PLL0 is bypassed. This bit is automatically cleared when Power-down mode is entered.	0
26	PLOCK0	Reflects the PLL0 Lock status. When zero, PLL0 is not locked. When one, PLL0 is locked onto the requested frequency. See text for details.	0
31:27	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.5.6 PLL0 Interrupt: PLOCK0

The PLOCK0 bit in the PLL0STAT register reflects the lock status of PLL0. When PLL0 is enabled, or parameters are changed, PLL0 requires some time to establish lock under the new conditions. PLOCK0 can be monitored to determine when PLL0 may be connected for use. The value of PLOCK0 may not be stable when the PLL reference frequency (F_{REF}, the frequency of REFCLK, which is equal to the PLL input frequency divided by the pre-divider value) is less than 100 kHz or greater than 20 MHz. In these cases, the PLL may be assumed to be stable after a start-up time has passed. This time is 500 μs when FREF is greater than 400 kHz and 200 / FREF seconds when FREF is less than 400 kHz

PLOCK0 is connected to the interrupt controller. This allows for software to turn on PLL0 and continue with other functions without having to wait for PLL0 to achieve lock. When the interrupt occurs, PLL0 may be connected, and the interrupt disabled. PLOCK0 appears as interrupt 32 in Table 50. Note that PLOCK0 remains asserted whenever PLL0 is locked, so if the interrupt is used, the interrupt service routine must disable the PLOCK0 interrupt prior to exiting.

4.5.7 PLL0 Modes

The combinations of PLLE0 and PLLC0 are shown in Table 23.

Table 23. PLL control bit combinations

PLLC0	PLLE0	PLL Function
0	0	PLL0 is turned off and disconnected. PLL0 outputs the unmodified clock input.
0	1	PLL0 is active, but not yet connected. PLL0 can be connected after PLOCK0 is asserted.
1	0	Same as 00 combination. This prevents the possibility of PLL0 being connected without also being enabled.
1	1	PLL0 is active and has been connected as the system clock source.

4.5.8 PLL0 Feed register (PLL0FEED - 0x400F C08C)

A correct feed sequence must be written to the PLL0FEED register in order for changes to the PLL0CON and PLL0CFG registers to take effect. The feed sequence is:

1. Write the value 0xAA to PLL0FEED.
2. Write the value 0x55 to PLL0FEED.

The two writes must be in the correct sequence, and there must be no other register access in the same address space (0x400F C000 to 0x400F FFFF) between them. Because of this, it may be necessary to disable interrupts for the duration of the PLL0 feed operation, if there is a possibility that an interrupt service routine could write to another register in that space. If either of the feed values is incorrect, or one of the previously mentioned conditions is not met, any changes to the PLL0CON or PLL0CFG register will not become effective.

Table 24. PLL Feed register (PLL0FEED - address 0x400F C08C) bit description

Bit	Symbol	Description	Reset value
7:0	PLL0FEED	The PLL0 feed sequence must be written to this register in order for PLL0 configuration and control register changes to take effect.	0x00
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.5.9 PLL0 and Power-down mode

Power-down mode automatically turns off and disconnects PLL0. Wake-up from Power-down mode does not automatically restore PLL0 settings, this must be done in software. Typically, a routine to activate PLL0, wait for lock, and then connect PLL0 can be called at the beginning of any interrupt service routine that might be called due to the wake-up. It is important not to attempt to restart PLL0 by simply feeding it when execution resumes after a wake-up from Power-down mode. This would enable and connect PLL0 at the same time, before PLL lock is established.

4.5.10 PLL0 frequency calculation

PLL0 equations use the following parameters:

Table 25. PLL frequency parameter

Parameter	Description
F_{IN}	the frequency of PLLCLKIN from the Clock Source Selection Multiplexer.
F_{CCO}	the frequency of the PLLCLK (output of the PLL Current Controlled Oscillator)
N	PLL0 Pre-divider value from the NSEL0 bits in the PLL0CFG register (PLL0CFG NSEL0 field + 1). N is an integer from 1 through 32.
M	PLL0 Multiplier value from the MSEL0 bits in the PLL0CFG register (PLL0CFG MSEL0 field + 1). Not all potential values are supported. See below.
F_{REF}	PLL internal reference frequency, FIN divided by N.

The PLL0 output frequency (when PLL0 is both active and connected) is given by:

$F_{CCO} = (2 \times M \times F_{IN}) / N$

PLL inputs and settings must meet the following:

- F_{IN} is in the range of 32 kHz to 50 MHz.
- F_{CCO} is in the range of 275 MHz to 550 MHz.

The equation can be solved for other PLL parameters:

$M = (F_{CCO} \times N) / (2 \times F_{IN})$

$N = (2 \times M \times F_{IN}) / F_{CCO}$

$F_{IN} = (F_{CCO} \times N) / (2 \times M)$

Allowed values for M:

At higher oscillator frequencies, in the MHz range, values of M from 6 through 512 are allowed. This supports the entire useful range of both the main oscillator and the IRC.

For lower frequencies, specifically when the RTC is used to clock PLL0, a set of 65 additional M values have been selected for supporting baud rate generation, CAN operation, and obtaining integer MHz frequencies. These values are shown in Table 26.

Table 26. Additional Multiplier Values for use with a Low Frequency Clock Input

Low Frequency PLL Multipliers				
4272	4395	4578	4725	4807
5127	5188	5400	5493	5859
6042	6075	6104	6409	6592
6750	6836	6866	6958	7050
7324	7425	7690	7813	7935
8057	8100	8545	8789	9155
9613	10254	10376	10986	11719
12085	12207	12817	13184	13672
13733	13916	14099	14420	14648
15381	15564	15625	15869	16113

Table 26. Additional Multiplier Values for use with a Low Frequency Clock Input

Low Frequency PLL Multipliers				
16479	17578	18127	18311	19226
19775	20508	20599	20874	21149
21973	23071	23438	23804	24170

4.5.11 Procedure for determining PLL0 settings

PLL0 parameter determination can be simplified by using a spreadsheet available from NXP. To determine PLL0 parameters by hand, the following general procedure may be used:

1. Determine if the application requires use of the USB interface, and whether it will be clocked from PLL0. The USB requires a 50% duty cycle clock of 48 MHz within a very small tolerance, which means that F_{CCO} must be an even integer multiple of 48 MHz (i.e. an integer multiple of 96 MHz), within a very small tolerance.
2. Choose the desired processor operating frequency (CCLK). This may be based on processor throughput requirements, need to support a specific set of UART baud rates, etc. Bear in mind that peripheral devices may be running from a lower clock frequency than that of the processor (see Section 4.7 “Clock dividers” on page 56 and Section 4.8 “Power control” on page 60). Find a value for F_{CCO} that is close to a multiple of the desired CCLK frequency, bearing in mind the requirement for USB support in [1] above, and that lower values of F_{CCO} result in lower power dissipation.
3. Choose a value for the PLL input frequency (F_{IN}). This can be a clock obtained from the main oscillator, the RTC oscillator, or the on-chip RC oscillator. For USB support, the main oscillator should be used. Bear in mind that if PLL1 rather than PLL0 is used to clock the USB subsystem, this affects the choice of the main oscillator frequency.
4. Calculate values for M and N to produce a sufficiently accurate F_{CCO} frequency. The desired M value -1 will be written to the MSEL0 field in PLL0CFG. The desired N value -1 will be written to the NSEL0 field in PLL0CFG.

In general, it is better to use a smaller value for N, to reduce the level of multiplication that must be accomplished by the CCO. Due to the difficulty in finding the best values in some cases, it is recommended to use a spreadsheet or similar method to show many possibilities at once, from which an overall best choice may be selected. A spreadsheet is available from NXP for this purpose.

4.5.12 Examples of PLL0 settings

The following table gives a summary of examples that illustrate selecting PLL0 values based on different system requirements.

Table 27. Summary of PLL0 examples

Example	Description
1	• The PLL0 clock source is 10 MHz. • PLL0 is not used as the USB clock source, or the USB interface is not used. • The desired CPU clock is 100 MHz.
2	• The PLL0 clock source is 4 MHz. • PLL0 is used as the USB clock source. • The desired CPU clock is 60 MHz.
3	• The PLL0 clock source is the 32.768 kHz RTC clock. • PLL0 is not used as the USB clock source, or the USB interface is not used. • The desired CPU clock is 72 MHz.

Example 1

Assumptions:

- The USB interface will not be used in the application, or will be clocked by PLL1.
- The desired CPU rate is 100 MHz.
- An external 10 MHz crystal or clock source will be used as the system clock source.

Calculations:

$M = (FCCO \times N) / (2 \times F_{IN})$

A smaller value for the PLL pre-divide (N) as well as a smaller value of the multiplier (M), both result in better PLL operational stability and lower output jitter. Lower values of F_{CCO} also save power. So, the process of determining PLL setup parameters involves looking for the smallest N and M values giving the lowest F_{CCO} value that will support the required CPU and/or USB clocks. It is usually easier to work backward from the desired output clock rate and determine a target F_{CCO} rate, then find a way to obtain that F_{CCO} rate from the available input clock.

Potential precise values of F_{CCO} are integer multiples of the desired CPU clock. In this example, it is clear that the smallest frequency for F_{CCO} that can produce the desired CPU clock rate and is within the PLL0 operating range of 275 to 550 MHz is 300 MHz (3 ´ 100 MHz).

Assuming that the PLL pre-divide is 1 (N = 1), the equation above gives $M = ((300 \text{ ´ } 10^{6} \text{ ´ } 1) / (2 \times 10 \text{ ´ } 10^{6}) = 300 / 20 = 15$. Since the result is an integer, there is no need to look any further for a good set of PLL0 configuration values. The value written to PLL0CFG would be 0x0E (N - 1 = 0; M - 1 = 14 gives 0x0E).

The PLL output must be further divided in order to produce the CPU clock. This is accomplished using a separate divider that is described later in this chapter, see Section 4.7.1.

Example 2

Assumptions:

- The USB interface will be used in the application and will be clocked from PLL0.
- The desired CPU rate is 60 MHz.
- An external 4 MHz crystal or clock source will be used as the system clock source. This clock source could be the Internal RC oscillator (IRC).

Calculations:

$M = (FCCO \times N) / (2 \times F_{IN})$

Because supporting USB requires a precise 48 MHz clock with a 50% duty cycle, that need must be addressed first. Potential precise values of F_{CCO} are integer multiples of the 2 ´ the 48 MHz USB clock. The 2 ´ insures that the clock has a 50% duty cycle, which would not be the case for a division of the PLL output by an odd number.

The possibilities for the F_{CCO} rate when the USB is used are 288 MHz, 384 MHz, and 480 MHz. The smallest frequency for F_{CCO} that can produce a valid USB clock rate and is within the PLL0 operating range is 288 MHz (3 ´ 2 ´ 48 MHz).

Start by assuming N = 1, since this produces the smallest multiplier needed for PLL0. So, $M = ((288 \text{ ´ } 10^{6)} \text{ ´ } 1)) / (2 \times (4 \text{ ´ } 10^{6)}) = 288 / 8 = 36$. The result is an integer, which is necessary to obtain a precise USB clock. The value written to PLL0CFG would be 0x23 (N - 1 = 0; M - 1 = 35 = 0x23).

The potential CPU clock rate can be determined by dividing F_{CCO} by the desired CPU frequency: $288 \text{ ´ } 10^6 / 60 \text{ ´ } 10^6 = 4.8$. The nearest integer value for the CPU Clock Divider is then 5, giving us 57.6 MHz as the nearest value to the desired CPU clock rate.

If it is important to obtain exactly 60 MHz, an F_{CCO} rate must be found that can be divided down to both 48 MHz and 60 MHz. As previously noted, the possibilities for the F_{CCO} rate when the USB is used are 288 MHz, 384 MHz, and 480 MHz. Of these, only is 480 MHz is also evenly divisible by 60. Divided by 10, this gives the 48 MHz with a 50% duty cycle needed by the USB subsystem. Divided by 8, it gives 60 MHz for the CPU clock. PLL0 settings for 480 MHz are N = 1 and M = 60.

The PLL output must be further divided in order to produce both the CPU clock and the USB clock. This is accomplished using separate dividers that are described later in this chapter. See Section 4.7.1 and Section 4.7.2.

Example 3

Assumptions:

- The USB interface will not be used in the application, or will be clocked by PLL1.
- The desired CPU rate is 72 MHz
- The 32.768 kHz RTC clock source will be used as the system clock source

Calculations:

$M = (F_{CCO} \times N) / (2 \times F_{IN})$

The smallest integer multiple of the desired CPU clock rate that is within the PLL0 operating range is 288 MHz (4 ´ 72 MHz).

Using the equation above and assuming that N = 1, M = ((288 ´ $10^{6)}$ ´ $^{1)}$ / (2 × 32,768) = 4,394.53125. This is not an integer, so the CPU frequency will not be exactly 72 MHz with this setting. Since this example is less obvious, it may be useful to make a table of possibilities for different values of N (see below).

Table 28. Potential values for PLL example

N	M	M Rounded	FREF in Hz (FIN / N)	FCCO in MHz (FREF x M)	CCLK in MHz (FCCO / 4)	% Error (CCLK-72) / 72
1	4394.53125	4395	32768	288.0307	72.0077	0.0107
2	8789.0625	8789	16384	287.9980	71.9995	-0.0007
3	13183.59375	13184	10922.67	288.0089	72.0022	0.0031
4	17578.125	17578	8192	287.9980	71.9995	-0.0007
5	21972.65625	21973	6553.6	288.0045	72.0011	0.0016

Beyond N = 5, the value of M is out of range or not supported, so the table stops at that point. In the third column of the table, the calculated M value is rounded to the nearest integer. If this results in CCLK being above the maximum operating frequency, it is allowed if it is not more than 1/2 % above the maximum frequency.

In general, larger values of FREF result in a more stable PLL when the input clock is a low frequency. Even the first table entry shows a very small error of just over 1 hundredth of a percent, or 107 parts per million (ppm). If that is not accurate enough in the application, the second case gives a much smaller error of 7 ppm. There are no allowed combinations that give a smaller error than that.

Remember that when a frequency below about 1 MHz is used as the PLL0 clock source, not all multiplier values are available. As it turns out, all of the rounded M values found in Table 28 of this example are supported, which may be confirmed in Table 26. If PLL0 calculations suggest use of unsupported multiplier values, those values must be disregarded and other values examined to find the best fit.

The value written to PLL0CFG for the second table entry would be 0x12254 (N - 1 = 1 = 0x1; M - 1 = 8788 = 0x2254).

The PLL output must be further divided in order to produce the CPU clock. This is accomplished using a separate divider that is described later in this chapter, see Section 4.7.1.

4.5.13 PLL0 setup sequence

The following sequence must be followed step by step in order to have PLL0 initialized and running:

1. Disconnect PLL0 with one feed sequence if PLL0 is already connected.
2. Disable PLL0 with one feed sequence.
3. Change the CPU Clock Divider setting to speed up operation without PLL0, if desired.
4. Write to the Clock Source Selection Control register to change the clock source if needed.
5. Write to the PLL0CFG and make it effective with one feed sequence. The PLL0CFG can only be updated when PLL0 is disabled.
6. Enable PLL0 with one feed sequence.
7. Change the CPU Clock Divider setting for the operation with PLL0. It is critical to do this before connecting PLL0.
8. Wait for PLL0 to achieve lock by monitoring the PLOCK0 bit in the PLL0STAT register, or using the PLOCK0 interrupt, or wait for a fixed time when the input clock to PLL0 is slow (i.e. 32 kHz). The value of PLOCK0 may not be stable when the PLL reference frequency (FREF, the frequency of REFCLK, which is equal to the PLL input frequency divided by the pre-divider value) is less than 100 kHz or greater than 20 MHz. In these cases, the PLL may be assumed to be stable after a start-up time has passed. This time is 500 µs when FREF is greater than 400 kHz and 200 / FREF seconds when FREF is less than 400 kHz.
9. Connect PLL0 with one feed sequence.

It is very important not to merge any steps above. For example, do not update the PLL0CFG and enable PLL0 simultaneously with the same feed sequence.

4.6 PLL1 (Phase Locked Loop 1)

PLL1 receives its clock input from the main oscillator only and can be used to provide a fixed 48 MHz clock only to the USB subsystem. This is an option in addition to the possibility of generating the USB clock from PLL0.

PLL1 is disabled and powered off on reset. If PLL1 is left disabled, the USB clock can be supplied by PLL0 if everything is set up to provide 48 MHz through that route. If PLL1 is enabled and connected via the PLL1CON register (see Section 4.6.2), it is automatically selected to drive the USB subsystem (see Figure 7).

PLL1 activation is controlled via the PLL1CON register. PLL1 multiplier and divider values are controlled by the PLL1CFG register. These two registers are protected in order to prevent accidental alteration of PLL1 parameters or deactivation of PLL1. The protection is accomplished by a feed sequence similar to that of the Watchdog Timer. Details are provided in the description of the PLL1FEED register.

PLL1 accepts an input clock frequency in the range of 10 MHz to 25 MHz only. The input frequency is multiplied up to the range of 48 MHz for the USB clock using a Current Controlled Oscillator (CCO). The multiplier can be an integer value from 1 to 32 (for USB, the multiplier value cannot be higher than 4. The CCO operates in the range of 156 MHz to 320 MHz, so there is an additional divider in the loop to keep the CCO within its frequency range while PLL1 is providing the desired output frequency. The output divider may be set to divide by 2, 4, 8, or 16 to produce the output clock. Since the minimum output divider value is 2, it is insured that the output of PLL1 has a 50% duty cycle. A block diagram of PLL1 is shown in Figure 10.

4.6.1 PLL1 register description

PLL1 is controlled by the registers shown in Table 29. More detailed descriptions follow. Writes to any unused bits are ignored. A read of any unused bits will return a logic zero.

Warning: Improper setting of PLL1 values may result in incorrect operation of the USB subsystem!

Table 29. PLL1 registers

Name	Description	Access	Reset value[1]	Address
PLL1CON	PLL1 Control Register. Holding register for updating PLL1 control bits. Values written to this register do not take effect until a valid PLL1 feed sequence has taken place.	R/W	0	0x400F C0A0

Table 29. PLL1 registers

Name	Description	Access	Reset value[1]	Address
PLL1CFG	PLL1 Configuration Register. Holding register for updating PLL1 configuration values. Values written to this register do not take effect until a valid PLL1 feed sequence has taken place.	R/W	0	0x400F C0A4
PLL1STAT	PLL1 Status Register. Read-back register for PLL1 control and configuration information. If PLL1CON or PLL1CFG have been written to, but a PLL1 feed sequence has not yet occurred, they will not reflect the current PLL1 state. Reading this register provides the actual values controlling PLL1, as well as PLL1 status.	RO	0	0x400F C0A8
PLL1FEED	PLL1 Feed Register. This register enables loading of PLL1 control and configuration information from the PLL1CON and PLL1CFG registers into the shadow registers that actually affect PLL1 operation.	WO	NA	0x400F C0AC

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

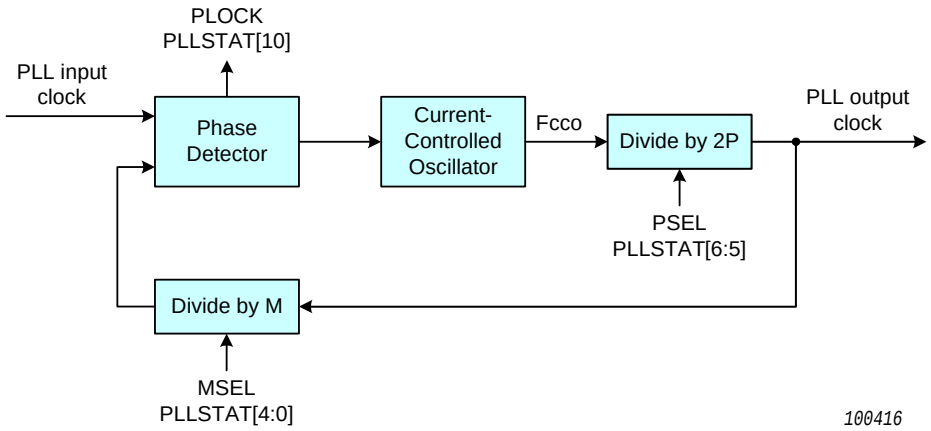

Fig 10. PLL1 block diagram

4.6.2 PLL1 Control register (PLL1CON - 0x400F C0A0)

The PLL1CON register contains the bits that enable and connect PLL1. Enabling PLL1 allows it to attempt to lock to the current settings of the multiplier and divider values. Connecting PLL1 causes the USB subsystem to run from the PLL1 output clock. Changes to the PLL1CON register do not take effect until a correct PLL feed sequence has been given (see Section 4.6.6 and Section 4.6.3).

Table 30. PLL1 Control register (PLL1CON - address 0x400F C0A0) bit description

Bit	Symbol	Description	Reset value
0	PLLE1	PLL1 Enable. When one, and after a valid PLL1 feed, this bit will activate PLL1 and allow it to lock to the requested frequency. See PLL1STAT register, Table 32.	0
1	PLLC1	PLL1 Connect. Setting PLLC to one after PLL1 has been enabled and locked, then followed by a valid PLL1 feed sequence causes PLL1 to become the clock source for the USB subsystem via the USB clock divider. See PLL1STAT register, Table 32.	0
31:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

PLL1 must be set up, enabled, and lock established before it may be used as a clock source for the USB subsystem. The hardware does not insure that the PLL is locked before it is connected nor does it automatically disconnect the PLL if lock is lost during operation.

4.6.3 PLL1 Configuration register (PLL1CFG - 0x400F C0A4)

The PLL1CFG register contains the PLL1 multiplier and divider values. Changes to the PLL1CFG register do not take effect until a correct PLL1 feed sequence has been given (see Section 4.6.6). Calculations for the PLL1 frequency, and multiplier and divider values are found in Section 4.6.9.

Table 31. PLL Configuration register (PLL1CFG - address 0x400F C0A4) bit description

Bit	Symbol	Description	Reset value
4:0	MSEL1	PLL1 Multiplier value. Supplies the value "M" in the PLL1 frequency calculations. **Note:** For details on selecting the right value for MSEL1 see Section 4.6.8.	0
6:5	PSEL1	PLL1 Divider value. Supplies the value "P" in the PLL1 frequency calculations. **Note:** For details on selecting the right value for PSEL1 see Section 4.6.8.	0
31:7	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.6.4 PLL1 Status register (PLL1STAT - 0x400F C0A8)

The read-only PLL1STAT register provides the actual PLL1 parameters that are in effect at the time it is read, as well as the PLL1 status. PLL1STAT may disagree with values found in PLL1CON and PLL1CFG because changes to those registers do not take effect until a proper PLL1 feed has occurred (see Section 4.6.6 “PLL1 Feed register (PLL1FEED - 0x400F C0AC)”).

Table 32. PLL1 Status register (PLL1STAT - address 0x400F C0A8) bit description

Bit	Symbol	Description	Reset value
4:0	MSEL1	Read-back for the PLL1 Multiplier value. This is the value currently used by PLL1.	0
6:5	PSEL1	Read-back for the PLL1 Divider value. This is the value currently used by PLL1.	0
7	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
8	PLLE1_STAT	Read-back for the PLL1 Enable bit. When one, PLL1 is currently activated. When zero, PLL1 is turned off. This bit is automatically cleared when Power-down mode is activated.	0
9	PLLC1_STAT	Read-back for the PLL1 Connect bit. When PLLC and PLLE are both one, PLL1 is connected as the clock source for the microcontroller. When either PLLC or PLLE is zero, PLL1 is bypassed and the oscillator clock is used directly by the microcontroller. This bit is automatically cleared when Power-down mode is activated.	0
10	PLOCK1	Reflects the PLL1 Lock status. When zero, PLL1 is not locked. When one, PLL1 is locked onto the requested frequency.	0
31:11	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.6.4.1 PLL1 modes

The combinations of PLLE1 and PLLC1 are shown in Table 33.

Table 33. PLL1 control bit combinations

PLLC1	PLLE1	PLL1 Function
0	0	PLL1 is turned off and disconnected.
0	1	PLL1 is active, but not yet connected. PLL1 can be connected after PLOCK1 is asserted.
1	0	Same as 00 combination. This prevents the possibility of PLL1 being connected without also being enabled.
1	1	PLL1 is active and has been connected. The clock for the USB subsystem is sourced from PLL1.

4.6.5 PLL1 Interrupt: PLOCK1

The PLOCK1 bit in the PLL1STAT register reflects the lock status of PLL1. When PLL1 is enabled, or parameters are changed, the PLL requires some time to establish lock under the new conditions. PLOCK1 can be monitored to determine when the PLL may be connected for use.

PLOCK1 is connected to the interrupt controller. This allows for software to turn on the PLL and continue with other functions without having to wait for the PLL to achieve lock. When the interrupt occurs, the PLL may be connected, and the interrupt disabled. PLOCK1 appears as interrupt 48 in Table 50. Note that PLOCK1 remains asserted whenever PLL1 is locked, so if the interrupt is used, the interrupt service routine must disable the PLOCK1 interrupt prior to exiting.

4.6.6 PLL1 Feed register (PLL1FEED - 0x400F C0AC)

A correct feed sequence must be written to the PLL1FEED register in order for changes to the PLL1CON and PLL1CFG registers to take effect. The feed sequence is:

1. Write the value 0xAA to PLL1FEED.
2. Write the value 0x55 to PLL1FEED.

The two writes must be in the correct sequence, and there must be no other register access in the same address space (0x400F C000 to 0x400F FFFF) between them. Because of this, it may be necessary to disable interrupts for the duration of the PLL feed operation, if there is a possibility that an interrupt service routine could write to another register in that space. If either of the feed values is incorrect, or one of the previously mentioned conditions is not met, any changes to the PLL1CON or PLL1CFG register will not become effective.

Table 34. PLL1 Feed register (PLL1FEED - address 0x400F C0AC) bit description

Bit	Symbol	Description	Reset value
7:0	PLL1FEED	The PLL1 feed sequence must be written to this register in order for PLL1 configuration and control register changes to take effect.	0x00
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.6.7 PLL1 and Power-down mode

Power-down mode automatically turns off and disconnects activated PLL(s). Wake-up from Power-down mode does not automatically restore PLL settings, this must be done in software. Typically, a routine to activate the PLL, wait for lock, and then connect the PLL can be called at the beginning of any interrupt service routine that might be called due to the wake-up. It is important not to attempt to restart a PLL by simply feeding it when execution resumes after a wake-up from Power-down mode. This would enable and connect the PLL at the same time, before PLL lock is established.

If activity on the USB data lines is not selected to wake the microcontroller from Power-down mode (see Section 4.8.8 for details of wake up from reduced modes), both the Main PLL (PLL0) and the USB PLL (PLL1) will be automatically be turned off and disconnected when Power-down mode is invoked, as described above. However, if the USB activity interrupt is enabled and USB_NEED_CLK = 1 (see Table 192 for a description of USB_NEED_CLK), it is not possible to go into Power-down mode and any attempt to set the PD bit will fail, leaving the PLLs in the current state.

4.6.8 PLL1 frequency calculation

The PLL1 equations use the following parameters:

Table 35. Elements determining PLL frequency

Element	Description
F_{OSC}	the frequency from the crystal oscillator
F_{CCO}	the frequency of the PLL1 current controlled oscillator
USBCLK	the PLL1 output frequency (48 MHz for USB)
M	PLL1 Multiplier value from the MSEL1 bits in the PLL1CFG register
P	PLL1 Divider value from the PSEL1 bits in the PLL1CFG register

The PLL1 output frequency (when the PLL is both active and connected) is given by:

$USBCLK = M \times F_{OSC}$ or $USBCLK = F_{CCO} / (2 \times P)$

The CCO frequency can be computed as:

$F_{CCO} = USBCLK \times 2 \times P$ or $F_{CCO} = F_{OSC}$ ´ $M \times 2 \times P$

The PLL1 inputs and settings must meet the following criteria:

- F_{OSC} is in the range of 10 MHz to 25 MHz.
- USBCLK is 48 MHz.
- F_{CCO} is in the range of 156 MHz to 320 MHz.

4.6.9 Procedure for determining PLL1 settings

The PLL1 configuration for USB may be determined as follows:

1. The desired PLL1 output frequency is USBCLK = 48 MHz.
2. Choose an oscillator frequency (F_{OSC}). USBCLK must be the whole (non-fractional) multiple of F_{OSC} meaning that the possible values for F_{OSC} are 12 MHz, 16 MHz, and 24 MHz.
3. Calculate the value of M to configure the MSEL1 bits. $M = USBCLK / F_{OSC}$. In this case, the possible values for M = 2, 3, or 4 (F_{OSC} = 24 MHz, 16 MHz, or 12 MHz). The value written to the MSEL1 bits in PLL1CFG is M - 1 (see Table 37).
4. Find a value for P to configure the PSEL1 bits, such that F_{CCO} is within its defined frequency limits of 156 MHz to 320 MHz. F_{CCO} is calculated using $F_{CCO} = USBCLK \times 2 \times P$. It follows that P = 2 is the only P value to yield F_{CCO} in the allowed range. The value written to the PSEL1 bits in PLL1CFG is ‘01’ for P = 2 (see Table 36).

Table 36. PLL1 Divider values

Values allowed for using PLL1 with USB are highlighted.

PSEL1 Bits (PLL1CFG bits [6:5])	Value of P
00	1
01	**2**
10	4
11	8

Table 37. PLL1 Multiplier values

Values allowed for using PLL1 with USB are highlighted.

MSEL1 Bits (PLL1CFG bits [4:0])	Value of M
00000	1
00001	**2**
00010	**3**
00011	**4**
...	...
11110	31
11111	32

4.7 Clock dividers

The output of the PLL0 must be divided down for use by the CPU and the USB subsystem (if used with PLL0, see Section 4.6). Separate dividers are provided such that the CPU frequency can be determined independently from the USB subsystem, which always requires 48 MHz with a 50% duty cycle for proper operation.

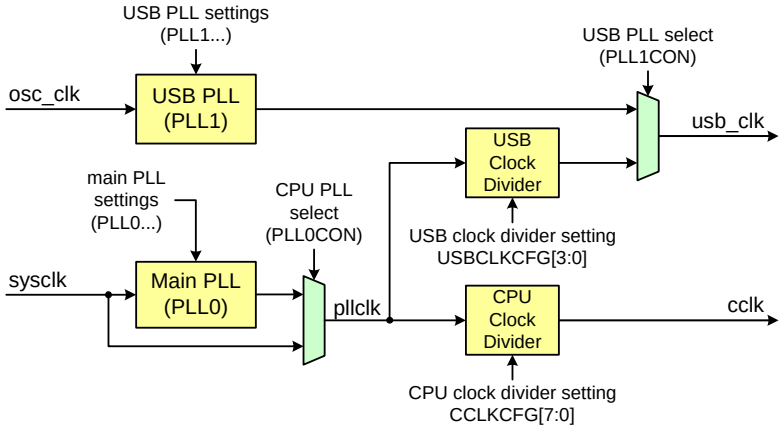

Fig 11. PLLs and clock dividers

4.7.1 CPU Clock Configuration register (CCLKCFG - 0x400F C104)

The CCLKCFG register controls the division of the PLL0 output before it is used by the CPU. When PLL0 is bypassed, the division may be by 1. When PLL0 is running, the output must be divided in order to bring the CPU clock frequency (CCLK) within operating limits. An 8-bit divider allows a range of options, including slowing CPU operation to a low rate for temporary power savings without turning off PLL0.

Table 38. CPU Clock Configuration register (CCLKCFG - address 0x400F C104) bit description

Bit	Symbol	Value	Description	Reset value
7:0	CCLKSEL		Selects the divide value for creating the CPU clock (CCLK) from the PLL0 output.	0x00
		0	pllclk is divided by 1 to produce the CPU clock. This setting is not allowed when the PLL0 is connected, because the rate would always be greater than the maximum allowed CPU clock.	
		1	pllclk is divided by 2 to produce the CPU clock. This setting is not allowed when the PLL0 is connected, because the rate would always be greater than the maximum allowed CPU clock.	
		2	pllclk is divided by 3 to produce the CPU clock.	
		3	pllclk is divided by 4 to produce the CPU clock.	
		4	pllclk is divided by 5 to produce the CPU clock.	
		:	:	
		255	pllclk is divided by 256 to produce the CPU clock.	
31:8	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

The CCLK is derived from the PLL0 output signal, divided by CCLKSEL + 1. Having CCLKSEL = 2 results in CCLK being one third of the PLL0 output, CCLKSEL = 3 results in CCLK being one quarter of the PLL0 output, etc.

4.7.2 USB Clock Configuration register (USBCLKCFG - 0x400F C108)

This register is used only if the USB PLL (PLL1) is not connected (via the PLLC1 bit in PLL1CON). If PLL1 is connected, its output is automatically used as the USB clock source, and PLL1 must be configured to supply the correct 48 MHz clock to the USB subsystem. If PLL1 is not connected, the USB subsystem will be driven by PLL0 via the USB clock divider.

The USBCLKCFG register controls the division of the PLL0 output before it is used by the USB subsystem.The PLL0 output must be divided in order to bring the USB clock frequency to 48 MHz with a 50% duty cycle. A 4-bit divider allows obtaining the correct USB clock from any even multiple of 48 MHz (i.e. any multiple of 96 MHz) within the PLL operating range.

Remark: The Internal RC oscillator should not be used to drive PLL0 when the USB is using PLL0 as a clock source because a more precise clock is needed for USB specification compliance (see Table 17).

Table 39. USB Clock Configuration register (USBCLKCFG - address 0x400F C108) bit description

Bit	Symbol	Value	Description	Reset value
3:0	USBSEL		Selects the divide value for creating the USB clock from the PLL0 output. Only the values shown below can produce even number multiples of 48 MHz from the PLL0 output. **Warning:** Improper setting of this value will result in incorrect operation of the USB interface.	0
		5	PLL0 output is divided by 6. PLL0 output must be 288 MHz.	
		7	PLL0 output is divided by 8. PLL0 output must be 384 MHz.	
		9	PLL0 output is divided by 10. PLL0 output must be 480 MHz.	
31:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

4.7.3 Peripheral Clock Selection registers 0 and 1 (PCLKSEL0 - 0x400F C1A8 and PCLKSEL1 - 0x400F C1AC)

A pair of bits in a Peripheral Clock Selection register controls the rate of the clock signal that will be supplied to the corresponding peripheral as specified in Table 40, Table 41 and Table 42.

Remark: The peripheral clock for the RTC block is fixed at CCLK/8.

Table 40. Peripheral Clock Selection register 0 (PCLKSEL0 - address 0x400F C1A8) bit description

Bit	Symbol	Description	Reset value
1:0	PCLK_WDT	Peripheral clock selection for WDT.	00
3:2	PCLK_TIMER0	Peripheral clock selection for TIMER0.	00
5:4	PCLK_TIMER1	Peripheral clock selection for TIMER1.	00
7:6	PCLK_UART0	Peripheral clock selection for UART0.	00
9:8	PCLK_UART1	Peripheral clock selection for UART1.	00
11:10	-	Reserved.	NA
13:12	PCLK_PWM1	Peripheral clock selection for PWM1.	00
15:14	PCLK_I2C0	Peripheral clock selection for I2C0.	00
17:16	PCLK_SPI	Peripheral clock selection for SPI.	00
19:18	-	Reserved.	NA
21:20	PCLK_SSP1	Peripheral clock selection for SSP1.	00
23:22	PCLK_DAC	Peripheral clock selection for DAC.	00
25:24	PCLK_ADC	Peripheral clock selection for ADC.	00
27:26	PCLK_CAN1	Peripheral clock selection for CAN1.[1]	00
29:28	PCLK_CAN2	Peripheral clock selection for CAN2.[1]	00
31:30	PCLK_ACF	Peripheral clock selection for CAN acceptance filtering.[1]	00

[1] PCLK_CAN1 and PCLK_CAN2 must have the same PCLK divide value when the CAN function is used.

Table 41. Peripheral Clock Selection register 1 (PCLKSEL1 - address 0x400F C1AC) bit description

Bit	Symbol	Description	Reset value
1:0	PCLK_QEI	Peripheral clock selection for the Quadrature Encoder Interface.	00
3:2	PCLK_GPIOINT	Peripheral clock selection for GPIO interrupts.	00
5:4	PCLK_PCB	Peripheral clock selection for the Pin Connect block.	00
7:6	PCLK_I2C1	Peripheral clock selection for I2C1.	00
9:8	-	Reserved.	NA
11:10	PCLK_SSP0	Peripheral clock selection for SSP0.	00
13:12	PCLK_TIMER2	Peripheral clock selection for TIMER2.	00
15:14	PCLK_TIMER3	Peripheral clock selection for TIMER3.	00
17:16	PCLK_UART2	Peripheral clock selection for UART2.	00
19:18	PCLK_UART3	Peripheral clock selection for UART3.	00
21:20	PCLK_I2C2	Peripheral clock selection for I2C2.	00
23:22	PCLK_I2S	Peripheral clock selection for I2S.	00
25:24	-	Reserved.	NA
27:26	PCLK_RIT	Peripheral clock selection for Repetitive Interrupt Timer.	00
29:28	PCLK_SYSCON	Peripheral clock selection for the System Control block.	00
31:30	PCLK_MC	Peripheral clock selection for the Motor Control PWM.	00

Table 42. Peripheral Clock Selection register bit values

PCLKSEL0 and PCLKSEL1 individual peripheral's clock select options	Function	Reset value
00	PCLK_peripheral = CCLK/4	00
01	PCLK_peripheral = CCLK	
10	PCLK_peripheral = CCLK/2	
11	PCLK_peripheral = CCLK/8, except for CAN1, CAN2, and CAN filtering when "11" selects = CCLK/6.	

4.8 Power control

The LPC176x/5x supports a variety of power control features: Sleep mode, Deep Sleep mode, Power-down mode, and Deep Power-down mode. The CPU clock rate may also be controlled as needed by changing clock sources, re-configuring PLL values, and/or altering the CPU clock divider value. This allows a trade-off of power versus processing speed based on application requirements. In addition, Peripheral Power Control allows shutting down the clocks to individual on-chip peripherals, allowing fine tuning of power consumption by eliminating all dynamic power use in any peripherals that are not required for the application.

Entry to any reduced power mode begins with the execution of either a WFI (Wait For Interrupt) or WFE (Wait For Exception) instruction by the Cortex-M3. The Cortex-M3 internally supports two reduced power modes: Sleep and Deep Sleep. These are selected by the SLEEPDEEP bit in the cortex-M3 System Control Register. Power-down and Deep Power-down modes are selected by bits in the PCON register. See Table 44. The same register contains flags that indicate whether entry into each reduced power mode actually occurred.

The LPC176x/5x also implements a separate power domain in order to allow turning off power to the bulk of the device while maintaining operation of the Real Time Clock.

Reduced power modes have some limitation during debug, see Section 33.5 for more information.

4.8.1 Sleep mode

Note: Sleep mode on the LPC176x/5x corresponds to the Idle mode on LPC2xxx series devices. The name is changed because ARM has incorporated portions of reduced power mode control into the Cortex-M3. LPC176x/5x documentation uses the Cortex-M3 terminology where applicable.

When Sleep mode is entered, the clock to the core is stopped, and the SMFLAG bit in PCON is set, see Table 44.Resumption from the Sleep mode does not need any special sequence but re-enabling the clock to the ARM core.

In Sleep mode, execution of instructions is suspended until either a Reset or an interrupt occurs. Peripheral functions 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 GPDMA may operate in Sleep mode to access AHB SRAMs and peripherals with GPDMA support, but the GPDMA cannot access the flash memory or the main SRAM, which are disabled in order to save power.

Wake-up from Sleep mode will occur whenever any enabled interrupt occurs.

4.8.2 Deep Sleep mode

Note: Deep Sleep mode on the LPC176x/5x corresponds to the Sleep mode on LPC23xx and LPC24xx series devices. The name is changed because ARM has incorporated portions of reduced power mode control into the Cortex-M3. LPC176x/5x documentation uses the Cortex-M3 terminology where applicable.

When the chip enters the Deep Sleep mode, the main oscillator is powered down, nearly all clocks are stopped, and the DSFLAG bit in PCON is set, see Table 44. The IRC remains running and can be configured to drive the Watchdog Timer, allowing the Watchdog to wake up the CPU. The 32 kHz RTC oscillator is not stopped and RTC interrupts may be used as a wake-up source. The flash is left in the standby mode allowing a quick wake-up. The PLLs are automatically turned off and disconnected. The CCLK and USBCLK clock dividers automatically get reset to zero.

The processor state and registers, peripheral registers, and internal SRAM values are preserved throughout Deep Sleep mode and the logic levels of chip pins remain static. The Deep Sleep mode can be terminated and normal operation resumed by either a Reset or certain specific interrupts that are able to function without clocks. Since all dynamic operation of the chip is suspended, Deep Sleep mode reduces chip power consumption to a very low value.

On the wake-up of Deep Sleep mode, if the IRC was used before entering Deep Sleep mode, a 2-bit IRC timer starts counting and the code execution and peripherals activities will resume after the timer expires (4 cycles). If the main external oscillator was used, the 12-bit main oscillator timer starts counting and the code execution will resume when the timer expires (4096 cycles). The user must remember to re-configure any required PLLs and clock dividers after the wake-up.

Wake-up from Deep Sleep mode can be brought about by NMI, External Interrupts EINT0 through EINT3, GPIO interrupts, the Ethernet Wake-on-LAN interrupt, Brownout Detect, an RTC Alarm interrupt, a Watchdog Timer time out, a USB input pin transition (USB activity interrupt), or a CAN input pin transition, when the related interrupt is enabled. Wake-up will occur whenever any enabled interrupt occurs.

4.8.3 Power-down mode

Power-down mode does everything that Deep Sleep mode does, but also turns off the flash memory. Entry to Power-down mode causes the PDFLAG bit in PCON to be set, see Table 44. This saves more power, but requires waiting for resumption of flash operation before execution of code or data access in the flash memory can be accomplished.

When the chip enters Power-down mode, the IRC, the main oscillator, and all clocks are stopped. The RTC remains running if it has been enabled and RTC interrupts may be used to wake up the CPU. The flash is forced into Power-down mode. The PLLs are automatically turned off and disconnected. The CCLK and USBCLK clock dividers automatically get reset to zero.

Upon wake-up from Power-down mode, if the IRC was used before entering Power-down mode, after IRC-start-up time (about 60 μs), the 2-bit IRC timer starts counting and expiring in 4 cycles. Code execution can then be resumed immediately following the expiration of the IRC timer if the code was running from SRAM. In the meantime, the flash wake-up timer measures flash start-up time of about 100 μs. When it times out, access to the flash is enabled. The user must remember to re-configure any required PLLs and clock dividers after the wake-up.

Wake-up from Power-down mode can be brought about by NMI, External Interrupts EINT0 through EINT3, GPIO interrupts, the Ethernet Wake-on-LAN interrupt, Brownout Detect, an RTC Alarm interrupt, a USB input pin transition (USB activity interrupt), or a CAN input pin transition, when the related interrupt is enabled.

4.8.4 Deep Power-down mode

In Deep Power-down mode, power is shut off to the entire chip with the exception of the Real-Time Clock, the RESET pin, the WIC, and the RTC backup registers. Entry to Deep Power-down mode causes the DPDFLAG bit in PCON to be set, see Table 44.

To optimize power conservation, the user has the additional option of turning off or retaining power to the 32 kHz oscillator. It is also possible to use external circuitry to turn off power to the on-chip regulator via the VDD(REG)(3V3) pins after entering Deep Power-down mode.Power to the on-chip regulator must be restored before device operation can be restarted.

Wake-up from Deep Power-down mode will occur when an external reset signal is applied, or the RTC interrupt is enabled and an RTC interrupt is generated.

4.8.5 Peripheral power control

A Power Control for Peripherals feature allows individual peripherals to be turned off if they are not needed in the application, resulting in additional power savings. This is detailed in the description of the PCONP register.

4.8.6 Register description

The Power Control function uses registers shown in Table 43. More detailed descriptions follow.

Table 43. Power Control registers

Name	Description	Access	Reset value[1]	Address
PCON	Power Control Register. This register contains control bits that enable some reduced power operating modes of the LPC176x/5x. See Table 44.	R/W	0x00	0x400F C0C0
PCONP	Power Control for Peripherals Register. This register contains control bits that enable and disable individual peripheral functions, allowing elimination of power consumption by peripherals that are not needed.	R/W		0x400F C0C4

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

4.8.7 Power Mode Control register (PCON - 0x400F C0C0)

Controls for some reduced power modes and other power related controls are contained in the PCON register, as described in Table 44.

Table 44. Power Mode Control register (PCON - address 0x400F C0C0) bit description

Bit	Symbol	Description	Reset value
0	PM0	Power mode control bit 0. This bit controls entry to the Power-down mode. See Section 4.8.7.1 below for details.	0
1	PM1	Power mode control bit 1. This bit controls entry to the Deep Power-down mode. See Section 4.8.7.1 below for details.	0
2	$\overline{\text{BODRPM}}$	Brown-Out Reduced Power Mode. When BODRPM is 1, the Brown-Out Detect circuitry will be turned off when chip Power-down mode or Deep Sleep mode is entered, resulting in a further reduction in power usage. However, the possibility of using Brown-Out Detect as a wake-up source from the reduced power mode will be lost. When 0, the Brown-Out Detect function remains active during Power-down and Deep Sleep modes. See the System Control Block chapter for details of Brown-Out detection.	0
3	$\overline{\text{BOGD}}$[1]	Brown-Out Global Disable. When BOGD is 1, the Brown-Out Detect circuitry is fully disabled at all times, and does not consume power. When 0, the Brown-Out Detect circuitry is enabled. See the System Control Block chapter for details of Brown-Out detection. **Note:** the Brown-Out Reset Disable (BORD, in this register) and the Brown-Out Interrupt (xx) must be disabled when software changes the value of this bit.	0
4	$\overline{\text{BORD}}$	Brown-Out Reset Disable. When BORD is 1, the BOD will not reset the device when the $V_{DD(REG)(3V3)}$ voltage dips goes below the BOD reset trip level. The Brown-Out interrupt is not affected. When BORD is 0, the BOD reset is enabled. See the Section 3.5 for details of Brown-Out detection.	0
7:5	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
8	SMFLAG	Sleep Mode entry flag. Set when the Sleep mode is successfully entered. Cleared by software writing a one to this bit.	0 [2][3]
9	DSFLAG	Deep Sleep entry flag. Set when the Deep Sleep mode is successfully entered. Cleared by software writing a one to this bit.	0 [2][3]
10	PDFLAG	Power-down entry flag. Set when the Power-down mode is successfully entered. Cleared by software writing a one to this bit.	0 [2][3]
11	DPDFLAG	Deep Power-down entry flag. Set when the Deep Power-down mode is successfully entered. Cleared by software writing a one to this bit.	0 [2][4]
31:12	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

[1] BOD reset (BORD- bit 4 in PCON register) and BOD interrupt needs to be disabled before a user disables or enables the power to BOD (BOGD - bit 3 in PCON register).

[2] Only one of these flags will be valid at a specific time.

[3] Hardware reset only for a power-up of core power or by a brownout detect event.

[4] Hardware reset only for a power-up event on Vbat.

4.8.7.1 Encoding of Reduced Power Modes

The PM1and PM0 bits in PCON allow entering reduced power modes as needed. The encoding of these bits allows backward compatibility with devices that previously only supported Sleep and Power-down modes. Table 45 below shows the encoding for the three reduced power modes supported by the LPC176x/5x.

Table 45. Encoding of reduced power modes

PM1, PM0	Description
00	Execution of WFI or WFE enters either Sleep or Deep Sleep mode as defined by the SLEEPDEEP bit in the Cortex-M3 System Control Register.
01	Execution of WFI or WFE enters Power-down mode if the SLEEPDEEP bit in the Cortex-M3 System Control Register is 1.
10	Reserved, this setting should not be used.
11	Execution of WFI or WFE enters Deep Power-down mode if the SLEEPDEEP bit in the Cortex-M3 System Control Register is 1.

4.8.8 Wake-up from Reduced Power Modes

Any enabled interrupt can wake up the CPU from Sleep mode. Certain interrupts can wake up the processor if it is in either Deep Sleep mode or Power-down mode.

Interrupts that can occur during Deep Sleep or Power-down mode will wake up the CPU if the interrupt is enabled. After wake-up, execution will continue to the appropriate interrupt service routine. These interrupts are NMI, External Interrupts EINT0 through EINT3, GPIO interrupts, Ethernet Wake-on-LAN interrupt, Brownout Detect, RTC Alarm, CAN Activity Interrupt, and USB Activity Interrupt. In addition, the watchdog timer can wake up the part from Deep Sleep mode if the watchdog timer is being clocked by the IRC oscillator. For the wake-up process to take place the corresponding interrupt must be enabled in the NVIC. For pin-related peripheral functions, the related functions must also be mapped to pins.

The CAN Activity Interrupt is generated by activity on the CAN bus pins, and the USB Activity Interrupt is generated by activity on the USB bus pins. These interrupts are only useful to wake up the CPU when it is on Deep Sleep or Power-down mode, when the peripheral functions are powered up, but not active. Typically, if these interrupts are used, their flags should be polled just before enabling the interrupt and entering the desired reduced power mode. This can save time and power by avoiding an immediate wake-up. Upon wake-up, the interrupt service can turn off the related activity interrupt, do any application specific setup, and exit to await a normal peripheral interrupt.

In Deep Power-down mode, internal power to most of the device is removed, which limits the possibilities for waking up from this mode. Wake-up from Deep Power-down mode will occur when an external reset signal is applied, or the RTC interrupt is enabled and an RTC interrupt is generated.

4.8.9 Power Control for Peripherals register (PCONP - 0x400F C0C4)

The PCONP register allows turning off selected peripheral functions for the purpose of saving power. This is accomplished by gating off the clock source to the specified peripheral blocks. A few peripheral functions cannot be turned off (i.e. the Watchdog timer, the Pin Connect block, and the System Control block).

Some peripherals, particularly those that include analog functions, may consume power that is not clock dependent. These peripherals may contain a separate disable control that turns off additional circuitry to reduce power. Information on peripheral specific power saving features may be found in the chapter describing that peripheral.

Each bit in PCONP controls one peripheral as shown in Table 46.

If a peripheral control bit is 1, that peripheral is enabled. If a peripheral control bit is 0, that peripheral's clock is disabled (gated off) to conserve power. For example if bit 19 is 1, the I^2C1 interface is enabled. If bit 19 is 0, the I^2C1 interface is disabled.

Important: valid read from a peripheral register and valid write to a peripheral register is possible only if that peripheral is enabled in the PCONP register!

Table 46. Power Control for Peripherals register (PCONP - address 0x400F C0C4) bit description

Bit	Symbol	Description	Reset value
0	-	Reserved.	NA
1	PCTIM0	Timer/Counter 0 power/clock control bit.	1
2	PCTIM1	Timer/Counter 1 power/clock control bit.	1
3	PCUART0	UART0 power/clock control bit.	1
4	PCUART1	UART1 power/clock control bit.	1
5	-	Reserved.	NA
6	PCPWM1	PWM1 power/clock control bit.	1
7	PCI2C0	The I^2C0 interface power/clock control bit.	1
8	PCSPI	The SPI interface power/clock control bit.	1
9	PCRTC	The RTC power/clock control bit.	1
10	PCSSP1	The SSP 1 interface power/clock control bit.	1
11	-	Reserved.	NA
12	PCADC	A/D converter (ADC) power/clock control bit. **Note:** Clear the PDN bit in the AD0CR before clearing this bit, and set this bit before setting PDN.	0
13	PCCAN1	CAN Controller 1 power/clock control bit.	0
14	PCCAN2	CAN Controller 2 power/clock control bit.	0
15	PCGPIO	Power/clock control bit for IOCON, GPIO, and GPIO interrupts.	1
16	PCRIT	Repetitive Interrupt Timer power/clock control bit.	0
17	PCMCPWM	Motor Control PWM	0
18	PCQEI	Quadrature Encoder Interface power/clock control bit.	0
19	PCI2C1	The I^2C1 interface power/clock control bit.	1
20	-	Reserved.	NA
21	PCSSP0	The SSP0 interface power/clock control bit.	1
22	PCTIM2	Timer 2 power/clock control bit.	0
23	PCTIM3	Timer 3 power/clock control bit.	0
24	PCUART2	UART 2 power/clock control bit.	0
25	PCUART3	UART 3 power/clock control bit.	0
26	PCI2C2	I2C interface 2 power/clock control bit.	1

Table 46. Power Control for Peripherals register (PCONP - address 0x400F C0C4) bit description

Bit	Symbol	Description	Reset value
27	PCI2S	I2S interface power/clock control bit.	0
28	-	Reserved.	NA
29	PCGPDMA	GPDMA function power/clock control bit.	0
30	PCENET	Ethernet block power/clock control bit.	0
31	PCUSB	USB interface power/clock control bit.	0

Note that the DAC peripheral does not have a control bit in PCONP. To enable the DAC, its output must be selected to appear on the related pin, P0.26, by configuring the PINSEL1 register. See Section 8.5.2 “Pin Function Select Register 1 (PINSEL1 - 0x4002 C004)”.

4.8.10 Power control usage notes

After every reset, the PCONP register contains the value that enables selected interfaces and peripherals controlled by the PCONP to be enabled. Therefore, apart from proper configuring via peripheral dedicated registers, the user’s application might have to access the PCONP in order to start using some of the on-board peripherals.

Power saving oriented systems should have 1s in the PCONP register only in positions that match peripherals really used in the application. All other bits, declared to be "Reserved" or dedicated to the peripherals not used in the current application, must be cleared to 0.

4.8.11 Power domains

The LPC176x/5x provides two independent power domains that allow the bulk of the device to have power removed while maintaining operation of the Real Time Clock.

The VBAT pin supplies power only to the RTC domain. The RTC requires a minimum of power to operate, which can be supplied by an external battery. Whenever the device core power is present, that power is used to operate the RTC, causing no power drain from a battery when main power is available.

4.9 Wake-up timer

The LPC176x/5x begins operation at power-up and when awakened from Power-down mode by using the 4 MHz IRC oscillator as the clock source. This allows chip operation to begin quickly. If the main oscillator or one or both PLLs are needed by the application, software will need to enable these features and wait for them to stabilize before they are used as a clock source.

When the main oscillator is initially activated, the wake-up timer allows software to ensure that the main oscillator is fully functional before the processor uses it as a clock source and starts to execute instructions. This is important at power-on, all types of Reset, and whenever any of the aforementioned functions are turned off for any reason. Since the oscillator and other functions are turned off during Power-down mode, any wake-up of the processor from Power-down mode makes use of the Wake-up Timer.

The Wake-up Timer monitors the crystal oscillator as the means of checking whether it is safe to begin code execution. When power is applied to the chip, or some event caused the chip to exit Power-down mode, some time is required for the oscillator to produce a signal of sufficient amplitude to drive the clock logic. The amount of time depends on many factors, including the rate of $V_{DD(REG)(3V3)}$ ramp (in the case of power on), the type of crystal and its electrical characteristics (if a quartz crystal is used), as well as any other external circuitry (e.g. capacitors), and the characteristics of the oscillator itself under the existing ambient conditions.

Once a clock is detected, the Wake-up Timer counts a fixed number of clocks (4,096), then sets the flag (OSCSTAT bit in the SCS register) that indicates that the main oscillator is ready for use. Software can then switch to the main oscillator and start any required PLLs. Refer to the Main Oscillator description in this chapter for details.

4.10 External clock output pin

For system test and development purposes, any one of several internal clocks may be brought out on the CLKOUT function available on the P1.27 pin, as shown in Figure 12.

Clocks that may be observed via CLKOUT are the CPU clock (cclk), the main oscillator (osc_clk), the internal RC oscillator (irc_osc), the USB clock (usb_clk), and the RTC clock (rtc_clk).

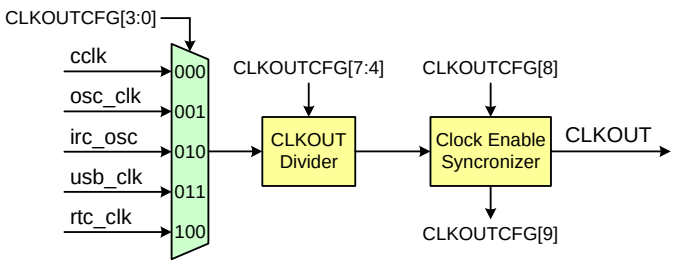

Fig 12. CLKOUT selection

4.10.1 Clock Output Configuration register (CLKOUTCFG - 0x400F C1C8)

The CLKOUTCFG register controls the selection of the internal clock that appears on the CLKOUT pin and allows dividing the clock by an integer value up to 16. The divider can be used to produce a system clock that is related to one of the on-chip clocks. For most clock sources, the division may be by 1. When the CPU clock is selected and is higher than approximately 50 MHz, the output must be divided in order to bring the frequency within the ability of the pin to switch with reasonable logic levels.

Note: The CLKOUT multiplexer is designed to switch cleanly, without glitches, between the possible clock sources. The divider is also designed to allow changing the divide value without glitches.

Table 47. Clock Output Configuration register (CLKOUTCFG - 0x400F C1C8) bit description

Bit	Symbol	Value	Description	Reset value
3:0	CLKOUTSEL		Selects the clock source for the CLKOUT function.	0
		0000	Selects the CPU clock as the CLKOUT source.	
		0001	Selects the main oscillator as the CLKOUT source.	
		0010	Selects the Internal RC oscillator as the CLKOUT source.	
		0011	Selects the USB clock as the CLKOUT source.	
		0100	Selects the RTC oscillator as the CLKOUT source.	
		others	Reserved, do not use these settings.	

Table 47. Clock Output Configuration register (CLKOUTCFG - 0x400F C1C8) bit description

Bit	Symbol	Value	Description	Reset value
7:4	CLKOUTDIV		Integer value to divide the output clock by, minus one.	0
		0000	Clock is divided by 1.	
		0001	Clock is divided by 2.	
		0010	Clock is divided by 3.	
		...	...	
		1111	Clock is divided by 16.	
8	CLKOUT_EN		CLKOUT enable control, allows switching the CLKOUT source without glitches. Clear to stop CLKOUT on the next falling edge. Set to enable CLKOUT.	0
9	CLKOUT_ACT		CLKOUT activity indication. Reads as 1 when CLKOUT is enabled. Read as 0 when CLKOUT has been disabled via the CLKOUT_EN bit and the clock has completed being stopped.	0
31:10	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

6.1 Features

- Nested Vectored Interrupt Controller that is an integral part of the ARM Cortex-M3
- Tightly coupled interrupt controller provides low interrupt latency
- Controls system exceptions and peripheral interrupts
- In the LPC176x/5x, the NVIC supports 35 vectored interrupts
- 32 programmable interrupt priority levels, with hardware priority level masking
- Relocatable vector table
- Non-Maskable Interrupt
- Software interrupt generation

6.2 Description

The Nested Vectored Interrupt Controller (NVIC) is an integral part of the Cortex-M3. The tight coupling to the CPU allows for low interrupt latency and efficient processing of late arriving interrupts.

Refer to the Cortex-M3 User Guide Section 34.4.2 for details of NVIC operation.

6.3 Interrupt sources

Table 50 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, as noted.

Exception numbers relate to where entries are stored in the exception vector table. Interrupt numbers are used in some other contexts, such as software interrupts.

In addition, the NVIC handles the Non-Maskable Interrupt (NMI). In order for NMI to operate from an external signal, the NMI function must be connected to the related device pin (P2.10 / EINT0n / NMI). When connected, a logic 1 on the pin will cause the NMI to be processed. For details, refer to the Cortex-M3 User Guide that is an appendix to this User Manual.

Table 50. Connection of interrupt sources to the Vectored Interrupt Controller

Interrupt ID	Exception Number	Vector Offset	Function	Flag(s)
0	16	0x40	WDT	Watchdog Interrupt (WDINT)
1	17	0x44	Timer 0	Match 0 - 1 (MR0, MR1) Capture 0 - 1 (CR0, CR1)
2	18	0x48	Timer 1	Match 0 - 2 (MR0, MR1, MR2) Capture 0 - 1 (CR0, CR1)
3	19	0x4C	Timer 2	Match 0-3 Capture 0-1
4	20	0x50	Timer 3	Match 0-3 Capture 0-1
5	21	0x54	UART0	Rx Line Status (RLS) Transmit Holding Register Empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI) End of Auto-Baud (ABEO) Auto-Baud Time-Out (ABTO)
6	22	0x58	UART1	Rx Line Status (RLS) Transmit Holding Register Empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI) Modem Control Change End of Auto-Baud (ABEO) Auto-Baud Time-Out (ABTO)
7	23	0x5C	UART 2	Rx Line Status (RLS) Transmit Holding Register Empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI) End of Auto-Baud (ABEO) Auto-Baud Time-Out (ABTO)
8	24	0x60	UART 3	Rx Line Status (RLS) Transmit Holding Register Empty (THRE) Rx Data Available (RDA) Character Time-out Indicator (CTI) End of Auto-Baud (ABEO) Auto-Baud Time-Out (ABTO)
9	25	0x64	PWM1	Match 0 - 6 of PWM1 Capture 0-1 of PWM1
10	26	0x68	I^2C0	SI (state change)
11	27	0x6C	I^2C1	SI (state change)
12	28	0x70	I^2C2	SI (state change)
13	29	0x74	SPI	SPI Interrupt Flag (SPIF) Mode Fault (MODF)

Table 50. Connection of interrupt sources to the Vectored Interrupt Controller

Interrupt ID	Exception Number	Vector Offset	Function	Flag(s)
14	30	0x78	SSP0	Tx FIFO half empty of SSP0 Rx FIFO half full of SSP0 Rx Timeout of SSP0 Rx Overrun of SSP0
15	31	0x7C	SSP 1	Tx FIFO half empty Rx FIFO half full Rx Timeout Rx Overrun
16	32	0x80	PLL0 (Main PLL)	PLL0 Lock (PLOCK0)
17	33	0x84	RTC	Counter Increment (RTCCIF) Alarm (RTCALF)
18	34	0x88	External Interrupt	External Interrupt 0 (EINT0)
19	35	0x8C	External Interrupt	External Interrupt 1 (EINT1)
20	36	0x90	External Interrupt	External Interrupt 2 (EINT2)
21	37	0x94	External Interrupt	External Interrupt 3 (EINT3). **Note:** EINT3 channel is shared with GPIO interrupts
22	38	0x98	ADC	A/D Converter end of conversion
23	39	0x9C	BOD	Brown Out detect
24	40	0xA0	USB	USB_INT_REQ_LP, USB_INT_REQ_HP, USB_INT_REQ_DMA
25	41	0xA4	CAN	CAN Common, CAN 0 Tx, CAN 0 Rx, CAN 1 Tx, CAN 1 Rx
26	42	0xA8	GPDMA	IntStatus of DMA channel 0, IntStatus of DMA channel 1
27	43	0xAC	I^2S	irq, dmareq1, dmareq2
28	44	0xB0	Ethernet	WakeupInt, SoftInt, TxDoneInt, TxFinishedInt, TxErrorInt, TxUnderrunInt, RxDoneInt, RxFinishedInt, RxErrorInt, RxOverrunInt.
29	45	0xB4	Repetitive Interrupt Timer	RITINT
30	46	0xB8	Motor Control PWM	IPER[2:0], IPW[2:0], ICAP[2:0], FES
31	47	0xBC	Quadrature Encoder	INX_Int, TIM_Int, VELC_Int, DIR_Int, ERR_Int, ENCLK_Int, POS0_Int, POS1_Int, POS2_Int, REV_Int, POS0REV_Int, POS1REV_Int, POS2REV_Int
32	48	0xC0	PLL1 (USB PLL)	PLL1 Lock (PLOCK1)
33	49	0xC4	USB Activity Interrupt	USB_NEED_CLK
34	50	0xC8	CAN Activity Interrupt	CAN1WAKE, CAN2WAKE

6.4 Vector table remapping

The Cortex-M3 incorporates a mechanism that allows remapping the interrupt vector table to alternate locations in the memory map. This is controlled via the Vector Table Offset Register (VTOR) contained in the Cortex-M3.

The vector table may be located anywhere within the bottom 1 GB of Cortex-M3 address space. The vector table should be located on a 256 word (1024 byte) boundary to insure alignment on LPC176x/5x family devices. Refer to Section 34.4.3.5 of the Cortex-M3 User Guide appended to this manual for details of the Vector Table Offset feature.

ARM describes bit 29 of the VTOR (TBLOFF) as selecting a memory region, either code or SRAM. For simplicity, this bit can be thought as simply part of the address offset since the split between the “code” space and the “SRAM” space occurs at the location corresponding to bit 29 in a memory address.

Examples:

To place the vector table at the beginning of the “local” static RAM, starting at address 0x1000 0000, place the value 0x1000 0000 in the VTOR register. This indicates address 0x1000 0000 in the code space, since bit 29 of the VTOR equals 0.

To place the vector table at the beginning of the AHB static RAM, starting at address 0x2007 C000, place the value 0x2007 C000 in the VTOR register. This indicates address 0x2007 C000 in the SRAM space, since bit 29 of the VTOR equals 1.

6.5 Register description

The following table summarizes the registers in the NVIC as implemented in the LPC176x/5x. The Cortex-M3 User Guide Section 34.4.2 provides a functional description of the NVIC.

Table 51. NVIC register map

Name	Description	Access	Reset value	Address
ISER0 to ISER1	Interrupt Set-Enable Registers. These 2 registers allow enabling interrupts and reading back the interrupt enables for specific peripheral functions.	RW	0	ISER0 - 0xE000 E100 ISER1 - 0xE000 E104
ICER0 to ICER1	Interrupt Clear-Enable Registers. These 2 registers allow disabling interrupts and reading back the interrupt enables for specific peripheral functions.	RW	0	ICER0 - 0xE000 E180 ICER1 - 0xE000 E184
ISPR0 to ISPR1	Interrupt Set-Pending Registers. These 2 registers allow changing the interrupt state to pending and reading back the interrupt pending state for specific peripheral functions.	RW	0	ISPR0 - 0xE000 E200 ISPR1 - 0xE000 E204
ICPR0 to ICPR1	Interrupt Clear-Pending Registers. These 2 registers allow changing the interrupt state to not pending and reading back the interrupt pending state for specific peripheral functions.	RW	0	ICPR0 - 0xE000 E280 ICPR1 - 0xE000 E284
IABR0 to IABR1	Interrupt Active Bit Registers. These 2 registers allow reading the current interrupt active state for specific peripheral functions.	RO	0	IABR0 - 0xE000 E300 IABR1 - 0xE000 E304
IPR0 to IPR8	Interrupt Priority Registers. These 9 registers allow assigning a priority to each interrupt. Each register contains the 5-bit priority fields for 4 interrupts.	RW	0	IPR0 - 0xE000 E400 IPR1 - 0xE000 E404 IPR2 - 0xE000 E408 IPR3 - 0xE000 E40C IPR4 - 0xE000 E410 IPR5 - 0xE000 E414 IPR6 - 0xE000 E418 IPR7 - 0xE000 E41C IPR8 - 0xE000 E420
STIR	Software Trigger Interrupt Register. This register allows software to generate an interrupt.	WO	0	STIR - 0xE000 EF00

6.5.1 Interrupt Set-Enable Register 0 register (ISER0 - 0xE000 E100)

The ISER0 register allows enabling the first 32 peripheral interrupts, or for reading the enabled state of those interrupts. The remaining interrupts are enabled via the ISER1 register (Section 6.5.2). Disabling interrupts is done through the ICER0 and ICER1 registers (Section 6.5.3 and Section 6.5.4).

Table 52. Interrupt Set-Enable Register 0 register (ISER0 - 0xE000 E100)

Bit	Name	Function
0	ISE_WDT	Watchdog Timer Interrupt Enable. 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.
1	ISE_TIMER0	Timer 0 Interrupt Enable. See functional description for bit 0.
2	ISE_TIMER1	Timer 1. Interrupt Enable. See functional description for bit 0.
3	ISE_TIMER2	Timer 2 Interrupt Enable. See functional description for bit 0.
4	ISE_TIMER3	Timer 3 Interrupt Enable. See functional description for bit 0.
5	ISE_UART0	UART0 Interrupt Enable. See functional description for bit 0.
6	ISE_UART1	UART1 Interrupt Enable. See functional description for bit 0.
7	ISE_UART2	UART2 Interrupt Enable. See functional description for bit 0.
8	ISE_UART3	UART3 Interrupt Enable. See functional description for bit 0.
9	ISE_PWM	PWM1 Interrupt Enable. See functional description for bit 0.
10	ISE_I2C0	I^2C0 Interrupt Enable. See functional description for bit 0.
11	ISE_I2C1	I^2C1 Interrupt Enable. See functional description for bit 0.
12	ISE_I2C2	I^2C2 Interrupt Enable. See functional description for bit 0.
13	ISE_SPI	SPI Interrupt Enable. See functional description for bit 0.
14	ISE_SSP0	SSP0 Interrupt Enable. See functional description for bit 0.
15	ISE_SSP1	SSP1 Interrupt Enable. See functional description for bit 0.
16	ISE_PLL0	PLL0 (Main PLL) Interrupt Enable. See functional description for bit 0.
17	ISE_RTC	Real Time Clock (RTC) Interrupt Enable. See functional description for bit 0.
18	ISE_EINT0	External Interrupt 0 Interrupt Enable. See functional description for bit 0.
19	ISE_EINT1	External Interrupt 1 Interrupt Enable. See functional description for bit 0.
20	ISE_EINT2	External Interrupt 2 Interrupt Enable. See functional description for bit 0.
21	ISE_EINT3	External Interrupt 3 Interrupt Enable. See functional description for bit 0.
22	ISE_ADC	ADC Interrupt Enable. See functional description for bit 0.
23	ISE_BOD	BOD Interrupt Enable. See functional description for bit 0.
24	ISE_USB	USB Interrupt Enable. See functional description for bit 0.
25	ISE_CAN	CAN Interrupt Enable. See functional description for bit 0.
26	ISE_DMA	GPDMA Interrupt Enable. See functional description for bit 0.
27	ISE_I2S	I^2S Interrupt Enable. See functional description for bit 0.
28	ISE_ENET	Ethernet Interrupt Enable. See functional description for bit 0.
29	ISE_RIT	Repetitive Interrupt Timer Interrupt Enable. See functional description for bit 0.
30	ISE_MCPWM	Motor Control PWM Interrupt Enable. See functional description for bit 0.
31	ISE_QEI	Quadrature Encoder Interface Interrupt Enable. See functional description for bit 0.

6.5.2 Interrupt Set-Enable Register 1 register (ISER1 - 0xE000 E104)

The ISER1 register allows enabling the second group of peripheral interrupts, or for reading the enabled state of those interrupts. Disabling interrupts is done through the ICER0 and ICER1 registers (Section 6.5.3 and Section 6.5.4).

Table 53. Interrupt Set-Enable Register 1 register (ISER1 - 0xE000 E104)

Bit	Name	Function
0	ISE_PLL1	PLL1 (USB PLL) Interrupt Enable. 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.
1	ISE_USBACT	USB Activity Interrupt Enable. See functional description for bit 0.
2	ISE_CANACT	CAN Activity Interrupt Enable. See functional description for bit 0.
31:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

6.5.3 Interrupt Clear-Enable Register 0 (ICER0 - 0xE000 E180)

The ICER0 register allows disabling the first 32 peripheral interrupts, or for reading the enabled state of those interrupts. The remaining interrupts are disabled via the ICER1 register (Section 6.5.4). Enabling interrupts is done through the ISER0 and ISER1 registers (Section 6.5.1 and Section 6.5.2).

Table 54. Interrupt Clear-Enable Register 0 (ICER0 - 0xE000 E180)

Bit	Name	Function
0	ICE_WDT	Watchdog Timer Interrupt Disable. 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.
1	ICE_TIMER0	Timer 0 Interrupt Disable. See functional description for bit 0.
2	ICE_TIMER1	Timer 1. Interrupt Disable. See functional description for bit 0.
3	ICE_TIMER2	Timer 2 Interrupt Disable. See functional description for bit 0.
4	ICE_TIMER3	Timer 3 Interrupt Disable. See functional description for bit 0.
5	ICE_UART0	UART0 Interrupt Disable. See functional description for bit 0.
6	ICE_UART1	UART1 Interrupt Disable. See functional description for bit 0.
7	ICE_UART2	UART2 Interrupt Disable. See functional description for bit 0.
8	ICE_UART3	UART3 Interrupt Disable. See functional description for bit 0.
9	ICE_PWM	PWM1 Interrupt Disable. See functional description for bit 0.
10	ICE_I2C0	I^2C0 Interrupt Disable. See functional description for bit 0.
11	ICE_I2C1	I^2C1 Interrupt Disable. See functional description for bit 0.
12	ICE_I2C2	I^2C2 Interrupt Disable. See functional description for bit 0.
13	ICE_SPI	SPI Interrupt Disable. See functional description for bit 0.
14	ICE_SSP0	SSP0 Interrupt Disable. See functional description for bit 0.
15	ICE_SSP1	SSP1 Interrupt Disable. See functional description for bit 0.
16	ICE_PLL0	PLL0 (Main PLL) Interrupt Disable. See functional description for bit 0.
17	ICE_RTC	Real Time Clock (RTC) Interrupt Disable. See functional description for bit 0.
18	ICE_EINT0	External Interrupt 0 Interrupt Disable. See functional description for bit 0.
19	ICE_EINT1	External Interrupt 1 Interrupt Disable. See functional description for bit 0.
20	ICE_EINT2	External Interrupt 2 Interrupt Disable. See functional description for bit 0.
21	ICE_EINT3	External Interrupt 3 Interrupt Disable. See functional description for bit 0.
22	ICE_ADC	ADC Interrupt Disable. See functional description for bit 0.
23	ICE_BOD	BOD Interrupt Disable. See functional description for bit 0.
24	ICE_USB	USB Interrupt Disable. See functional description for bit 0.
25	ICE_CAN	CAN Interrupt Disable. See functional description for bit 0.
26	ICE_DMA	GPDMA Interrupt Disable. See functional description for bit 0.
27	ICE_I2S	I^2S Interrupt Disable. See functional description for bit 0.
28	ICE_ENET	Ethernet Interrupt Disable. See functional description for bit 0.
29	ICE_RIT	Repetitive Interrupt Timer Interrupt Disable. See functional description for bit 0.
30	ICE_MCPWM	Motor Control PWM Interrupt Disable. See functional description for bit 0.
31	ICE_QEI	Quadrature Encoder Interface Interrupt Disable. See functional description for bit 0.

6.5.4 Interrupt Clear-Enable Register 1 register (ICER1 - 0xE000 E184)

The ICER1 register allows disabling the second group of peripheral interrupts, or for reading the enabled state of those interrupts. Enabling interrupts is done through the ISER0 and ISER1 registers (Section 6.5.1 and Section 6.5.2).

Table 55. Interrupt Clear-Enable Register 1 register (ICER1 - 0xE000 E184)

Bit	Name	Function
0	ICE_PLL1	PLL1 (USB PLL) Interrupt Disable. 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.
1	ICE_USBACT	USB Activity Interrupt Disable. See functional description for bit 0.
2	ICE_CANACT	CAN Activity Interrupt Disable. See functional description for bit 0.
31:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

6.5.5 Interrupt Set-Pending Register 0 register (ISPR0 - 0xE000 E200)

The ISPR0 register allows setting the pending state of the first 32 peripheral interrupts, or for reading the pending state of those interrupts. The remaining interrupts can have their pending state set via the ISPR1 register (Section 6.5.6). Clearing the pending state of interrupts is done through the ICPR0 and ICPR1 registers (Section 6.5.7 and Section 6.5.8).

Table 56. Interrupt Set-Pending Register 0 register (ISPR0 - 0xE000 E200)

Bit	Name	Function
0	ISP_WDT	Watchdog Timer Interrupt Pending set. 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.
1	ISP_TIMER0	Timer 0 Interrupt Pending set. See functional description for bit 0.
2	ISP_TIMER1	Timer 1. Interrupt Pending set. See functional description for bit 0.
3	ISP_TIMER2	Timer 2 Interrupt Pending set. See functional description for bit 0.
4	ISP_TIMER3	Timer 3 Interrupt Pending set. See functional description for bit 0.
5	ISP_UART0	UART0 Interrupt Pending set. See functional description for bit 0.
6	ISP_UART1	UART1 Interrupt Pending set. See functional description for bit 0.
7	ISP_UART2	UART2 Interrupt Pending set. See functional description for bit 0.
8	ISP_UART3	UART3 Interrupt Pending set. See functional description for bit 0.
9	ISP_PWM	PWM1 Interrupt Pending set. See functional description for bit 0.
10	ISP_I2C0	I^2C0 Interrupt Pending set. See functional description for bit 0.
11	ISP_I2C1	I^2C1 Interrupt Pending set. See functional description for bit 0.
12	ISP_I2C2	I^2C2 Interrupt Pending set. See functional description for bit 0.
13	ISP_SPI	SPI Interrupt Pending set. See functional description for bit 0.
14	ISP_SSP0	SSP0 Interrupt Pending set. See functional description for bit 0.
15	ISP_SSP1	SSP1 Interrupt Pending set. See functional description for bit 0.
16	ISP_PLL0	PLL0 (Main PLL) Interrupt Pending set. See functional description for bit 0.
17	ISP_RTC	Real Time Clock (RTC) Interrupt Pending set. See functional description for bit 0.
18	ISP_EINT0	External Interrupt 0 Interrupt Pending set. See functional description for bit 0.
19	ISP_EINT1	External Interrupt 1 Interrupt Pending set. See functional description for bit 0.
20	ISP_EINT2	External Interrupt 2 Interrupt Pending set. See functional description for bit 0.
21	ISP_EINT3	External Interrupt 3 Interrupt Pending set. See functional description for bit 0.
22	ISP_ADC	ADC Interrupt Pending set. See functional description for bit 0.
23	ISP_BOD	BOD Interrupt Pending set. See functional description for bit 0.
24	ISP_USB	USB Interrupt Pending set. See functional description for bit 0.
25	ISP_CAN	CAN Interrupt Pending set. See functional description for bit 0.
26	ISP_DMA	GPDMA Interrupt Pending set. See functional description for bit 0.
27	ISP_I2S	I^2S Interrupt Pending set. See functional description for bit 0.
28	ISP_ENET	Ethernet Interrupt Pending set. See functional description for bit 0.
29	ISP_RIT	Repetitive Interrupt Timer Interrupt Pending set. See functional description for bit 0.
30	ISP_MCPWM	Motor Control PWM Interrupt Pending set. See functional description for bit 0.
31	ISP_QEI	Quadrature Encoder Interface Interrupt Pending set. See functional description for bit 0.

6.5.6 Interrupt Set-Pending Register 1 register (ISPR1 - 0xE000 E204)

The ISPR1 register allows setting the pending state of the second group of peripheral interrupts, or for reading the pending state of those interrupts. Clearing the pending state of interrupts is done through the ICPR0 and ICPR1 registers (Section 6.5.7 and Section 6.5.8).

Table 57. Interrupt Set-Pending Register 1 register (ISPR1 - 0xE000 E204)

Bit	Name	Function
0	ISP_PLL1	PLL1 (USB PLL) Interrupt Pending set. 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.
1	ISP_USBACT	USB Activity Interrupt Pending set. See functional description for bit 0.
2	ISP_CANACT	CAN Activity Interrupt Pending set. See functional description for bit 0.
31:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

6.5.7 Interrupt Clear-Pending Register 0 register (ICPR0 - 0xE000 E280)

The ICPR0 register allows clearing the pending state of the first 32 peripheral interrupts, or for reading the pending state of those interrupts. The remaining interrupts can have their pending state cleared via the ICPR1 register (Section 6.5.8). Setting the pending state of interrupts is done through the ISPR0 and ISPR1 registers (Section 6.5.5 and Section 6.5.6).

Table 58. Interrupt Clear-Pending Register 0 register (ICPR0 - 0xE000 E280)

Bit	Name	Function
0	ICP_WDT	Watchdog Timer Interrupt Pending clear. 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.
1	ICP_TIMER0	Timer 0 Interrupt Pending clear. See functional description for bit 0.
2	ICP_TIMER1	Timer 1. Interrupt Pending clear. See functional description for bit 0.
3	ICP_TIMER2	Timer 2 Interrupt Pending clear. See functional description for bit 0.
4	ICP_TIMER3	Timer 3 Interrupt Pending clear. See functional description for bit 0.
5	ICP_UART0	UART0 Interrupt Pending clear. See functional description for bit 0.
6	ICP_UART1	UART1 Interrupt Pending clear. See functional description for bit 0.
7	ICP_UART2	UART2 Interrupt Pending clear. See functional description for bit 0.
8	ICP_UART3	UART3 Interrupt Pending clear. See functional description for bit 0.
9	ICP_PWM	PWM1 Interrupt Pending clear. See functional description for bit 0.
10	ICP_I2C0	I^2C0 Interrupt Pending clear. See functional description for bit 0.
11	ICP_I2C1	I^2C1 Interrupt Pending clear. See functional description for bit 0.
12	ICP_I2C2	I^2C2 Interrupt Pending clear. See functional description for bit 0.
13	ICP_SPI	SPI Interrupt Pending clear. See functional description for bit 0.
14	ICP_SSP0	SSP0 Interrupt Pending clear. See functional description for bit 0.
15	ICP_SSP1	SSP1 Interrupt Pending clear. See functional description for bit 0.
16	ICP_PLL0	PLL0 (Main PLL) Interrupt Pending clear. See functional description for bit 0.
17	ICP_RTC	Real Time Clock (RTC) Interrupt Pending clear. See functional description for bit 0.
18	ICP_EINT0	External Interrupt 0 Interrupt Pending clear. See functional description for bit 0.
19	ICP_EINT1	External Interrupt 1 Interrupt Pending clear. See functional description for bit 0.
20	ICP_EINT2	External Interrupt 2 Interrupt Pending clear. See functional description for bit 0.
21	ICP_EINT3	External Interrupt 3 Interrupt Pending clear. See functional description for bit 0.
22	ICP_ADC	ADC Interrupt Pending clear. See functional description for bit 0.
23	ICP_BOD	BOD Interrupt Pending clear. See functional description for bit 0.
24	ICP_USB	USB Interrupt Pending clear. See functional description for bit 0.
25	ICP_CAN	CAN Interrupt Pending clear. See functional description for bit 0.
26	ICP_DMA	GPDMA Interrupt Pending clear. See functional description for bit 0.
27	ICP_I2S	I^2S Interrupt Pending clear. See functional description for bit 0.
28	ICP_ENET	Ethernet Interrupt Pending clear. See functional description for bit 0.
29	ICP_RIT	Repetitive Interrupt Timer Interrupt Pending clear. See functional description for bit 0.
30	ICP_MCPWM	Motor Control PWM Interrupt Pending clear. See functional description for bit 0.
31	ICP_QEI	Quadrature Encoder Interface Interrupt Pending clear. See functional description for bit 0.

6.5.8 Interrupt Clear-Pending Register 1 register (ICPR1 - 0xE000 E284)

The ICPR1 register allows clearing the pending state of the second group of peripheral interrupts, or for reading the pending state of those interrupts. Setting the pending state of interrupts is done through the ISPR0 and ISPR1 registers (Section 6.5.5 and Section 6.5.6).

Table 59. Interrupt Set-Pending Register 1 register (ISPR1 - 0xE000 E204)

Bit	Name	Function
0	ICP_PLL1	PLL1 (USB PLL) Interrupt Pending clear. 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.
1	ICP_USBACT	USB Activity Interrupt Pending clear. See functional description for bit 0.
2	ICP_CANACT	CAN Activity Interrupt Pending clear. See functional description for bit 0.
31:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

6.5.9 Interrupt Active Bit Register 0 (IABR0 - 0xE000 E300)

The IABR0 register is a read-only register that allows reading the active state of the first 32 peripheral interrupts. This allows determining which peripherals are asserting an interrupt to the NVIC, and may also be pending if they are enabled. The remaining interrupts can have their active state read via the IABR1 register (Section 6.5.10).

Table 60. Interrupt Active Bit Register 0 (IABR0 - 0xE000 E300)

Bit	Name	Function
0	IAB_WDT	Watchdog Timer Interrupt Active. Read: 0 indicates that the interrupt is not active, 1 indicates that the interrupt is active.
1	IAB_TIMER0	Timer 0 Interrupt Active. See functional description for bit 0.
2	IAB_TIMER1	Timer 1. Interrupt Active. See functional description for bit 0.
3	IAB_TIMER2	Timer 2 Interrupt Active. See functional description for bit 0.
4	IAB_TIMER3	Timer 3 Interrupt Active. See functional description for bit 0.
5	IAB_UART0	UART0 Interrupt Active. See functional description for bit 0.
6	IAB_UART1	UART1 Interrupt Active. See functional description for bit 0.
7	IAB_UART2	UART2 Interrupt Active. See functional description for bit 0.
8	IAB_UART3	UART3 Interrupt Active. See functional description for bit 0.
9	IAB_PWM	PWM1 Interrupt Active. See functional description for bit 0.
10	IAB_I2C0	I^2C0 Interrupt Active. See functional description for bit 0.
11	IAB_I2C1	I^2C1 Interrupt Active. See functional description for bit 0.
12	IAB_I2C2	I^2C2 Interrupt Active. See functional description for bit 0.
13	IAB_SPI	SPI Interrupt Active. See functional description for bit 0.
14	IAB_SSP0	SSP0 Interrupt Active. See functional description for bit 0.
15	IAB_SSP1	SSP1 Interrupt Active. See functional description for bit 0.
16	IAB_PLL0	PLL0 (Main PLL) Interrupt Active. See functional description for bit 0.
17	IAB_RTC	Real Time Clock (RTC) Interrupt Active. See functional description for bit 0.
18	IAB_EINT0	External Interrupt 0 Interrupt Active. See functional description for bit 0.
19	IAB_EINT1	External Interrupt 1 Interrupt Active. See functional description for bit 0.
20	IAB_EINT2	External Interrupt 2 Interrupt Active. See functional description for bit 0.
21	IAB_EINT3	External Interrupt 3 Interrupt Active. See functional description for bit 0.
22	IAB_ADC	ADC Interrupt Active. See functional description for bit 0.
23	IAB_BOD	BOD Interrupt Active. See functional description for bit 0.
24	IAB_USB	USB Interrupt Active. See functional description for bit 0.
25	IAB_CAN	CAN Interrupt Active. See functional description for bit 0.
26	IAB_DMA	GPDMA Interrupt Active. See functional description for bit 0.
27	IAB_I2S	I^2S Interrupt Active. See functional description for bit 0.
28	IAB_ENET	Ethernet Interrupt Active. See functional description for bit 0.
29	IAB_RIT	Repetitive Interrupt Timer Interrupt Active. See functional description for bit 0.
30	IAB_MCPWM	Motor Control PWM Interrupt Active. See functional description for bit 0.
31	IAB_QEI	Quadrature Encoder Interface Interrupt Active. See functional description for bit 0.

6.5.10 Interrupt Active Bit Register 1 (IABR1 - 0xE000 E304)

The IABR1 register is a read-only register that allows reading the active state of the second group of peripheral interrupts. This allows determining which peripherals are asserting an interrupt to the NVIC, and may also be pending if they are enabled.

Table 61. Interrupt Active Bit Register 1 (IABR1 - 0xE000 E304)

Bit	Name	Function
0	IAB_PLL1	PLL1 (USB PLL) Interrupt Active. Read: 0 indicates that the interrupt is not active, 1 indicates that the interrupt is active.
1	IAB_USBACT	USB Activity Interrupt Active. See functional description for bit 0.
2	IAB_CANACT	CAN Activity Interrupt Active. See functional description for bit 0.
31:3	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

6.5.11 Interrupt Priority Register 0 (IPR0 - 0xE000 E400)

The IPR0 register controls the priority of the first 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 62. Interrupt Priority Register 0 (IPR0 - 0xE000 E400)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_WDT	Watchdog Timer Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_TIMER0	Timer 0 Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_TIMER1	Timer 1 Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_TIMER2	Timer 2 Interrupt Priority. See functional description for bits 7-3.

6.5.12 Interrupt Priority Register 1 (IPR1 - 0xE000 E404)

The IPR1 register controls the priority of the second group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 63. Interrupt Priority Register 1 (IPR1 - 0xE000 E404)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_TIMER3	Timer 3 Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_UART0	UART0 Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_UART1	UART1 Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_UART2	UART2 Interrupt Priority. See functional description for bits 7-3.

6.5.13 Interrupt Priority Register 2 (IPR2 - 0xE000 E408)

The IPR2 register controls the priority of the third group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 64. Interrupt Priority Register 2 (IPR2 - 0xE000 E408)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_UART3	UART3 Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_PWM	PWM Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_I2C0	I^2C0 Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_I2C1	I^2C1 Interrupt Priority. See functional description for bits 7-3.

6.5.14 Interrupt Priority Register 3 (IPR3 - 0xE000 E40C)

The IPR3 register controls the priority of the fourth group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 65. Interrupt Priority Register 3 (IPR3 - 0xE000 E40C)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_I2C2	I^2C2 Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_SPI	SPI Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_SSP0	SSP0 Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_SSP1	SSP1 Interrupt Priority. See functional description for bits 7-3.

6.5.15 Interrupt Priority Register 4 (IPR4 - 0xE000 E410)

The IPR4 register controls the priority of the fifth group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 66. Interrupt Priority Register 4 (IPR4 - 0xE000 E410)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_PLL0	PLL0 (Main PLL) Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_RTC	Real Time Clock (RTC) Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_EINT0	External Interrupt 0 Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_EINT1	External Interrupt 1 Interrupt Priority. See functional description for bits 7-3.

6.5.16 Interrupt Priority Register 5 (IPR5 - 0xE000 E414)

The IPR5 register controls the priority of the sixth group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 67. Interrupt Priority Register 5 (IPR5 - 0xE000 E414)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_EINT2	External Interrupt 2 Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_EINT3	External Interrupt 3 Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_ADC	ADC Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_BOD	BOD Interrupt Priority. See functional description for bits 7-3.

6.5.17 Interrupt Priority Register 6 (IPR6 - 0xE000 E418)

The IPR6 register controls the priority of the seventh group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 68. Interrupt Priority Register 6 (IPR6 - 0xE000 E418)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_USB	USB Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_CAN	CAN Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_DMA	GPDMA Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_I2S	I^2S Interrupt Priority. See functional description for bits 7-3.

6.5.18 Interrupt Priority Register 7 (IPR7 - 0xE000 E41C)

The IPR7 register controls the priority of the eighth group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 69. Interrupt Priority Register 7 (IPR7 - 0xE000 E41C)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_ENET	Ethernet Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_RIT	Repetitive Interrupt Timer Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_MCPWM	Motor Control PWM Interrupt Priority. See functional description for bits 7-3.
26:24	Unimplemented	These bits ignore writes, and read as 0.
31:27	IP_QEI	Quadrature Encoder Interface Interrupt Priority. See functional description for bits 7-3.

6.5.19 Interrupt Priority Register 8 (IPR8 - 0xE000 E420)

The IPR8 register controls the priority of the ninth and last group of 4 peripheral interrupts. Each interrupt can have one of 32 priorities, where 0 is the highest priority.

Table 70. Interrupt Priority Register 8 (IPR8 - 0xE000 E420)

Bit	Name	Function
2:0	Unimplemented	These bits ignore writes, and read as 0.
7:3	IP_PLL1	PLL1 (USB PLL) Interrupt Priority. 0 = highest priority. 31 (0x1F) = lowest priority.
10:8	Unimplemented	These bits ignore writes, and read as 0.
15:11	IP_USBACT	USB Activity Interrupt Priority. See functional description for bits 7-3.
18:16	Unimplemented	These bits ignore writes, and read as 0.
23:19	IP_CANACT	CAN Activity Interrupt Priority. See functional description for bits 7-3.
31:24	Unimplemented	These bits ignore writes, and read as 0.

6.5.20 Software Trigger Interrupt Register (STIR - 0xE000 EF00)

The STIR register provides an alternate way for software to generate an interrupt, in addition to using the ISPR registers. This mechanism can only be used to generate peripheral interrupts, not system exceptions.

By default, only privileged software can write to the STIR register. Unprivileged software can be given this ability if privileged software sets the USERSETMPEND bit in the CCR register (see Section 34.4.3.8).

Table 71. Software Trigger Interrupt Register (STIR - 0xE000 EF00)

Bit	Name	Function
8:0	INTID	Writing a value to this field generates an interrupt for the specified the interrupt number (see Table 50). The range allowed for the LPC176x/5x is 0 to 111.
31:9	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

7.1 LPC176x/5x pin configuration

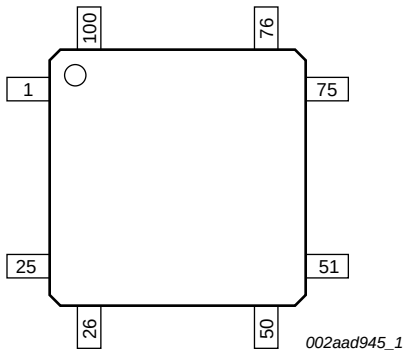

Fig 14. LPC176x LQFP100 pin configuration

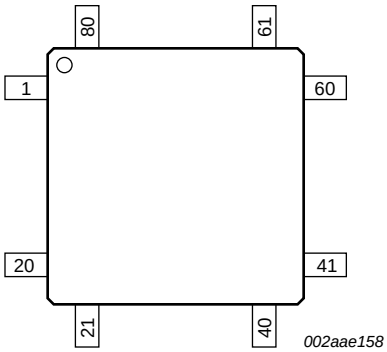

Fig 15. LPC175x LQFP80 pin configuration

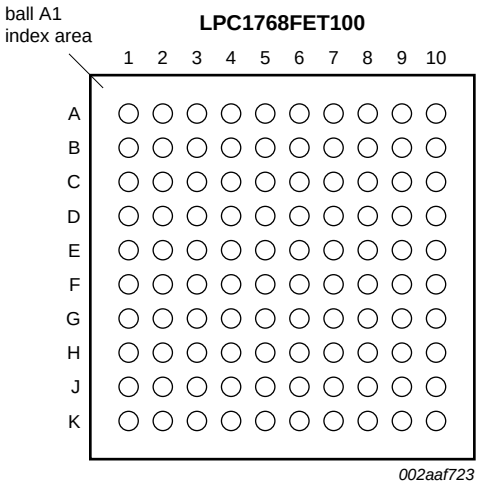

Fig 16. Pin configuration TFBGA100 package

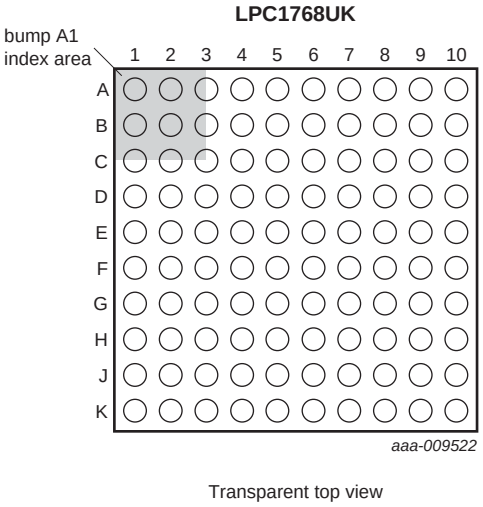

Fig 17. Pin configuration WLCSP100 package

Table 72. Pin allocation table TFBGA100 package

Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
Row A							
1	TDO/SWO	2	P0[3]/RXD0/AD0[6]	3	$V_{DD(3V3)}$	4	P1[4]/ENET_TX_EN
5	P1[10]/ENET_RXD1	6	P1[16]/ENET_MDC	7	$V_{DD(REG)(3V3)}$	8	P0[4]/I2SRX_CLK/ RD2/CAP2[0]
9	P0[7]/I2STX_CLK/ SCK1/MAT2[1]	10	P0[9]/I2STX_SDA/ MOSI1/MAT2[3]	11	-	12	-
Row B							
1	TMS/SWDIO	2	RTCK	3	V_{SS}	4	P1[1]/ENET_TXD1

Table 72. Pin allocation table TFBGA100 package *…continued*

Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
5	P1[9]/ENET_RXD0	6	P1[17]/ ENET_MDIO	7	V_{SS}	8	P0[6]/I2SRX_SDA/ SSEL1/MAT2[0]
9	P2[0]/PWM1[1]/TXD1	10	P2[1]/PWM1[2]/RXD1	11	-	12	-
Row C							
1	TCK/SWDCLK	2	$\overline{\text{TRST}}$	3	TDI	4	P0[2]/TXD0/AD0[7]
5	P1[8]/ENET_CRS	6	P1[15]/ ENET_REF_CLK	7	P4[28]/RX_MCLK/ MAT2[0]/TXD3	8	P0[8]/I2STX_WS/ MISO1/MAT2[2]
9	V_{SS}	10	$V_{DD(3V3)}$	11	-	12	-
Row D							
1	P0[24]/AD0[1]/ I2SRX_WS/CAP3[1]	2	P0[25]/AD0[2]/ I2SRX_SDA/TXD3	3	P0[26]/AD0[3]/ AOUT/RXD3	4	n.c.
5	P1[0]/ENET_TXD0	6	P1[14]/ENET_RX_ER	7	P0[5]/I2SRX_WS/ TD2/CAP2[1]	8	P2[2]/PWM1[3]/ CTS1/TRACEDATA[3]
9	P2[4]/PWM1[5]/ DSR1/TRACEDATA[1]	10	P2[5]/PWM1[6]/ DTR1/TRACEDATA[0]	11	-	12	-
Row E							
1	V_{SSA}	2	V_{DDA}	3	VREFP	4	n.c.
5	P0[23]/AD0[0]/ I2SRX_CLK/CAP3[0]	6	P4[29]/TX_MCLK/ MAT2[1]/RXD3	7	P2[3]/PWM1[4]/ DCD1/TRACEDATA[2]	8	P2[6]/PCAP1[0]/ RI1/TRACECLK
9	P2[7]/RD2/RTS1	10	P2[8]/TD2/TXD2	11	-	12	-
Row F							
1	VREFN	2	RTCX1	3	$\overline{\text{RESET}}$	4	P1[31]/SCK1/ AD0[5]
5	P1[21]/$\overline{\text{MCABORT}}$/ PWM1[3]/SSEL0	6	P0[18]/DCD1/ MOSI0/MOSI	7	P2[9]/USB_CONNECT/ RXD2	8	P0[16]/RXD1/ SSEL0/SSEL
9	P0[17]/CTS1/ MISO0/MISO	10	P0[15]/TXD1/ SCK0/SCK	11	-	12	-
Row G							
1	RTCX2	2	VBAT	3	XTAL2	4	P0[30]/USB_D–
5	P1[25]/MCOA1/ MAT1[1]	6	P1[29]/MCOB2/ PCAP1[1]/MAT0[1]	7	V_{SS}	8	P0[21]/RI1/RD1
9	P0[20]/DTR1/SCL1	10	P0[19]/DSR1/SDA1	11	-	12	-
Row H							
1	P1[30]/V_{BUS}/ AD0[4]	2	XTAL1	3	P3[25]/MAT0[0]/ PWM1[2]	4	P1[18]/USB_UP_LED/ PWM1[1]/CAP1[0]
5	P1[24]/MCI2/ PWM1[5]/MOSI0	6	$V_{DD(REG)(3V3)}$	7	P0[10]/TXD2/ SDA2/MAT3[0]	8	P2[11]/$\overline{\text{EINT1}}$/ I2STX_CLK
9	$V_{DD(3V3)}$	10	P0[22]/RTS1/TD1	11	-	12	-

Table 72. Pin allocation table TFBGA100 package *...continued*

Pin	Symbol	Pin	Symbol	Pin	Symbol	Pin	Symbol
Row J							
1	P0[28]/SCL0/ USB_SCL	2	P0[27]/SDA0/ USB_SDA	3	P0[29]/USB_D+	4	P1[19]/MCOA0/ USB_PPWR/ CAP1[1]
5	P1[22]/MCOB0/ USB_PWRD/ MAT1[0]	6	V_{SS}	7	P1[28]/MCOA2/ PCAP1[0]/ MAT0[0]	8	P0[1]/TD1/RXD3/SCL1
9	P2[13]/EINT3/ I2STX_SDA	10	P2[10]/EINT0/NMI	11	-	12	-
Row K							
1	P3[26]/STCLK/ MAT0[1]/PWM1[3]	2	$V_{DD(3V3)}$	3	V_{SS}	4	P1[20]/MCI0/ PWM1[2]/SCK0
5	P1[23]/MCI1/ PWM1[4]/MISO0	6	P1[26]/MCOB1/ PWM1[6]/CAP0[0]	7	P1[27]/CLKOUT /USB_OVRCR/ CAP0[1]	8	P0[0]/RD1/TXD3/SDA1
9	P0[11]/RXD2/ SCL2/MAT3[1]	10	P2[12]/EINT2/ I2STX_WS	11	-	12	-

7.1.1 LPC176x/5x pin description

I/O pins on the LPC176x/5x are 5V tolerant and have input hysteresis unless indicated in the table below. Crystal pins, power pins, and reference voltage pins are not 5V tolerant. In addition, when pins are selected to be A to D converter inputs, they are no longer 5V tolerant and must be limited to the voltage at the ADC positive reference pin (V_{REFP}).

Table 73. Pin description (LPC175x)

Symbol	Pin	Type	Description
P0[0] to P0[31]		I/O	**Port 0:** Port 0 is a 32-bit I/O port with individual direction controls for each bit. The operation of Port 0 pins depends upon the pin function selected via the pin connect block. Some port pins are not available on the LQFP80 package.
P0[0]/RD1/TXD3/ SDA1	37[1]	I/O	**P0[0] —** General purpose digital input/output pin.
		I	**RD1 —** CAN1 receiver input.
		O	**TXD3 —** Transmitter output for UART3.
		I/O	**SDA1 —** I^2C1 data input/output (this is not an I^2C-bus compliant open-drain pin).
P0[1]/TD1/RXD3/ SCL1	38[1]	I/O	**P0[1] —** General purpose digital input/output pin.
		O	**TD1 —** CAN1 transmitter output.
		I	**RXD3 —** Receiver input for UART3.
		I/O	**SCL1 —** I^2C1 clock input/output (this is not an I^2C-bus compliant open-drain pin).
P0[2]/TXD0/AD0[7]	79[2]	I/O	**P0[2] —** General purpose digital input/output pin.
		O	**TXD0 —** Transmitter output for UART0.
		I	**AD0[7] —** A/D converter 0, input 7.
P0[3]/RXD0/AD0[6]	80[2]	I/O	**P0[3] —** General purpose digital input/output pin.
		I	**RXD0 —** Receiver input for UART0.
		I	**AD0[6] —** A/D converter 0, input 6.

Table 73. Pin description (LPC175x) *…continued*

Symbol	Pin	Type	Description
P0[6]/ I2SRX_SDA/ SSEL1/MAT2[0]	64[1]	I/O	**P0[6] —** General purpose digital input/output pin.
		I/O	**I2SRX_SDA —** Receive data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I²S-bus specification*. (LPC1759/58/56 only).
		I/O	**SSEL1 —** Slave Select for SSP1.
		O	**MAT2[0] —** Match output for Timer 2, channel 0.
P0[7]/I2STX_CLK/ SCK1/MAT2[1]	63[1]	I/O	**P0[7] —** General purpose digital input/output pin.
		I/O	**I2STX_CLK —** Transmit Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the *I²S-bus specification*. (LPC1759/58/56 only).
		I/O	**SCK1 —** Serial Clock for SSP1.
		O	**MAT2[1] —** Match output for Timer 2, channel 1.
P0[8]/I2STX_WS/ MISO1/MAT2[2]	62[1]	I/O	**P0[8] —** General purpose digital input/output pin.
		I/O	**I2STX_WS —** Transmit Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the *I²S-bus specification*. (LPC1759/58/56 only).
		I/O	**MISO1 —** Master In Slave Out for SSP1.
		O	**MAT2[2] —** Match output for Timer 2, channel 2.
P0[9]/I2STX_SDA/ MOSI1/MAT2[3]	61[1]	I/O	**P0[9] —** General purpose digital input/output pin.
		I/O	**I2STX_SDA —** Transmit data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I²S-bus specification*. (LPC1759/58/56 only).
		I/O	**MOSI1 —** Master Out Slave In for SSP1.
		O	**MAT2[3] —** Match output for Timer 2, channel 3.
P0[10]/TXD2/ SDA2/MAT3[0]	39[1]	I/O	**P0[10] —** General purpose digital input/output pin.
		O	**TXD2 —** Transmitter output for UART2.
		I/O	**SDA2 —** I²C2 data input/output (this is not an open-drain pin).
		O	**MAT3[0] —** Match output for Timer 3, channel 0.
P0[11]/RXD2/ SCL2/MAT3[1]	40[1]	I/O	**P0[11] —** General purpose digital input/output pin.
		I	**RXD2 —** Receiver input for UART2.
		I/O	**SCL2 —** I²C2 clock input/output (this is not an open-drain pin).
		O	**MAT3[1] —** Match output for Timer 3, channel 1.
P0[15]/TXD1/ SCK0/SCK	47[1]	I/O	**P0[15] —** General purpose digital input/output pin.
		O	**TXD1 —** Transmitter output for UART1.
		I/O	**SCK0 —** Serial clock for SSP0.
		I/O	**SCK —** Serial clock for SPI.
P0[16]/RXD1/ SSEL0/SSEL	48[1]	I/O	**P0[16] —** General purpose digital input/output pin.
		I	**RXD1 —** Receiver input for UART1.
		I/O	**SSEL0 —** Slave Select for SSP0.
		I/O	**SSEL —** Slave Select for SPI.

Table 73. Pin description (LPC175x) *…continued*

Symbol	Pin	Type	Description
P0[17]/CTS1/ MISO0/MISO	46[1]	I/O	**P0[17] —** General purpose digital input/output pin.
		I	**CTS1 —** Clear to Send input for UART1.
		I/O	**MISO0 —** Master In Slave Out for SSP0.
		I/O	**MISO —** Master In Slave Out for SPI.
P0[18]/DCD1/ MOSI0/MOSI	45[1]	I/O	**P0[18] —** General purpose digital input/output pin.
		I	**DCD1 —** Data Carrier Detect input for UART1.
		I/O	**MOSI0 —** Master Out Slave In for SSP0.
		I/O	**MOSI —** Master Out Slave In for SPI.
P0[22]/RTS1/TD1	44[1]	I/O	**P0[22] —** General purpose digital input/output pin.
		O	**RTS1 —** Request to Send output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
		O	**TD1 —** CAN1 transmitter output.
P0[25]/AD0[2]/ I2SRX _SDA/ TXD3	7[2]	I/O	**P0[25] —** General purpose digital input/output pin.
		I	**AD0[2] —** A/D converter 0, input 2.
		I/O	**I2SRX_SDA —** Receive data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I^2S-bus specification*. (LPC1759/58/56 only).
		O	**TXD3 —** Transmitter output for UART3.
P0[26]/AD0[3]/ AOUT/RXD3	6[3]	I/O	**P0[26] —** General purpose digital input/output pin.
		I	**AD0[3] —** A/D converter 0, input 3.
		O	**AOUT —** DAC output. (LPC1759/58/56/54 only).
		I	**RXD3 —** Receiver input for UART3.
P0[29]/USB_D+	22[4]	I/O	**P0[29] —** General purpose digital input/output pin.
		I/O	**USB_D+ —** USB bidirectional D+ line.
P0[30]/USB_D−	23[4]	I/O	**P0[30] —** General purpose digital input/output pin.
		I/O	**USB_D− —** USB bidirectional D− line.
P1[0] to P1[31]		I/O	**Port 1:** Port 1 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Some port pins are not available on the LQFP80 package.
P1[0]/ ENET_TXD0	76[1]	I/O	**P1[0] —** General purpose digital input/output pin.
		O	**ENET_TXD0 —** Ethernet transmit data 0. (LPC1758 only).
P1[1]/ ENET_TXD1	75[1]	I/O	**P1[1] —** General purpose digital input/output pin.
		O	**ENET_TXD1 —** Ethernet transmit data 1. (LPC1758 only).
P1[4]/ ENET_TX_EN	74[1]	I/O	**P1[4] —** General purpose digital input/output pin.
		O	**ENET_TX_EN —** Ethernet transmit data enable. (LPC1758 only).
P1[8]/ ENET_CRS	73[1]	I/O	**P1[8] —** General purpose digital input/output pin.
		I	**ENET_CRS —** Ethernet carrier sense. (LPC1758 only).
P1[9]/ ENET_RXD0	72[1]	I/O	**P1[9] —** General purpose digital input/output pin.
		I	**ENET_RXD0 —** Ethernet receive data. (LPC1758 only).
P1[10]/ ENET_RXD1	71[1]	I/O	**P1[10] —** General purpose digital input/output pin.
		I	**ENET_RXD1 —** Ethernet receive data. (LPC1758 only).

Table 73. Pin description (LPC175x) *…continued*

Symbol	Pin	Type	Description
P1[14]/ ENET_RX_ER	70[1]	I/O	**P1[14] —** General purpose digital input/output pin.
		I	**ENET_RX_ER —** Ethernet receive error. (LPC1758 only).
P1[15]/ ENET_REF_CLK	69[1]	I/O	**P1[15] —** General purpose digital input/output pin.
		I	**ENET_REF_CLK —** Ethernet reference clock. (LPC1758 only).
P1[18]/ USB_UP_LED/ PWM1[1]/ CAP1[0]	25[1]	I/O	**P1[18] —** General purpose digital input/output pin.
		O	**USB_UP_LED —** USB GoodLink LED indicator. It is LOW when the device is configured (non-control endpoints enabled), or when the host is enabled and has detected a device on the bus. It is HIGH when the device is not configured, or when host is enabled and has not detected a device on the bus, or during global suspend. It transitions between LOW and HIGH (flashes) when the host is enabled and detects activity on the bus.
		O	**PWM1[1] —** Pulse Width Modulator 1, channel 1 output.
		I	**CAP1[0] —** Capture input for Timer 1, channel 0.
P1[19]/MCOA0/ USB_PPWR CAP1[1]	26[1]	I/O	**P1[19] —** General purpose digital input/output pin.
		O	**MCOA0 —** Motor control PWM channel 0, output A.
		O	**USB_PPWR —** Port Power enable signal for USB port. (LPC1759/58/56/54 only).
		I	**CAP1[1] —** Capture input for Timer 1, channel 1.
P1[20]/MCI0/ PWM1[2]/SCK0	27[1]	I/O	**P1[20] —** General purpose digital input/output pin.
		I	**MCI0 —** Motor control PWM channel 0, input. Also Quadrature Encoder Interface PHA input.
		O	**PWM1[2] —** Pulse Width Modulator 1, channel 2 output.
		I/O	**SCK0 —** Serial clock for SSP0.
P1[22]/MCOB0/ USB_PWRD/ MAT1[0]	28[1]	I/O	**P1[22] —** General purpose digital input/output pin.
		O	**MCOB0 —** Motor control PWM channel 0, output B.
		I	**USB_PWRD —** Power Status for USB port (host power switch). (LPC1759/58/56/54 only).
		O	**MAT1[0] —** Match output for Timer 1, channel 0.
P1[23]/MCI1/ PWM1[4]/MISO0	29[1]	I/O	**P1[23] —** General purpose digital input/output pin.
		I	**MCI1 —** Motor control PWM channel 1, input. Also Quadrature Encoder Interface PHB input.
		O	**PWM1[4] —** Pulse Width Modulator 1, channel 4 output.
		I/O	**MISO0 —** Master In Slave Out for SSP0.
P1[24]/MCI2/ PWM1[5]/MOSI0	30[1]	I/O	**P1[24] —** General purpose digital input/output pin.
		I	**MCI2 —** Motor control PWM channel 2, input. Also Quadrature Encoder Interface INDEX input.
		O	**PWM1[5] —** Pulse Width Modulator 1, channel 5 output.
		I/O	**MOSI0 —** Master Out Slave in for SSP0.
P1[25]/MCOA1/ MAT1[1]	31[1]	I/O	**P1[25] —** General purpose digital input/output pin.
		O	**MCOA1 —** Motor control PWM channel 1, output A.
		O	**MAT1[1] —** Match output for Timer 1, channel 1.

Table 73. Pin description (LPC175x) *…continued*

Symbol	Pin	Type	Description
P1[26]/MCOB1/ PWM1[6]/CAP0[0]	32[1]	I/O	**P1[26] —** General purpose digital input/output pin.
		O	**MCOB1 —** Motor control PWM channel 1, output B.
		O	**PWM1[6] —** Pulse Width Modulator 1, channel 6 output.
		I	**CAP0[0] —** Capture input for Timer 0, channel 0.
P1[28]/MCOA2/ PCAP1[0]/ MAT0[0]	35[1]	I/O	**P1[28] —** General purpose digital input/output pin.
		O	**MCOA2 —** Motor control PWM channel 2, output A.
		I	**PCAP1[0] —** Capture input for PWM1, channel 0.
		O	**MAT0[0] —** Match output for Timer 0, channel 0.
P1[29]/MCOB2/ PCAP1[1]/ MAT0[1]	36[1]	I/O	**P1[29] —** General purpose digital input/output pin.
		O	**MCOB2 —** Motor control PWM channel 2, output B.
		I	**PCAP1[1] —** Capture input for PWM1, channel 1.
		O	**MAT0[1] —** Match output for Timer 0, channel 1.
P1[30]/V_{BUS}/ AD0[4]	18[2]	I/O	**P1[30] —** General purpose digital input/output pin.
		I	**V_{BUS} —** Monitors the presence of USB bus power. **Note:** This signal must be HIGH for USB reset to occur.
		I	**AD0[4] —** A/D converter 0, input 4.
P1[31]/SCK1/ AD0[5]	17[2]	I/O	**P1[31] —** General purpose digital input/output pin.
		I/O	**SCK1 —** Serial Clock for SSP1.
		I	**AD0[5] —** A/D converter 0, input 5.
P2[0] to P2[31]		I/O	**Port 2:** Port 2 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 2 pins depends upon the pin function selected via the pin connect block. Some port pins are not available on the LQFP80 package.
P2[0]/PWM1[1]/ TXD1	60[1]	I/O	**P2[0] —** General purpose digital input/output pin.
		O	**PWM1[1] —** Pulse Width Modulator 1, channel 1 output.
		O	**TXD1 —** Transmitter output for UART1.
P2[1]/PWM1[2]/ RXD1	59[1]	I/O	**P2[1] —** General purpose digital input/output pin.
		O	**PWM1[2] —** Pulse Width Modulator 1, channel 2 output.
		I	**RXD1 —** Receiver input for UART1.
P2[2]/PWM1[3]/ CTS1/ TRACEDATA[3]	58[1]	I/O	**P2[2] —** General purpose digital input/output pin.
		O	**PWM1[3] —** Pulse Width Modulator 1, channel 3 output.
		I	**CTS1 —** Clear to Send input for UART1.
		O	**TRACEDATA[3] —** Trace data, bit 3.
P2[3]/PWM1[4]/ DCD1/ TRACEDATA[2]	55[1]	I/O	**P2[3] —** General purpose digital input/output pin.
		O	**PWM1[4] —** Pulse Width Modulator 1, channel 4 output.
		I	**DCD1 —** Data Carrier Detect input for UART1.
		O	**TRACEDATA[2] —** Trace data, bit 2.
P2[4]/PWM1[5]/ DSR1/ TRACEDATA[1]	54[1]	I/O	**P2[4] —** General purpose digital input/output pin.
		O	**PWM1[5] —** Pulse Width Modulator 1, channel 5 output.
		I	**DSR1 —** Data Set Ready input for UART1.
		O	**TRACEDATA[1] —** Trace data, bit 1.

Table 73. Pin description (LPC175x) *…continued*

Symbol	Pin	Type	Description
P2[5]/PWM1[6]/ DTR1/ TRACEDATA[0]	53[1]	I/O	**P2[5] —** General purpose digital input/output pin.
		O	**PWM1[6] —** Pulse Width Modulator 1, channel 6 output.
		O	**DTR1 —** Data Terminal Ready output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
		O	**TRACEDATA[0] —** Trace data, bit 0.
P2[6]/PCAP1[0]/ RI1/TRACECLK	52[1]	I/O	**P2[6] —** General purpose digital input/output pin.
		I	**PCAP1[0] —** Capture input for PWM1, channel 0.
		I	**RI1 —** Ring Indicator input for UART1.
		O	**TRACECLK —** Trace Clock.
P2[7]/RD2/ RTS1	51[1]	I/O	**P2[7] —** General purpose digital input/output pin.
		I	**RD2 —** CAN2 receiver input. (LPC1759/58/56 only).
		O	**RTS1 —** Request to Send output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
P2[8]/TD2/ TXD2	50[1]	I/O	**P2[8] —** General purpose digital input/output pin.
		O	**TD2 —** CAN2 transmitter output. (LPC1759/58/56 only).
		O	**TXD2 —** Transmitter output for UART2.
P2[9]/ USB_CONNECT/ RXD2	49[1]	I/O	**P2[9] —** General purpose digital input/output pin.
		O	**USB_CONNECT —** Signal used to switch an external 1.5 kΩ resistor under software control. Used with the SoftConnect USB feature.
		I	**RXD2 —** Receiver input for UART2.
P2[10]/$\overline{\text{EINT0}}$/NMI	41[5]	I/O	**P2[10] —** General purpose digital input/output pin. A LOW level on this pin during reset starts the ISP command handler.
		I	**$\overline{\text{EINT0}}$ —** External interrupt 0 input.
		I	**NMI —** Non-maskable interrupt input.
P4[0] to P4[31]		I/O	**Port 4:** Port 4 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 4 pins depends upon the pin function selected via the pin connect block. Some port pins are not available on the LQFP80 package.
P4[28]/RX_MCLK/ MAT2[0]/TXD3	65[1]	I/O	**P4[28] —** General purpose digital input/output pin.
		O	**RX_MCLK —** I^2S receive master clock. (LPC1759/58/56 only).
		O	**MAT2[0] —** Match output for Timer 2, channel 0.
		O	**TXD3 —** Transmitter output for UART3.
P4[29]/TX_MCLK/ MAT2[1]/RXD3	68[1]	I/O	**P4[29] —** General purpose digital input/output pin.
		O	**TX_MCLK —** I^2S transmit master clock. (LPC1759/58/56 only).
		O	**MAT2[1] —** Match output for Timer 2, channel 1.
		I	**RXD3 —** Receiver input for UART3.
TDO/SWO	1[6]	O	**TDO —** Test Data out for JTAG interface.
		O	**SWO —** Serial wire trace output.
TDI	2[7]	I	**TDI —** Test Data in for JTAG interface.
TMS/SWDIO	3[7]	I	**TMS —** Test Mode Select for JTAG interface.
		I/O	**SWDIO —** Serial wire debug data input/output.
$\overline{\text{TRST}}$	4[7]	I	**$\overline{\text{TRST}}$ —** Test Reset for JTAG interface.

Table 73. Pin description (LPC175x) *…continued*

Symbol	Pin	Type	Description
TCK/SWDCLK	5[6]	I	**TCK —** Test Clock for JTAG interface.
		I	**SWDCLK —** Serial wire clock.
$\overline{\text{RSTOUT}}$	11	O	**$\overline{\text{RSTOUT}}$ —** This is a 3.3 V pin. LOW on this pin indicates UM10360 being in Reset state.
$\overline{\text{RESET}}$	14[8]	I	**External reset input:** A LOW-going pulse as short as 50 ns on this pin resets the device, causing I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. TTL with hysteresis, 5 V tolerant.
XTAL1	19[9][10]	I	Input to the oscillator circuit and internal clock generator circuits.
XTAL2	20[9][10]	O	Output from the oscillator amplifier.
RTCX1	13[9][11]	I	Input to the RTC oscillator circuit.
RTCX2	15[9]	O	Output from the RTC oscillator circuit.
V_{SS}	24, 33, 43, 57, 66, 78	I	**ground:** 0 V reference.
V_{SSA}	9	I	**analog ground:** 0 V reference. This should nominally be the same voltage as V_{SS}, but should be isolated to minimize noise and error.
$V_{DD(3V3)}$	21, 42, 56, 77	I	**3.3 V supply voltage:** This is the power supply voltage for the I/O ports.
$V_{DD(REG)(3V3)}$	34, 67	I	**3.3 V voltage regulator supply voltage:** This is the supply voltage for the on-chip voltage regulator only.
V_{DDA}	8	I	**analog 3.3 V pad supply voltage:** This should be nominally the same voltage as $V_{DD(3V3)}$ but should be isolated to minimize noise and error. This voltage is used to power the ADC and DAC. This pin should be tied to 3.3 V if the ADC and DAC are not used.
VREFP	10	I	**ADC positive reference voltage:** This should be nominally the same voltage as V_{DDA} but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC and DAC. This pin should be tied to 3.3 V if the ADC and DAC are not used.
VREFN	12	I	**ADC negative reference voltage:** This should be nominally the same voltage as V_{SS} but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC and DAC.
VBAT	16[11]	I	**RTC pin power supply:** 3.3 V on this pin supplies the power to the RTC peripheral.

[1] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[2] 5 V tolerant pad providing digital I/O functions (with TTL levels and hysteresis) and analog input. When configured as a ADC input, digital section of the pad is disabled and the pin is not 5 V tolerant. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[3] 5 V tolerant pad providing digital I/O with TTL levels and hysteresis and analog output function. When configured as the DAC output, digital section of the pad is disabled. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[4] Pad provides digital I/O and USB functions. It is designed in accordance with the *USB specification, revision 2.0* (Full-speed and Low-speed mode only). This pad is not 5 V tolerant.

[5] 5 V tolerant pad with 10 ns glitch filter providing digital I/O functions with TTL levels and hysteresis. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[6] 5 V tolerant pad with TTL levels and hysteresis. Internal pull-up and pull-down resistors disabled.

[7] 5 V tolerant pad with TTL levels and hysteresis and internal pull-up resistor.

[8] 5 V tolerant pad with 20 ns glitch filter providing digital I/O function with TTL levels and hysteresis.

[9] Pad provides special analog functionality. 32 kHz crystal oscillator must be used with the RTC.

[10] When the system oscillator is not used, connect XTAL1 and XTAL2 as follows: XTAL1 can be left floating or can be grounded (grounding is preferred to reduce susceptibility to noise). XTAL2 should be left floating.

[11] When the RTC is not used, connect VBAT to $V_{DD(REG)(3V3)}$ and leave RTCX1 floating.

Table 74. Pin description (LPC176x)

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P0[0] to P0[31]					I/O	**Port 0:** Port 0 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 0 pins depends upon the pin function selected via the pin connect block. Pins 12, 13, 14, and 31 of this port are not available.
P0[0]/RD1/TXD3/ SDA1	46	K8	H10	[1]	I/O	**P0[0] —** General purpose digital input/output pin.
					I	**RD1 —** CAN1 receiver input. (LPC1769/68/66/65/64 only).
					O	**TXD3 —** Transmitter output for UART3.
					I/O	**SDA1 —** I^2C1 data input/output. (This is not an I^2C-bus compliant open-drain pin).
P0[1]/TD1/RXD3/ SCL1	47	J8	H9	[1]	I/O	**P0[1] —** General purpose digital input/output pin.
					O	**TD1 —** CAN1 transmitter output. (LPC1769/68/66/65/64 only).
					I	**RXD3 —** Receiver input for UART3.
					I/O	**SCL1 —** I^2C1 clock input/output. (This is not an I^2C-bus compliant open-drain pin).
P0[2]/TXD0/AD0[7]	98	C4	B1	[2]	I/O	**P0[2] —** General purpose digital input/output pin.
					O	**TXD0 —** Transmitter output for UART0.
					I	**AD0[7] —** A/D converter 0, input 7.
P0[3]/RXD0/AD0[6]	99	A2	C3	[2]	I/O	**P0[3] —** General purpose digital input/output pin.
					I	**RXD0 —** Receiver input for UART0.
					I	**AD0[6] —** A/D converter 0, input 6.
P0[4]/ I2SRX_CLK/ RD2/CAP2[0]	81	A8	G2	[1]	I/O	**P0[4] —** General purpose digital input/output pin.
					I/O	**I2SRX_CLK —** Receive Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I	**RD2 —** CAN2 receiver input. (LPC1769/68/66/65/64 only).
					I	**CAP2[0] —** Capture input for Timer 2, channel 0.
P0[5]/ I2SRX_WS/ TD2/CAP2[1]	80	D7	H1	[1]	I/O	**P0[5] —** General purpose digital input/output pin.
					I/O	**I2SRX_WS —** Receive Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					O	**TD2 —** CAN2 transmitter output. (LPC1769/68/66/65/64 only).
					I	**CAP2[1] —** Capture input for Timer 2, channel 1.

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P0[6]/ I2SRX_SDA/ SSEL1/MAT2[0]	79	B8	G3	[1]	I/O	**P0[6] —** General purpose digital input/output pin.
					I/O	**I2SRX_SDA —** Receive data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I/O	**SSEL1 —** Slave Select for SSP1.
					O	**MAT2[0] —** Match output for Timer 2, channel 0.
P0[7]/ I2STX_CLK/ SCK1/MAT2[1]	78	A9	J1	[1]	I/O	**P0[7] —** General purpose digital input/output pin.
					I/O	**I2STX_CLK —** Transmit Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I/O	**SCK1 —** Serial Clock for SSP1.
					O	**MAT2[1] —** Match output for Timer 2, channel 1.
P0[8]/ I2STX_WS/ MISO1/MAT2[2]	77	C8	H2	[1]	I/O	**P0[8] —** General purpose digital input/output pin.
					I/O	**I2STX_WS —** Transmit Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I/O	**MISO1 —** Master In Slave Out for SSP1.
					O	**MAT2[2] —** Match output for Timer 2, channel 2.
P0[9]/ I2STX_SDA/ MOSI1/MAT2[3]	76	A10	H3	[1]	I/O	**P0[9] —** General purpose digital input/output pin.
					I/O	**I2STX_SDA —** Transmit data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I/O	**MOSI1 —** Master Out Slave In for SSP1.
					O	**MAT2[3] —** Match output for Timer 2, channel 3.
P0[10]/TXD2/ SDA2/MAT3[0]	48	H7	H8	[1]	I/O	**P0[10] —** General purpose digital input/output pin.
					O	**TXD2 —** Transmitter output for UART2.
					I/O	**SDA2 —** I^2C2 data input/output (this is not an open-drain pin).
					O	**MAT3[0] —** Match output for Timer 3, channel 0.
P0[11]/RXD2/ SCL2/MAT3[1]	49	K9	J10	[1]	I/O	**P0[11] —** General purpose digital input/output pin.
					I	**RXD2 —** Receiver input for UART2.
					I/O	**SCL2 —** I^2C2 clock input/output (this is not an open-drain pin).
					O	**MAT3[1] —** Match output for Timer 3, channel 1.
P0[15]/TXD1/ SCK0/SCK	62	F10	H6	[1]	I/O	**P0[15] —** General purpose digital input/output pin.
					O	**TXD1 —** Transmitter output for UART1.
					I/O	**SCK0 —** Serial clock for SSP0.
					I/O	**SCK —** Serial clock for SPI.

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P0[16]/RXD1/ SSEL0/SSEL	63	F8	J5	[1]	I/O	**P0[16] —** General purpose digital input/output pin.
					I	**RXD1 —** Receiver input for UART1.
					I/O	**SSEL0 —** Slave Select for SSP0.
					I/O	**SSEL —** Slave Select for SPI.
P0[17]/CTS1/ MISO0/MISO	61	F9	K6	[1]	I/O	**P0[17] —** General purpose digital input/output pin.
					I	**CTS1 —** Clear to Send input for UART1.
					I/O	**MISO0 —** Master In Slave Out for SSP0.
					I/O	**MISO —** Master In Slave Out for SPI.
P0[18]/DCD1/ MOSI0/MOSI	60	F6	J6	[1]	I/O	**P0[18] —** General purpose digital input/output pin.
					I	**DCD1 —** Data Carrier Detect input for UART1.
					I/O	**MOSI0 —** Master Out Slave In for SSP0.
					I/O	**MOSI —** Master Out Slave In for SPI.
P0[19]/DSR1/ SDA1	59	G10	K7	[1]	I/O	**P0[19] —** General purpose digital input/output pin.
					I	**DSR1 —** Data Set Ready input for UART1.
					I/O	**SDA1 —** I^2C1 data input/output (this is not an I^2C-bus compliant open-drain pin).
P0[20]/DTR1/SCL1	58	G9	J7	[1]	I/O	**P0[20] —** General purpose digital input/output pin.
					O	**DTR1 —** Data Terminal Ready output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
					I/O	**SCL1 —** I^2C1 clock input/output (this is not an I^2C-bus compliant open-drain pin).
P0[21]/RI1/RD1	57	G8	H7	[1]	I/O	**P0[21] —** General purpose digital input/output pin.
					I	**RI1 —** Ring Indicator input for UART1.
					I	**RD1 —** CAN1 receiver input. (LPC1769/68/66/65/64 only).
P0[22]/RTS1/TD1	56	H10	K8	[1]	I/O	**P0[22] —** General purpose digital input/output pin.
					O	**RTS1 —** Request to Send output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
					O	**TD1 —** CAN1 transmitter output. (LPC1769/68/66/65/64 only).
P0[23]/AD0[0]/ I2SRX_CLK/ CAP3[0]	9	E5	D5	[2]	I/O	**P0[23] —** General purpose digital input/output pin.
					I	**AD0[0] —** A/D converter 0, input 0.
					I/O	**I2SRX_CLK —** Receive Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I	**CAP3[0] —** Capture input for Timer 3, channel 0.

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P0[24]/AD0[1]/ I2SRX_WS/ CAP3[1]	8	D1	B4	[2]	I/O	**P0[24] —** General purpose digital input/output pin.
					I	**AD0[1] —** A/D converter 0, input 1.
					I/O	**I2SRX_WS —** Receive Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					I	**CAP3[1] —** Capture input for Timer 3, channel 1.
P0[25]/AD0[2]/ I2SRX_SDA/ TXD3	7	D2	A3	[2]	I/O	**P0[25] —** General purpose digital input/output pin.
					I	**AD0[2] —** A/D converter 0, input 2.
					I/O	**I2SRX_SDA —** Receive data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
					O	**TXD3 —** Transmitter output for UART3.
P0[26]/AD0[3]/ AOUT/RXD3	6	D3	C5	[3]	I/O	**P0[26] —** General purpose digital input/output pin.
					I	**AD0[3] —** A/D converter 0, input 3.
					O	**AOUT —** DAC output (LPC1769/68/67/66/65/63 only).
					I	**RXD3 —** Receiver input for UART3.
P0[27]/SDA0/ USB_SDA	25	J2	C8	[4]	I/O	**P0[27] —** General purpose digital input/output pin. Output is open-drain.
					I/O	**SDA0 —** I^2C0 data input/output. Open-drain output (for I^2C-bus compliance).
					I/O	**USB_SDA —** USB port I^2C serial data (OTG transceiver, LPC1769/68/66/65 only).
P0[28]/SCL0/ USB_SCL	24	J1	B9	[4]	I/O	**P0[28] —** General purpose digital input/output pin. Output is open-drain.
					I/O	**SCL0 —** I^2C0 clock input/output. Open-drain output (for I^2C-bus compliance).
					I/O	**USB_SCL —** USB port I^2C serial clock (OTG transceiver, LPC1769/68/66/65 only).
P0[29]/USB_D+	29	J3	B10	[5]	I/O	**P0[29] —** General purpose digital input/output pin.
					I/O	**USB_D+ —** USB bidirectional D+ line. (LPC1769/68/66/65/64 only).
P0[30]/USB_D−	30	G4	C9	[5]	I/O	**P0[30] —** General purpose digital input/output pin.
					I/O	**USB_D− —** USB bidirectional D− line. (LPC1769/68/66/65/64 only).
P1[0] to P1[31]					I/O	**Port 1:** Port 1 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 1 pins depends upon the pin function selected via the pin connect block. Pins 2, 3, 5, 6, 7, 11, 12, and 13 of this port are not available.
P1[0]/ ENET_TXD0	95	D5	C1	[1]	I/O	**P1[0] —** General purpose digital input/output pin.
					O	**ENET_TXD0 —** Ethernet transmit data 0. (LPC1769/68/67/66/64 only).

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P1[1]/ ENET_TXD1	94	B4	C2	[1]	I/O	**P1[1] —** General purpose digital input/output pin.
					O	**ENET_TXD1 —** Ethernet transmit data 1. (LPC1769/68/67/66/64 only).
P1[4]/ ENET_TX_EN	93	A4	D2	[1]	I/O	**P1[4] —** General purpose digital input/output pin.
					O	**ENET_TX_EN —** Ethernet transmit data enable. (LPC1769/68/67/66/64 only).
P1[8]/ ENET_CRS	92	C5	D1	[1]	I/O	**P1[8] —** General purpose digital input/output pin.
					I	**ENET_CRS —** Ethernet carrier sense. (LPC1769/68/67/66/64 only).
P1[9]/ ENET_RXD0	91	B5	D3	[1]	I/O	**P1[9] —** General purpose digital input/output pin.
					I	**ENET_RXD0 —** Ethernet receive data. (LPC1769/68/67/66/64 only).
P1[10]/ ENET_RXD1	90	A5	E3	[1]	I/O	**P1[10] —** General purpose digital input/output pin.
					I	**ENET_RXD1 —** Ethernet receive data. (LPC1769/68/67/66/64 only).
P1[14]/ ENET_RX_ER	89	D6	E2	[1]	I/O	**P1[14] —** General purpose digital input/output pin.
					I	**ENET_RX_ER —** Ethernet receive error. (LPC1769/68/67/66/64 only).
P1[15]/ ENET_REF_CLK	88	C6	E1	[1]	I/O	**P1[15] —** General purpose digital input/output pin.
					I	**ENET_REF_CLK —** Ethernet reference clock. (LPC1769/68/67/66/64 only).
P1[16]/ ENET_MDC	87	A6	F3	[1]	I/O	**P1[16] —** General purpose digital input/output pin.
					O	**ENET_MDC —** Ethernet MIIM clock (LPC1769/68/67/66/64 only).
P1[17]/ ENET_MDIO	86	B6	F2	[1]	I/O	**P1[17] —** General purpose digital input/output pin.
					I/O	**ENET_MDIO —** Ethernet MIIM data input and output. (LPC1769/68/67/66/64 only).
P1[18]/ USB_UP_LED/ PWM1[1]/ CAP1[0]	32	H4	D9	[1]	I/O	**P1[18] —** General purpose digital input/output pin.
					O	**USB_UP_LED —** USB GoodLink LED indicator. It is LOW when the device is configured (non-control endpoints enabled), or when the host is enabled and has detected a device on the bus. It is HIGH when the device is not configured, or when host is enabled and has not detected a device on the bus, or during global suspend. It transitions between LOW and HIGH (flashes) when the host is enabled and detects activity on the bus. (LPC1769/68/66/65/64 only).
					O	**PWM1[1] —** Pulse Width Modulator 1, channel 1 output.
					I	**CAP1[0] —** Capture input for Timer 1, channel 0.
P1[19]/MCOA0/ USB_PPWR/ CAP1[1]	33	J4	C10	[1]	I/O	**P1[19] —** General purpose digital input/output pin.
					O	**MCOA0 —** Motor control PWM channel 0, output A.
					O	**USB_PPWR —** Port Power enable signal for USB port. (LPC1769/68/66/65 only).
					I	**CAP1[1] —** Capture input for Timer 1, channel 1.

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P1[20]/MCI0/ PWM1[2]/SCK0	34	K4	E8	[1]	I/O	**P1[20] —** General purpose digital input/output pin.
					I	**MCI0 —** Motor control PWM channel 0, input. Also Quadrature Encoder Interface PHA input.
					O	**PWM1[2] —** Pulse Width Modulator 1, channel 2 output.
					I/O	**SCK0 —** Serial clock for SSP0.
P1[21]/$\overline{\text{MCABORT}}$/ PWM1[3]/ SSEL0	35	F5	E9	[1]	I/O	**P1[21] —** General purpose digital input/output pin.
					O	**$\overline{\text{MCABORT}}$ —** Motor control PWM, LOW-active fast abort.
					O	**PWM1[3] —** Pulse Width Modulator 1, channel 3 output.
					I/O	**SSEL0 —** Slave Select for SSP0.
P1[22]/MCOB0/ USB_PWRD/ MAT1[0]	36	J5	D10	[1]	I/O	**P1[22] —** General purpose digital input/output pin.
					O	**MCOB0 —** Motor control PWM channel 0, output B.
					I	**USB_PWRD —** Power Status for USB port (host power switch, LPC1769/68/66/65 only).
					O	**MAT1[0] —** Match output for Timer 1, channel 0.
P1[23]/MCI1/ PWM1[4]/MISO0	37	K5	E7	[1]	I/O	**P1[23] —** General purpose digital input/output pin.
					I	**MCI1 —** Motor control PWM channel 1, input. Also Quadrature Encoder Interface PHB input.
					O	**PWM1[4] —** Pulse Width Modulator 1, channel 4 output.
					I/O	**MISO0 —** Master In Slave Out for SSP0.
P1[24]/MCI2/ PWM1[5]/MOSI0	38	H5	F8	[1]	I/O	**P1[24] —** General purpose digital input/output pin.
					I	**MCI2 —** Motor control PWM channel 2, input. Also Quadrature Encoder Interface INDEX input.
					O	**PWM1[5] —** Pulse Width Modulator 1, channel 5 output.
					I/O	**MOSI0 —** Master Out Slave in for SSP0.
P1[25]/MCOA1/ MAT1[1]	39	G5	F9	[1]	I/O	**P1[25] —** General purpose digital input/output pin.
					O	**MCOA1 —** Motor control PWM channel 1, output A.
					O	**MAT1[1] —** Match output for Timer 1, channel 1.
P1[26]/MCOB1/ PWM1[6]/CAP0[0]	40	K6	E10	[1]	I/O	**P1[26] —** General purpose digital input/output pin.
					O	**MCOB1 —** Motor control PWM channel 1, output B.
					O	**PWM1[6] —** Pulse Width Modulator 1, channel 6 output.
					I	**CAP0[0] —** Capture input for Timer 0, channel 0.
P1[27]/CLKOUT /USB_OVRCR/ CAP0[1]	43	K7	G9	[1]	I/O	**P1[27] —** General purpose digital input/output pin.
					O	**CLKOUT —** Clock output pin.
					I	**$\overline{\text{USB\_OVRCR}}$ —** USB port Over-Current status. (LPC1769/68/66/65 only).
					I	**CAP0[1] —** Capture input for Timer 0, channel 1.

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P1[28]/MCOA2/ PCAP1[0]/ MAT0[0]	44	J7	G10	[1]	I/O	**P1[28] —** General purpose digital input/output pin.
					O	**MCOA2 —** Motor control PWM channel 2, output A.
					I	**PCAP1[0] —** Capture input for PWM1, channel 0.
					O	**MAT0[0] —** Match output for Timer 0, channel 0.
P1[29]/MCOB2/ PCAP1[1]/ MAT0[1]	45	G6	G8	[1]	I/O	**P1[29] —** General purpose digital input/output pin.
					O	**MCOB2 —** Motor control PWM channel 2, output B.
					I	**PCAP1[1] —** Capture input for PWM1, channel 1.
					O	**MAT0[1] —** Match output for Timer 0, channel 1.
P1[30]/V_{BUS}/ AD0[4]	21	H1	B8	[2]	I/O	**P1[30] —** General purpose digital input/output pin.
					I	**V_{BUS} —** Monitors the presence of USB bus power. (LPC1769/68/66/65/64 only). **Note:** This signal must be HIGH for USB reset to occur.
					I	**AD0[4] —** A/D converter 0, input 4.
P1[31]/SCK1/ AD0[5]	20	F4	C7	[2]	I/O	**P1[31] —** General purpose digital input/output pin.
					I/O	**SCK1 —** Serial Clock for SSP1.
					I	**AD0[5] —** A/D converter 0, input 5.
P2[0] to P2[31]					I/O	**Port 2:** Port 2 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 2 pins depends upon the pin function selected via the pin connect block. Pins 14 through 31 of this port are not available.
P2[0]/PWM1[1]/ TXD1	75	B9	K1	[1]	I/O	**P2[0] —** General purpose digital input/output pin.
					O	**PWM1[1] —** Pulse Width Modulator 1, channel 1 output.
					O	**TXD1 —** Transmitter output for UART1.
P2[1]/PWM1[2]/ RXD1	74	B10	J2	[1]	I/O	**P2[1] —** General purpose digital input/output pin.
					O	**PWM1[2] —** Pulse Width Modulator 1, channel 2 output.
					I	**RXD1 —** Receiver input for UART1.
P2[2]/PWM1[3]/ CTS1/ TRACEDATA[3]	73	D8	K2	[1]	I/O	**P2[2] —** General purpose digital input/output pin.
					O	**PWM1[3] —** Pulse Width Modulator 1, channel 3 output.
					I	**CTS1 —** Clear to Send input for UART1.
					O	**TRACEDATA[3] —** Trace data, bit 3.
P2[3]/PWM1[4]/ DCD1/ TRACEDATA[2]	70	E7	K3	[1]	I/O	**P2[3] —** General purpose digital input/output pin.
					O	**PWM1[4] —** Pulse Width Modulator 1, channel 4 output.
					I	**DCD1 —** Data Carrier Detect input for UART1.
					O	**TRACEDATA[2] —** Trace data, bit 2.
P2[4]/PWM1[5]/ DSR1/ TRACEDATA[1]	69	D9	J3	[1]	I/O	**P2[4] —** General purpose digital input/output pin.
					O	**PWM1[5] —** Pulse Width Modulator 1, channel 5 output.
					I	**DSR1 —** Data Set Ready input for UART1.
					O	**TRACEDATA[1] —** Trace data, bit 1.

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P2[5]/PWM1[6]/ DTR1/ TRACEDATA[0]	68	D10	H4	[1]	I/O	**P2[5] —** General purpose digital input/output pin.
					O	**PWM1[6] —** Pulse Width Modulator 1, channel 6 output.
					O	**DTR1 —** Data Terminal Ready output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
					O	**TRACEDATA[0] —** Trace data, bit 0.
P2[6]/PCAP1[0]/ RI1/TRACECLK	67	E8	K4	[1]	I/O	**P2[6] —** General purpose digital input/output pin.
					I	**PCAP1[0] —** Capture input for PWM1, channel 0.
					I	**RI1 —** Ring Indicator input for UART1.
					O	**TRACECLK —** Trace Clock.
P2[7]/RD2/ RTS1	66	E9	J4	[1]	I/O	**P2[7] —** General purpose digital input/output pin.
					I	**RD2 —** CAN2 receiver input. (LPC1769/68/66/65/64 only).
					O	**RTS1 —** Request to Send output for UART1. Can also be configured to be an RS-485/EIA-485 output enable signal.
P2[8]/TD2/ TXD2	65	E10	H5	[1]	I/O	**P2[8] —** General purpose digital input/output pin.
					O	**TD2 —** CAN2 transmitter output. (LPC1769/68/66/65/64 only).
					O	**TXD2 —** Transmitter output for UART2.
P2[9]/ USB_CONNECT/ RXD2	64	F7	K5	[1]	I/O	**P2[9] —** General purpose digital input/output pin.
					O	**USB_CONNECT —** Signal used to switch an external 1.5 kΩ resistor under software control. Used with the SoftConnect USB feature. (LPC1769/68/66/65/64 only).
					I	**RXD2 —** Receiver input for UART2.
P2[10]/$\overline{\text{EINT0}}$/NMI	53	J10	K9	[6]	I/O	**P2[10] —** General purpose digital input/output pin. A LOW level on this pin during reset starts the ISP command handler.
					I	**$\overline{\text{EINT0}}$ —** External interrupt 0 input.
					I	**NMI —** Non-maskable interrupt input.
P2[11]/$\overline{\text{EINT1}}$/ I2STX_CLK	52	H8	J8	[6]	I/O	**P2[11] —** General purpose digital input/output pin.
					I	**$\overline{\text{EINT1}}$ —** External interrupt 1 input.
					I/O	**I2STX_CLK —** Transmit Clock. It is driven by the master and received by the slave. Corresponds to the signal SCK in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
P2[12]/$\overline{\text{EINT2}}$/ I2STX_WS	51	K10	K10	[6]	I/O	**P2[12] —** General purpose digital input/output pin.
					I	**$\overline{\text{EINT2}}$ —** External interrupt 2 input.
					I/O	**I2STX_WS —** Transmit Word Select. It is driven by the master and received by the slave. Corresponds to the signal WS in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).
P2[13]/$\overline{\text{EINT3}}$/ I2STX_SDA	50	J9	J9	[6]	I/O	**P2[13] —** General purpose digital input/output pin.
					I	**$\overline{\text{EINT3}}$ —** External interrupt 3 input.
					I/O	**I2STX_SDA —** Transmit data. It is driven by the transmitter and read by the receiver. Corresponds to the signal SD in the *I^2S-bus specification*. (LPC1769/68/67/66/65/63 only).

Table 74. Pin description (LPC176x) *…continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
P3[0] to P3[31]					I/O	**Port 3:** Port 3 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 3 pins depends upon the pin function selected via the pin connect block. Pins 0 through 24, and 27 through 31 of this port are not available.
P3[25]/MAT0[0]/ PWM1[2]	27	H3	D8	[1]	I/O	**P3[25] —** General purpose digital input/output pin.
					O	**MAT0[0] —** Match output for Timer 0, channel 0.
					O	**PWM1[2] —** Pulse Width Modulator 1, output 2.
P3[26]/STCLK/ MAT0[1]/PWM1[3]	26	K1	A10	[1]	I/O	**P3[26] —** General purpose digital input/output pin.
					I	**STCLK —** System tick timer clock input. The maximum STCLK frequency is 1/4 of the ARM processor clock frequency CCLK.
					O	**MAT0[1] —** Match output for Timer 0, channel 1.
					O	**PWM1[3] —** Pulse Width Modulator 1, output 3.
P4[0] to P4[31]					I/O	**Port 4:** Port 4 is a 32-bit I/O port with individual direction controls for each bit. The operation of port 4 pins depends upon the pin function selected via the pin connect block. Pins 0 through 27, 30, and 31 of this port are not available.
P4[28]/RX_MCLK/ MAT2[0]/TXD3	82	C7	G1	[1]	I/O	**P4[28] —** General purpose digital input/output pin.
					O	**RX_MCLK —** I^2S receive master clock. (LPC1769/68/67/66/65 only).
					O	**MAT2[0] —** Match output for Timer 2, channel 0.
					O	**TXD3 —** Transmitter output for UART3.
P4[29]/TX_MCLK/ MAT2[1]/RXD3	85	E6	F1	[1]	I/O	**P4[29] —** General purpose digital input/output pin.
					O	**TX_MCLK —** I^2S transmit master clock. (LPC1769/68/67/66/65 only).
					O	**MAT2[1] —** Match output for Timer 2, channel 1.
					I	**RXD3 —** Receiver input for UART3.
TDO/SWO	1	A1	A1	[1][7]	O	**TDO —** Test Data out for JTAG interface.
					O	**SWO —** Serial wire trace output.
TDI	2	C3	C4	[1][8]	I	**TDI —** Test Data in for JTAG interface.
TMS/SWDIO	3	B1	B3	[1][8]	I	**TMS —** Test Mode Select for JTAG interface.
					I/O	**SWDIO —** Serial wire debug data input/output.
$\overline{\text{TRST}}$	4	C2	A2	[1][8]	I	**$\overline{\text{TRST}}$ —** Test Reset for JTAG interface.
TCK/SWDCLK	5	C1	D4	[1][7]	I	**TCK —** Test Clock for JTAG interface.
					I	**SWDCLK —** Serial wire clock.
RTCK	100	B2	B2	[1][7]	O	**RTCK —** JTAG interface control signal.
$\overline{\text{RSTOUT}}$	14	-	-	-	O	**$\overline{\text{RSTOUT}}$ —** This is a 3.3 V pin. LOW on this pin indicates the microcontroller being in Reset state.
$\overline{\text{RESET}}$	17	F3	C6	[9]	I	**External reset input:** A LOW-going pulse as short as 50 ns on this pin resets the device, causing I/O ports and peripherals to take on their default states, and processor execution to begin at address 0. TTL with hysteresis, 5 V tolerant.

Table 74. Pin description (LPC176x) *...continued*

Symbol	Pin/ball				Type	Description
	LQFP100	TFBGA100	WLCSP100			
XTAL1	22	H2	D7	[10][11]	I	Input to the oscillator circuit and internal clock generator circuits.
XTAL2	23	G3	A9	[10][11]	O	Output from the oscillator amplifier.
RTCX1	16	F2	A7	[10][11]	I	Input to the RTC oscillator circuit.
RTCX2	18	G1	B7	[10]	O	Output from the RTC oscillator circuit.
V_{SS}	31, 41, 55, 72, 83, 97	B3, B7, C9, G7, J6, K3	E5, F5, F6, G5, G6, G7	[10]	I	**ground:** 0 V reference.
V_{SSA}	11	E1	B5	[10]	I	**analog ground:** 0 V reference. This should nominally be the same voltage as V_{SS}, but should be isolated to minimize noise and error.
$V_{DD(3V3)}$	28, 54, 71, 96	K2, H9, C10, A3	E4, E6, F7, G4	[10]	I	**3.3 V supply voltage:** This is the power supply voltage for the I/O ports.
$V_{DD(REG)(3V3)}$	42, 84	H6, A7	F4, F10	[10]	I	**3.3 V voltage regulator supply voltage:** This is the supply voltage for the on-chip voltage regulator only.
V_{DDA}	10	E2	A4	[10]	I	**analog 3.3 V pad supply voltage:** This should be nominally the same voltage as $V_{DD(3V3)}$ but should be isolated to minimize noise and error. This voltage is used to power the ADC and DAC. This pin should be tied to 3.3 V if the ADC and DAC are not used.
VREFP	12	E3	A5	[10]	I	**ADC positive reference voltage:** This should be nominally the same voltage as V_{DDA} but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC and DAC. This pin should be tied to 3.3 V if the ADC and DAC are not used.
VREFN	15	F1	A6		I	**ADC negative reference voltage:** This should be nominally the same voltage as V_{SS} but should be isolated to minimize noise and error. Level on this pin is used as a reference for ADC and DAC.
VBAT	19	G2	A8	[10]	I	**RTC pin power supply:** 3.3 V on this pin supplies the power to the RTC peripheral.
n.c.	13	D4, E4	B6, D6		-	not connected.

[1] 5 V tolerant pad providing digital I/O functions with TTL levels and hysteresis. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[2] 5 V tolerant pad providing digital I/O functions (with TTL levels and hysteresis) and analog input. When configured as a ADC input, digital section of the pad is disabled and the pin is not 5 V tolerant. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[3] 5 V tolerant pad providing digital I/O with TTL levels and hysteresis and analog output function. When configured as the DAC output, digital section of the pad is disabled. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[4] Open-drain 5 V tolerant digital I/O pad, compatible with I^2C-bus 400 kHz specification. This pad requires an external pull-up to provide output functionality. When power is switched off, this pin connected to the I^2C-bus is floating and does not disturb the I^2C lines. Open-drain configuration applies to all functions on this pin.

[5] Pad provides digital I/O and USB functions. It is designed in accordance with the *USB specification, revision 2.0* (Full-speed and Low-speed mode only). This pad is not 5 V tolerant.

[6] 5 V tolerant pad with 10 ns glitch filter providing digital I/O functions with TTL levels and hysteresis. This pin is pulled up to a voltage level of 2.3 V to 2.6 V.

[7] 5 V tolerant pad with TTL levels and hysteresis. Internal pull-up and pull-down resistors disabled.

[8] 5 V tolerant pad with TTL levels and hysteresis and internal pull-up resistor.

[9] 5 V tolerant pad with 20 ns glitch filter providing digital I/O function with TTL levels and hysteresis.

[10] Pad provides special analog functionality. A 32 kHz crystal oscillator must be used with the RTC.

[11] When the system oscillator is not used, connect XTAL1 and XTAL2 as follows: XTAL1 can be left floating or can be grounded (grounding is preferred to reduce susceptibility to noise). XTAL2 should be left floating.

[12] When the RTC is not used, connect VBAT to $V_{DD(REG)(3V3)}$ and leave RTCX1 floating.

8.1 How to read this chapter

Table 75 shows the functions of the PINSEL registers in the LPC176x/5x.

Table 75. Summary of PINSEL registers

Register	Controls	Table
PINSEL0	P0[15:0]	Table 80
PINSEL1	P0 [31:16]	Table 81
PINSEL2	P1 [15:0] (Ethernet)	Table 82
PINSEL3	P1 [31:16]	Table 83
PINSEL4	P2 [15:0]	Table 84
PINSEL5	P2 [31:16]	not used
PINSEL6	P3 [15:0]	not used
PINSEL7	P3 [31:16]	Table 85
PINSEL8	P4 [15:0]	not used
PINSEL9	P4 [31:16]	Table 86
PINSEL10	Trace port enable	Table 87

8.2 Description

The pin connect block allows most pins of the microcontroller to have more than one potential function. Configuration registers control the multiplexers to allow connection between the pin and the on chip peripherals.

Peripherals should be connected to the appropriate pins prior to being activated and prior to any related interrupt(s) being enabled. Activity of any enabled peripheral function that is not mapped to a related pin should be considered undefined.

Selection of a single function on a port pin excludes other peripheral functions available on the same pin. However, the GPIO input stays connected and may be read by software or used to contribute to the GPIO interrupt feature.

8.3 Pin function select register values

The PINSEL registers control the functions of device pins as shown below. Pairs of bits in these registers correspond to specific device pins.

Table 76. Pin function select register bits

PINSEL0 to PINSEL9 Values	Function	Value after Reset
00	Primary (default) function, typically GPIO port	00
01	First alternate function	
10	Second alternate function	
11	Third alternate function	

The direction control bit in the GPIO registers is effective only when the GPIO function is selected for a pin. For other functions, direction is controlled automatically. Each derivative typically has a different pinout and therefore a different set of functions possible for each pin. Details for a specific derivative may be found in the appropriate data sheet.

Multiple connections

Since a particular peripheral function may be allowed on more than one pin, it is in principle possible to configure more than one pin to perform the same function. If a peripheral output function is configured to appear on more than one pin, it will in fact be routed to those pins. If a peripheral input function is configured to appear on more than one pin for some reason, the peripheral will receive its input from the lowest port number. For instance, any pin of port 0 will take precedence over any pin of a higher numbered port, and pin 0 of any port will take precedence over a higher numbered pin of the same port.

8.4 Pin mode select register values

The PINMODE registers control the input mode of all ports. This includes the use of the on-chip pull-up/pull-down resistor feature and a special open drain operating mode. The on-chip pull-up/pull-down resistor can be selected for every port pin regardless of the function on this pin with the exception of the I^2C pins for the I2C0 interface and the USB pins (see Section 8.5.10). Three bits are used to control the mode of a port pin, two in a PINMODE register, and an additional one in a PINMODE_OD register. Bits are reserved for unused pins as in the PINSEL registers.

Table 77. Pin Mode Select register Bits

PINMODE0 to PINMODE9 Values	Function	Value after Reset
00	Pin has an on-chip pull-up resistor enabled.	00
01	Repeater mode (see text below).	
10	Pin has neither pull-up nor pull-down resistor enabled.	
11	Pin has an on-chip pull-down resistor enabled.	

Repeater mode enables the pull-up resistor if the pin is at a logic high and enables the pull-down resistor if the pin is at a logic low. This causes the pin to retain its last known state if it is configured as an input and is not driven externally. The state retention is not applicable to the Deep Power-down mode. 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.

The PINMODE_OD registers control the open drain mode for ports. The open drain mode causes the pin to be pulled low normally if it is configured as an output and the data value is 0. If the data value is 1, the output drive of the pin is turned off, equivalent to changing the pin direction. This combination simulates an open drain output.

Table 78. Open Drain Pin Mode Select register Bits

PINMODE_OD0 to PINMODE_OD4 Values	Function	Value after Reset
0	Pin is in the normal (not open drain) mode.	00
1	Pin is in the open drain mode.	

Function of PINMODE in open drain mode

Normally the value of PINMODE applies to a pin only when it is in the input mode. When a pin is in the open drain mode, caused by a 1 in the corresponding bit of one of the PINMODE_OD registers, the input mode still does not apply when the pin is outputting a 0. However, when the pin value is 1, PINMODE applies since this state turns off the pin's output driver. For example, this allows for the possibility of configuring a pin to be open drain with an on-chip pullup. A pullup in this case which is only on when the pin is not being pulled low by the pin's own output.

8.5 Register description

The Pin Control Module contains 11 registers as shown in Table 79 below.

Table 79. Pin Connect Block Register Map

Name	Description	Access	Reset Value[1]	Address
PINSEL0	Pin function select register 0.	R/W	0	0x4002 C000
PINSEL1	Pin function select register 1.	R/W	0	0x4002 C004
PINSEL2	Pin function select register 2.	R/W	0	0x4002 C008
PINSEL3	Pin function select register 3.	R/W	0	0x4002 C00C
PINSEL4	Pin function select register 4	R/W	0	0x4002 C010
PINSEL7	Pin function select register 7	R/W	0	0x4002 C01C
PINSEL8	Pin function select register 8	R/W	0	0x4002 C020
PINSEL9	Pin function select register 9	R/W	0	0x4002 C024
PINSEL10	Pin function select register 10	R/W	0	0x4002 C028
PINMODE0	Pin mode select register 0	R/W	0	0x4002 C040
PINMODE1	Pin mode select register 1	R/W	0	0x4002 C044
PINMODE2	Pin mode select register 2	R/W	0	0x4002 C048
PINMODE3	Pin mode select register 3.	R/W	0	0x4002 C04C
PINMODE4	Pin mode select register 4	R/W	0	0x4002 C050
PINMODE5	Pin mode select register 5	R/W	0	0x4002 C054
PINMODE6	Pin mode select register 6	R/W	0	0x4002 C058
PINMODE7	Pin mode select register 7	R/W	0	0x4002 C05C
PINMODE9	Pin mode select register 9	R/W	0	0x4002 C064
PINMODE_OD0	Open drain mode control register 0	R/W	0	0x4002 C068
PINMODE_OD1	Open drain mode control register 1	R/W	0	0x4002 C06C
PINMODE_OD2	Open drain mode control register 2	R/W	0	0x4002 C070
PINMODE_OD3	Open drain mode control register 3	R/W	0	0x4002 C074
PINMODE_OD4	Open drain mode control register 4	R/W	0	0x4002 C078
I2CPADCFG	I^2C Pin Configuration register	R/W	0	0x4002 C07C

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

Pin control module register reset values

On external reset, watchdog reset, power-on-reset (POR), and BOD reset, all registers in this module are reset to '0'.

8.5.1 Pin Function Select register 0 (PINSEL0 - 0x4002 C000)

The PINSEL0 register controls the functions of the lower half of Port 0. The direction control bit in FIO0DIR register is effective only when the GPIO function is selected for a pin. For other functions, the direction is controlled automatically.

Table 80. Pin function select register 0 (PINSEL0 - address 0x4002 C000) bit description

PINSEL0	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.0	GPIO Port 0.0	RD1	TXD3	SDA1	00
3:2	P0.1	GPIO Port 0.1	TD1	RXD3	SCL1	00
5:4	P0.2	GPIO Port 0.2	TXD0	AD0.7	Reserved	00
7:6	P0.3	GPIO Port 0.3	RXD0	AD0.6	Reserved	00
9:8	P0.4[1]	GPIO Port 0.4	I2SRX_CLK	RD2	CAP2.0	00
11:10	P0.5[1]	GPIO Port 0.5	I2SRX_WS	TD2	CAP2.1	00
13:12	P0.6	GPIO Port 0.6	I2SRX_SDA	SSEL1	MAT2.0	00
15:14	P0.7	GPIO Port 0.7	I2STX_CLK	SCK1	MAT2.1	00
17:16	P0.8	GPIO Port 0.8	I2STX_WS	MISO1	MAT2.2	00
19:18	P0.9	GPIO Port 0.9	I2STX_SDA	MOSI1	MAT2.3	00
21:20	P0.10	GPIO Port 0.10	TXD2	SDA2	MAT3.0	00
23:22	P0.11	GPIO Port 0.11	RXD2	SCL2	MAT3.1	00
29:24	-	Reserved	Reserved	Reserved	Reserved	0
31:30	P0.15	GPIO Port 0.15	TXD1	SCK0	SCK	00

[1] Not available on 80-pin package.

8.5.2 Pin Function Select Register 1 (PINSEL1 - 0x4002 C004)

The PINSEL1 register controls the functions of the upper half of Port 0. The direction control bit in the FIO0DIR register is effective only when the GPIO function is selected for a pin. For other functions the direction is controlled automatically.

Table 81. Pin function select register 1 (PINSEL1 - address 0x4002 C004) bit description

PINSEL1	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P0.16	GPIO Port 0.16	RXD1	SSEL0	SSEL	00
3:2	P0.17	GPIO Port 0.17	CTS1	MISO0	MISO	00
5:4	P0.18	GPIO Port 0.18	DCD1	MOSI0	MOSI	00
7:6	P0.19[1]	GPIO Port 0.19	DSR1	Reserved	SDA1	00
9:8	P0.20[1]	GPIO Port 0.20	DTR1	Reserved	SCL1	00
11:10	P0.21[1]	GPIO Port 0.21	RI1	Reserved	RD1	00
13:12	P0.22	GPIO Port 0.22	RTS1	Reserved	TD1	00
15:14	P0.23[1]	GPIO Port 0.23	AD0.0	I2SRX_CLK	CAP3.0	00
17:16	P0.24[1]	GPIO Port 0.24	AD0.1	I2SRX_WS	CAP3.1	00
19:18	P0.25	GPIO Port 0.25	AD0.2	I2SRX_SDA	TXD3	00
21:20	P0.26	GPIO Port 0.26	AD0.3	AOUT	RXD3	00
23:22	P0.27[1][2]	GPIO Port 0.27	SDA0	USB_SDA	Reserved	00

Table 81. Pin function select register 1 (PINSEL1 - address 0x4002 C004) bit description *...continued*

PINSEL1	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
25:24	P0.28[1][2]	GPIO Port 0.28	SCL0	USB_SCL	Reserved	00
27:26	P0.29	GPIO Port 0.29	USB_D+	Reserved	Reserved	00
29:28	P0.30	GPIO Port 0.30	USB_D–	Reserved	Reserved	00
31:30	-	Reserved	Reserved	Reserved	Reserved	00

[1] Not available on 80-pin package.

[2] Pins P0[27] and P0[28] are open-drain for I^2C-bus compliance.

8.5.3 Pin Function Select register 2 (PINSEL2 - 0x4002 C008)

The PINSEL2 register controls the functions of the lower half of Port 1, which contains the Ethernet related pins. The direction control bit in the FIO1DIR register is effective only when the GPIO function is selected for a pin. For other functions, the direction is controlled automatically.

Table 82. Pin function select register 2 (PINSEL2 - address 0x4002 C008) bit description

PINSEL2	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P1.0	GPIO Port 1.0	ENET_TXD0	Reserved	Reserved	00
3:2	P1.1	GPIO Port 1.1	ENET_TXD1	Reserved	Reserved	00
7:4	-	Reserved	Reserved	Reserved	Reserved	0
9:8	P1.4	GPIO Port 1.4	ENET_TX_EN	Reserved	Reserved	00
15:10	-	Reserved	Reserved	Reserved	Reserved	0
17:16	P1.8	GPIO Port 1.8	ENET_CRS	Reserved	Reserved	00
19:18	P1.9	GPIO Port 1.9	ENET_RXD0	Reserved	Reserved	00
21:20	P1.10	GPIO Port 1.10	ENET_RXD1	Reserved	Reserved	00
27:22	-	Reserved	Reserved	Reserved	Reserved	0
29:28	P1.14	GPIO Port 1.14	ENET_RX_ER	Reserved	Reserved	00
31:30	P1.15	GPIO Port 1.15	ENET_REF_CLK	Reserved	Reserved	00

8.5.4 Pin Function Select Register 3 (PINSEL3 - 0x4002 C00C)

The PINSEL3 register controls the functions of the upper half of Port 1. The direction control bit in the FIO1DIR register is effective only when the GPIO function is selected for a pin. For other functions, direction is controlled automatically.

Table 83. Pin function select register 3 (PINSEL3 - address 0x4002 C00C) bit description

PINSEL3	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P1.16[1]	GPIO Port 1.16	ENET_MDC	Reserved	Reserved	00
3:2	P1.17[1]	GPIO Port 1.17	ENET_MDIO	Reserved	Reserved	00
5:4	P1.18	GPIO Port 1.18	USB_UP_LED	PWM1.1	CAP1.0	00
7:6	P1.19	GPIO Port 1.19	MCOA0	$\overline{\text{USB\_PPWR}}$	CAP1.1	00
9:8	P1.20	GPIO Port 1.20	MCI0	PWM1.2	SCK0	00

Table 83. Pin function select register 3 (PINSEL3 - address 0x4002 C00C) bit description *...continued*

PINSEL3	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
11:10	P1.21[1]	GPIO Port 1.21	$\overline{\text{MCABORT}}$	PWM1.3	SSEL0	00
13:12	P1.22	GPIO Port 1.22	MCOB0	USB_PWRD	MAT1.0	00
15:14	P1.23	GPIO Port 1.23	MCI1	PWM1.4	MISO0	00
17:16	P1.24	GPIO Port 1.24	MCI2	PWM1.5	MOSI0	00
19:18	P1.25	GPIO Port 1.25	MCOA1	Reserved	MAT1.1	00
21:20	P1.26	GPIO Port 1.26	MCOB1	PWM1.6	CAP0.0	00
23:22	P1.27[1]	GPIO Port 1.27	CLKOUT	$\overline{\text{USB\_OVRCR}}$	CAP0.1	00
25:24	P1.28	GPIO Port 1.28	MCOA2	PCAP1.0	MAT0.0	00
27:26	P1.29	GPIO Port 1.29	MCOB2	PCAP1.1	MAT0.1	00
29:28	P1.30	GPIO Port 1.30	Reserved	V_{BUS}	AD0.4	00
31:30	P1.31	GPIO Port 1.31	Reserved	SCK1	AD0.5	00

[1] Not available on 80-pin package.

8.5.5 Pin Function Select Register 4 (PINSEL4 - 0x4002 C010)

The PINSEL4 register controls the functions of the lower half of Port 2. The direction control bit in the FIO2DIR register is effective only when the GPIO function is selected for a pin. For other functions, direction is controlled automatically.

Table 84. Pin function select register 4 (PINSEL4 - address 0x4002 C010) bit description

PINSEL4	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
1:0	P2.0	GPIO Port 2.0	PWM1.1	TXD1	Reserved	00
3:2	P2.1	GPIO Port 2.1	PWM1.2	RXD1	Reserved	00
5:4	P2.2	GPIO Port 2.2	PWM1.3	CTS1	Reserved [2]	00
7:6	P2.3	GPIO Port 2.3	PWM1.4	DCD1	Reserved [2]	00
9:8	P2.4	GPIO Port 2.4	PWM1.5	DSR1	Reserved [2]	00
11:10	P2.5	GPIO Port 2.5	PWM1.6	DTR1	Reserved [2]	00
13:12	P2.6	GPIO Port 2.6	PCAP1.0	RI1	Reserved [2]	00
15:14	P2.7	GPIO Port 2.7	RD2	RTS1	Reserved	00
17:16	P2.8	GPIO Port 2.8	TD2	TXD2	ENET_MDC	00
19:18	P2.9	GPIO Port 2.9	USB_CONNECT	RXD2	ENET_MDIO	00
21:20	P2.10	GPIO Port 2.10	$\overline{\text{EINT0}}$	NMI	Reserved	00
23:22	P2.11[1]	GPIO Port 2.11	$\overline{\text{EINT1}}$	Reserved	I2STX_CLK	00
25:24	P2.12[1]	GPIO Port 2.12	$\overline{\text{EINT2}}$	Reserved	I2STX_WS	00
27:26	P2.13[1]	GPIO Port 2.13	$\overline{\text{EINT3}}$	Reserved	I2STX_SDA	00
31:28	-	Reserved	Reserved	Reserved	Reserved	0

[1] Not available on 80-pin package.

[2] These pins support a debug trace function when selected via a development tool or by writing to the PINSEL10 register. See Section 8.5.8 “Pin Function Select Register 10 (PINSEL10 - 0x4002 C028)” for details.

8.5.6 Pin Function Select Register 7 (PINSEL7 - 0x4002 C01C)

The PINSEL7 register controls the functions of the upper half of Port 3. The direction control bit in the FIO3DIR register is effective only when the GPIO function is selected for a pin. For other functions, direction is controlled automatically.

Table 85. Pin function select register 7 (PINSEL7 - address 0x4002 C01C) bit description

PINSEL7	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
17:0	-	Reserved	Reserved	Reserved	Reserved	0
19:18	P3.25[1]	GPIO Port 3.25	Reserved	MAT0.0	PWM1.2	00
21:20	P3.26[1]	GPIO Port 3.26	STCLK	MAT0.1	PWM1.3	00
31:22	-	Reserved	Reserved	Reserved	Reserved	0

[1] Not available on 80-pin package.

8.5.7 Pin Function Select Register 9 (PINSEL9 - 0x4002 C024)

The PINSEL9 register controls the functions of the upper half of Port 4. The direction control bit in the FIO4DIR register is effective only when the GPIO function is selected for a pin. For other functions, direction is controlled automatically.

Table 86. Pin function select register 9 (PINSEL9 - address 0x4002 C024) bit description

PINSEL9	Pin name	Function when 00	Function when 01	Function when 10	Function when 11	Reset value
23:0	-	Reserved	Reserved	Reserved	Reserved	00
25:24	P4.28	GPIO Port 4.28	RX_MCLK	MAT2.0	TXD3	00
27:26	P4.29	GPIO Port 4.29	TX_MCLK	MAT2.1	RXD3	00
31:28	-	Reserved	Reserved	Reserved	Reserved	00

8.5.8 Pin Function Select Register 10 (PINSEL10 - 0x4002 C028)

Only bit 3 of this register is used to control the Trace function on pins P2.2 through P2.6.

Table 87. Pin function select register 10 (PINSEL10 - address 0x4002 C028) bit description

Bit	Symbol	Value	Description	Reset value
2:0	-	-	Reserved. Software should not write 1 to these bits.	NA
3	GPIO/TRACE		TPIU interface pins control.	0
		0	TPIU interface is disabled.	
		1	TPIU interface is enabled. TPIU signals are available on the pins hosting them regardless of the PINSEL4 content.	
31:4	-	-	Reserved. Software should not write 1 to these bits.	NA

8.5.9 Pin Mode select register 0 (PINMODE0 - 0x4002 C040)

This register controls pull-up/pull-down resistor configuration for Port 0 pins 0 to 15.

Table 88. Pin Mode select register 0 (PINMODE0 - address 0x4002 C040) bit description

PINMODE0	Symbol	Value	Description	Reset value
1:0	P0.00MODE		Port 0 pin 0 on-chip pull-up/down resistor control.	00
		00	P0.0 pin has a pull-up resistor enabled.	
		01	P0.0 pin has repeater mode enabled.	
		10	P0.0 pin has neither pull-up nor pull-down.	
		11	P0.0 has a pull-down resistor enabled.	
3:2	P0.01MODE		Port 0 pin 1 control, see P0.00MODE.	00
5:4	P0.02MODE		Port 0 pin 2 control, see P0.00MODE.	00
7:6	P0.03MODE		Port 0 pin 3 control, see P0.00MODE.	00
9:8	P0.04MODE[1]		Port 0 pin 4 control, see P0.00MODE.	00
11:10	P0.05MODE[1]		Port 0 pin 5 control, see P0.00MODE.	00
13:12	P0.06MODE		Port 0 pin 6 control, see P0.00MODE.	00
15:14	P0.07MODE		Port 0 pin 7 control, see P0.00MODE.	00
17:16	P0.08MODE		Port 0 pin 8 control, see P0.00MODE.	00
19:18	P0.09MODE		Port 0 pin 9control, see P0.00MODE.	00
21:20	P0.10MODE		Port 0 pin 10 control, see P0.00MODE.	00
23:22	P0.11MODE		Port 0 pin 11 control, see P0.00MODE.	00
29:24	-		Reserved.	NA
31:30	P0.15MODE		Port 0 pin 15 control, see P0.00MODE.	00

[1] Not available on 80-pin package.

8.5.10 Pin Mode select register 1 (PINMODE1 - 0x4002 C044)

This register controls pull-up/pull-down resistor configuration for Port 1 pins 16 to 26. For details see Section 8.4 “Pin mode select register values”.

Table 89. Pin Mode select register 1 (PINMODE1 - address 0x4002 C044) bit description

PINMODE1	Symbol	Description	Reset value
1:0	P0.16MODE	Port 0 pin 16 control, see P0.00MODE.	00
3:2	P0.17MODE	Port 0 pin 17 control, see P0.00MODE.	00
5:4	P0.18MODE	Port 0 pin 18 control, see P0.00MODE.	00
7:6	P0.19MODE[1]	Port 0 pin 19 control, see P0.00MODE.	00
9:8	P0.20MODE[1]	Port 0 pin 20control, see P0.00MODE.	00
11:10	P0.21MODE[1]	Port 0 pin 21 control, see P0.00MODE.	00
13:12	P0.22MODE	Port 0 pin 22 control, see P0.00MODE.	00
15:14	P0.23MODE[1]	Port 0 pin 23 control, see P0.00MODE.	00
17:16	P0.24MODE[1]	Port 0 pin 24 control, see P0.00MODE.	00
19:18	P0.25MODE	Port 0 pin 25 control, see P0.00MODE.	00

Table 89. Pin Mode select register 1 (PINMODE1 - address 0x4002 C044) bit description

PINMODE1	Symbol	Description	Reset value
21:20	P0.26MODE	Port 0 pin 26 control, see P0.00MODE.	00
29:22	-	Reserved. [2]	NA
31:30	-	Reserved.	NA

[1] Not available on 80-pin package.

[2] The pin mode cannot be selected for pins P0[27] to P0[30]. Pins P0[27] and P0[28] are dedicated I2C open-drain pins without pull-up/down. Pins P0[29] and P0[30] are USB specific pins without configurable pull-up or pull-down resistors. Pins P0[29] and P0[30] also must have the same direction since they operate as a unit for the USB function, see Section 9.5.1 “GPIO port Direction register FIOxDIR (FIO0DIR to FIO4DIR- 0x2009 C000 to 0x2009 C080)”.

8.5.11 Pin Mode select register 2 (PINMODE2 - 0x4002 C048)

This register controls pull-up/pull-down resistor configuration for Port 1 pins 0 to 15. For details see Section 8.4 “Pin mode select register values”.

Table 90. Pin Mode select register 2 (PINMODE2 - address 0x4002 C048) bit description

PINMODE2	Symbol	Description	Reset value
1:0	P1.00MODE	Port 1 pin 0 control, see P0.00MODE.	00
3:2	P1.01MODE	Port 1 pin 1 control, see P0.00MODE.	00
7:4	-	Reserved.	NA
9:8	P1.04MODE	Port 1 pin 4 control, see P0.00MODE.	00
15:10	-	Reserved.	NA
17:16	P1.08MODE	Port 1 pin 8 control, see P0.00MODE.	00
19:18	P1.09MODE	Port 1 pin 9 control, see P0.00MODE.	00
21:20	P1.10MODE	Port 1 pin 10 control, see P0.00MODE.	00
27:22	-	Reserved.	NA
29:28	P1.14MODE	Port 1 pin 14 control, see P0.00MODE.	00
31:30	P1.15MODE	Port 1 pin 15 control, see P0.00MODE.	00

8.5.12 Pin Mode select register 3 (PINMODE3 - 0x4002 C04C)

This register controls pull-up/pull-down resistor configuration for Port 1 pins 16 to 31. For details see Section 8.4 “Pin mode select register values”.

Table 91. Pin Mode select register 3 (PINMODE3 - address 0x4002 C04C) bit description

PINMODE3	Symbol	Description	Reset value
1:0	P1.16MODE[1]	Port 1 pin 16 control, see P0.00MODE.	00
3:2	P1.17MODE[1]	Port 1 pin 17 control, see P0.00MODE.	00
5:4	P1.18MODE	Port 1 pin 18 control, see P0.00MODE.	00
7:6	P1.19MODE	Port 1 pin 19 control, see P0.00MODE.	00
9:8	P1.20MODE	Port 1 pin 20 control, see P0.00MODE.	00
11:10	P1.21MODE[1]	Port 1 pin 21 control, see P0.00MODE.	00
13:12	P1.22MODE	Port 1 pin 22 control, see P0.00MODE.	00

Table 91. Pin Mode select register 3 (PINMODE3 - address 0x4002 C04C) bit description

PINMODE3	Symbol	Description	Reset value
15:14	P1.23MODE	Port 1 pin 23 control, see P0.00MODE.	00
17:16	P1.24MODE	Port 1 pin 24 control, see P0.00MODE.	00
19:18	P1.25MODE	Port 1 pin 25 control, see P0.00MODE.	00
21:20	P1.26MODE	Port 1 pin 26 control, see P0.00MODE.	00
23:22	P1.27MODE[1]	Port 1 pin 27 control, see P0.00MODE.	00
25:24	P1.28MODE	Port 1 pin 28 control, see P0.00MODE.	00
27:26	P1.29MODE	Port 1 pin 29 control, see P0.00MODE.	00
29:28	P1.30MODE	Port 1 pin 30 control, see P0.00MODE.	00
31:30	P1.31MODE	Port 1 pin 31 control, see P0.00MODE.	00

[1] Not available on 80-pin package.

8.5.13 Pin Mode select register 4 (PINMODE4 - 0x4002 C050)

This register controls pull-up/pull-down resistor configuration for Port 2 pins 0 to 15. For details see Section 8.4 “Pin mode select register values”.

Table 92. Pin Mode select register 4 (PINMODE4 - address 0x4002 C050) bit description

PINMODE4	Symbol	Description	Reset value
1:0	P2.00MODE	Port 2 pin 0 control, see P0.00MODE.	00
3:2	P2.01MODE	Port 2 pin 1 control, see P0.00MODE.	00
5:4	P2.02MODE	Port 2 pin 2 control, see P0.00MODE.	00
7:6	P2.03MODE	Port 2 pin 3 control, see P0.00MODE.	00
9:8	P2.04MODE	Port 2 pin 4 control, see P0.00MODE.	00
11:10	P2.05MODE	Port 2 pin 5 control, see P0.00MODE.	00
13:12	P2.06MODE	Port 2 pin 6 control, see P0.00MODE.	00
15:14	P2.07MODE	Port 2 pin 7 control, see P0.00MODE.	00
17:16	P2.08MODE	Port 2 pin 8 control, see P0.00MODE.	00
19:18	P2.09MODE	Port 2 pin 9 control, see P0.00MODE.	00
21:20	P2.10MODE	Port 2 pin 10 control, see P0.00MODE.	00
23:22	P2.11MODE[1]	Port 2 pin 11 control, see P0.00MODE.	00
25:24	P2.12MODE[1]	Port 2 pin 12 control, see P0.00MODE.	00
27:26	P2.13MODE[1]	Port 2 pin 13 control, see P0.00MODE.	00
31:28	-	Reserved.	NA

[1] Not available on 80-pin package.

8.5.14 Pin Mode select register 7 (PINMODE7 - 0x4002 C05C)

This register controls pull-up/pull-down resistor configuration for Port 3 pins 16 to 31. For details see Section 8.4 “Pin mode select register values”.

Table 93. Pin Mode select register 7 (PINMODE7 - address 0x4002 C05C) bit description

PINMODE7	Symbol	Description	Reset value
17:0	-	Reserved	NA
19:18	P3.25MODE[1]	Port 3 pin 25 control, see P0.00MODE.	00
21:20	P3.26MODE[1]	Port 3 pin 26 control, see P0.00MODE.	00
31:22	-	Reserved.	NA

[1] Not available on 80-pin package.

8.5.15 Pin Mode select register 9 (PINMODE9 - 0x4002 C064)

This register controls pull-up/pull-down resistor configuration for Port 4 pins 16 to 31. For details see Section 8.4 “Pin mode select register values”.

Table 94. Pin Mode select register 9 (PINMODE9 - address 0x4002 C064) bit description

PINMODE9	Symbol	Description	Reset value
23:0	-	Reserved.	NA
25:24	P4.28MODE	Port 4 pin 28 control, see P0.00MODE.	00
27:26	P4.29MODE	Port 4 pin 29 control, see P0.00MODE.	00
31:28	-	Reserved.	NA

8.5.16 Open Drain Pin Mode select register 0 (PINMODE_OD0 - 0x4002 C068)

This register controls the open drain mode for Port 0 pins. For details see Section 8.4 “Pin mode select register values”.

Table 95. Open Drain Pin Mode select register 0 (PINMODE_OD0 - address 0x4002 C068) bit description

PINMODE_OD0	Symbol	Value	Description	Reset value
0	P0.00OD[3]		Port 0 pin 0 open drain mode control.	0
		0	P0.0 pin is in the normal (not open drain) mode.	
		1	P0.0 pin is in the open drain mode.	
1	P0.01OD[3]		Port 0 pin 1 open drain mode control, see P0.00OD	0
2	P0.02OD		Port 0 pin 2 open drain mode control, see P0.00OD	0
3	P0.03OD		Port 0 pin 3 open drain mode control, see P0.00OD	0
4	P0.04OD		Port 0 pin 4 open drain mode control, see P0.00OD	0
5	P0.05OD		Port 0 pin 5 open drain mode control, see P0.00OD	0
6	P0.06OD		Port 0 pin 6 open drain mode control, see P0.00OD	0
7	P0.07OD		Port 0 pin 7 open drain mode control, see P0.00OD	0
8	P0.08OD		Port 0 pin 8 open drain mode control, see P0.00OD	0
9	P0.09OD		Port 0 pin 9 open drain mode control, see P0.00OD	0

Table 95. Open Drain Pin Mode select register 0 (PINMODE_OD0 - address 0x4002 C068) bit description *...continued*

PINMODE_OD0	Symbol	Value	Description	Reset value
10	P0.10OD[3]		Port 0 pin 10 open drain mode control, see P0.00OD	0
11	P0.11OD[3]		Port 0 pin 11 open drain mode control, see P0.00OD	0
14:12	-		Reserved.	NA
15	P0.15OD		Port 0 pin 15 open drain mode control, see P0.00OD	0
16	P0.16OD		Port 0 pin 16 open drain mode control, see P0.00OD	0
17	P0.17OD		Port 0 pin 17 open drain mode control, see P0.00OD	0
18	P0.18OD		Port 0 pin 18 open drain mode control, see P0.00OD	0
19	P0.19OD[3]		Port 0 pin 19 open drain mode control, see P0.00OD	0
20	P0.20OD[3]		Port 0 pin 20open drain mode control, see P0.00OD	0
21	P0.21OD		Port 0 pin 21 open drain mode control, see P0.00OD	0
22	P0.22OD		Port 0 pin 22 open drain mode control, see P0.00OD	0
23	P0.23OD		Port 0 pin 23 open drain mode control, see P0.00OD	0
24	P0.24OD		Port 0 pin 24open drain mode control, see P0.00OD	0
25	P0.25OD		Port 0 pin 25 open drain mode control, see P0.00OD	0
26	P0.26OD		Port 0 pin 26 open drain mode control, see P0.00OD	0
28:27	- [2]		Reserved.	NA
29	P0.29OD		Port 0 pin 29 open drain mode control, see P0.00OD	0
30	P0.30OD		Port 0 pin 30 open drain mode control, see P0.00OD	0
31	-		Reserved.	NA

[1] Not available on 80-pin package.

[2] Port 0 pins 27 and 28 should be set up using the I2CPADCFG register if they are used for an I^2C-bus. Bits 27 and 28 of PINMODE_OD0 do not have any affect on these pins, they are special open drain I^2C-bus compatible pins.

[3] Port 0 bits 1:0, 11:10, and 20:19 may potentially be used for I^2C-buses using standard port pins. If so, they should be configured for open drain mode via the related bits in PINMODE_OD0.

8.5.17 Open Drain Pin Mode select register 1 (PINMODE_OD1 - 0x4002 C06C)

This register controls the open drain mode for Port 1 pins. For details see Section 8.4 “Pin mode select register values”.

Table 96. Open Drain Pin Mode select register 1 (PINMODE_OD1 - address 0x4002 C06C) bit description

PINMODE_OD1	Symbol	Value	Description	Reset value
0	P1.00OD		Port 1 pin 0 open drain mode control.	0
		0	P1.0 pin is in the normal (not open drain) mode.	
		1	P1.0 pin is in the open drain mode.	
1	P1.01OD		Port 1 pin 1 open drain mode control, see P1.00OD	0
3:2	-		Reserved.	NA
4	P1.04OD		Port 1 pin 4 open drain mode control, see P1.00OD	0

Table 96. Open Drain Pin Mode select register 1 (PINMODE_OD1 - address 0x4002 C06C) bit description *...continued*

PINMODE_OD1	Symbol	Value	Description	Reset value
7:5	-		Reserved.	NA
8	P1.08OD		Port 1 pin 8 open drain mode control, see P1.00OD	0
9	P1.09OD		Port 1 pin 9 open drain mode control, see P1.00OD	0
10	P1.10OD		Port 1 pin 10 open drain mode control, see P1.00OD	0
13:11	-		Reserved.	NA
14	P1.14OD		Port 1 pin 14 open drain mode control, see P1.00OD	0
15	P1.15OD		Port 1 pin 15 open drain mode control, see P1.00OD	0
16	P1.16OD[1]		Port 1 pin 16 open drain mode control, see P1.00OD	0
17	P1.17OD[1]		Port 1 pin 17 open drain mode control, see P1.00OD	0
18	P1.18OD		Port 1 pin 18 open drain mode control, see P1.00OD	0
19	P1.19OD		Port 1 pin 19 open drain mode control, see P1.00OD	0
20	P1.20OD		Port 1 pin 20open drain mode control, see P1.00OD	0
21	P1.21OD[1]		Port 1 pin 21 open drain mode control, see P1.00OD	0
22	P1.22OD		Port 1 pin 22 open drain mode control, see P1.00OD	0
23	P1.23OD		Port 1 pin 23 open drain mode control, see P1.00OD	0
24	P1.24OD		Port 1 pin 24open drain mode control, see P1.00OD	0
25	P1.25OD		Port 1 pin 25 open drain mode control, see P1.00OD	0
26	P1.26OD		Port 1 pin 26 open drain mode control, see P1.00OD	0
27	P1.27OD[1]		Port 1 pin 27 open drain mode control, see P1.00OD	0
28	P1.28OD		Port 1 pin 28 open drain mode control, see P1.00OD	0
29	P1.29OD		Port 1 pin 29 open drain mode control, see P1.00OD	0
30	P1.30OD		Port 1 pin 30 open drain mode control, see P1.00OD	0
31	P1.31OD		Port 1 pin 31 open drain mode control.	0

[1] Not available on 80-pin package.

8.5.18 Open Drain Pin Mode select register 2 (PINMODE_OD2 - 0x4002 C070)

This register controls the open drain mode for Port 2 pins. For details see Section 8.4 “Pin mode select register values”.

Table 97. Open Drain Pin Mode select register 2 (PINMODE_OD2 - address 0x4002 C070) bit description

PINMODE_OD2	Symbol	Value	Description	Reset value
0	P2.00OD		Port 2 pin 0 open drain mode control.	0
		0	P2.0 pin is in the normal (not open drain) mode.	
		1	P2.0 pin is in the open drain mode.	
1	P2.01OD		Port 2 pin 1 open drain mode control, see P2.00OD	0
2	P2.02OD		Port 2 pin 2 open drain mode control, see P2.00OD	0
3	P2.03OD		Port 2 pin 3 open drain mode control, see P2.00OD	0
4	P2.04OD		Port 2 pin 4 open drain mode control, see P2.00OD	0

Table 97. Open Drain Pin Mode select register 2 (PINMODE_OD2 - address 0x4002 C070) bit description *...continued*

PINMODE_OD2	Symbol	Value	Description	Reset value
5	P2.05OD		Port 2 pin 5 open drain mode control, see P2.00OD	0
6	P2.06OD		Port 2 pin 6 open drain mode control, see P2.00OD	0
7	P2.07OD		Port 2 pin 7 open drain mode control, see P2.00OD	0
8	P2.08OD		Port 2 pin 8 open drain mode control, see P2.00OD	0
9	P2.09OD		Port 2 pin 9 open drain mode control, see P2.00OD	0
10	P2.10OD		Port 2 pin 10 open drain mode control, see P2.00OD	0
11	P2.11OD[1]		Port 2 pin 11 open drain mode control, see P2.00OD	0
12	P2.12OD[1]		Port 2 pin 12 open drain mode control, see P2.00OD	0
13	P2.13OD[1]		Port 2 pin 13 open drain mode control, see P2.00OD	0
31:14	-		Reserved.	NA

[1] Not available on 80-pin package.

8.5.19 Open Drain Pin Mode select register 3 (PINMODE_OD3 - 0x4002 C074)

This register controls the open drain mode for Port 3 pins. For details see Section 8.4 “Pin mode select register values”.

Table 98. Open Drain Pin Mode select register 3 (PINMODE_OD3 - address 0x4002 C074) bit description

PINMODE_OD3	Symbol	Value	Description	Reset value
24:0	-		Reserved.	NA
25	P3.25OD[1]		Port 3 pin 25 open drain mode control.	0
		0	P3.25 pin is in the normal (not open drain) mode.	
		1	P3.25 pin is in the open drain mode.	
26	P3.26OD[1]		Port 3 pin 26 open drain mode control, see P3.25OD	0
31:27	-		Reserved.	NA

[1] Not available on 80-pin package.

8.5.20 Open Drain Pin Mode select register 4 (PINMODE_OD4 - 0x4002 C078)

This register controls the open drain mode for Port 4 pins. For details see Section 8.4 “Pin mode select register values”.

Table 99. Open Drain Pin Mode select register 4 (PINMODE_OD4 - address 0x4002 C078) bit description

PINMODE_OD4	Symbol	Value	Description	Reset value
27:0	-		Reserved.	NA

Table 99. Open Drain Pin Mode select register 4 (PINMODE_OD4 - address 0x4002 C078) bit description *...continued*

PINMODE_OD4	Symbol	Value	Description	Reset value
28	P4.28OD		Port 4 pin 28 open drain mode control.	0
		0	P4.28 pin is in the normal (not open drain) mode.	
		1	P4.28 pin is in the open drain mode.	
29	P4.28OD		Port 4 pin 29 open drain mode control, see P4.28OD	0
31:30	-		Reserved.	NA

8.5.21 I²C Pin Configuration register (I2CPADCFG - 0x4002 C07C)

The I2CPADCFG register allows configuration of the I²C pins for the I2C0 interface only, in order to support various I²C-bus operating modes. For use in standard or Fast Mode I²C, the 4 bits in I2CPADCFG should be 0, the default value for this register. For Fast Mode Plus, the SDADRV0 and SCLDRV0 bits should be 1. For non-I²C use of these pins, it may be desirable to turn off I²C filtering and slew rate control by setting SDAI2C0 and SCLI2C0 to 1. See Table 100 below.

Table 100. I²C Pin Configuration register (I2CPADCFG - address 0x4002 C07C) bit description

I2CPADCFG	Symbol	Value	Description	Reset value
0	SDADRV0		Drive mode control for the SDA0 pin, P0.27.	0
		0	The SDA0 pin is in the standard drive mode.	
		1	The SDA0 pin is in Fast Mode Plus drive mode.	
1	SDAI2C0		I²C mode control for the SDA0 pin, P0.27.	0
		0	The SDA0 pin has I²C glitch filtering and slew rate control enabled.	
		1	The SDA0 pin has I²C glitch filtering and slew rate control disabled.	
2	SCLDRV0		Drive mode control for the SCL0 pin, P0.28.	0
		0	The SCL0 pin is in the standard drive mode.	
		1	The SCL0 pin is in Fast Mode Plus drive mode.	
3	SCLI2C0		I²C mode control for the SCL0 pin, P0.28.	0
		0	The SCL0 pin has I²C glitch filtering and slew rate control enabled.	
		1	The SCL0 pin has I²C glitch filtering and slew rate control disabled.	
31:4	-		Reserved.	NA

9.1 Basic configuration

GPIOs are configured using the following registers:

1. Power: always enabled.
2. Pins: See Section 8.3 for GPIO pins and their modes.
3. Wake-up: GPIO ports 0 and 2 can be used for wake-up if needed, see (Section 4.8.8).
4. Interrupts: Enable GPIO interrupts in IO0/2IntEnR (Table 115) or IO0/2IntEnF (Table 117). Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register.

9.2 Features

9.2.1 Digital I/O ports

- Accelerated GPIO functions:
  - GPIO registers are located on a peripheral AHB bus for fast I/O timing.
  - Mask registers allow treating sets of port bits as a group, leaving other bits unchanged.
  - All GPIO registers are byte, half-word, and word addressable.
  - Entire port value can be written in one instruction.
  - GPIO registers are accessible by the GPDMA.
- Bit-level set and clear registers allow a single instruction set or clear of any number of bits in one port.
- All GPIO registers support Cortex-M3 bit-banding.
- GPIO registers are accessible by the GPDMA controller to allow DMA of data to or from GPIOs, synchronized to any DMA request.
- Direction control of individual port bits.
- All I/Os default to input with pullup after reset.

9.2.2 Interrupt generating digital ports

- Port 0 and Port 2 can provide a single interrupt for any combination of port pins.
- Each port pin can be programmed to generate an interrupt on a rising edge, a falling edge, or both.
- Edge detection is asynchronous, so it may operate when clocks are not present, such as during Power-down mode. With this feature, level triggered interrupts are not needed.
- Each enabled interrupt contributes to a wake-up signal that can be used to bring the part out of Power-down mode.

- Registers provide a software view of pending rising edge interrupts, pending falling edge interrupts, and overall pending GPIO interrupts.
- GPIO0 and GPIO2 interrupts share the same position in the NVIC with External Interrupt 3.

9.3 Applications

- General purpose I/O
- Driving LEDs or other indicators
- Controlling off-chip devices
- Sensing digital inputs, detecting edges
- Bringing the part out of Power-down mode

9.4 Pin description

Table 101. GPIO pin description

Pin Name	Type	Description
P0[30:0][1]; P1[31:0][2]; P2[13:0]; P3[26:25]; P4[29:28]	Input/ Output	General purpose input/output. These are typically shared with other peripherals functions and will therefore not all be available in an application. Packaging options may affect the number of GPIOs available in a particular device. Some pins may be limited by requirements of the alternate functions of the pin. For example, the pins containing the I^2C0 functions are open-drain for any function selected on that pin. Details may be found in Section 7.1.1.

[1] P0[14:12] are not available.

[2] P1[2], P1[3], P1[7:5], P1[13:11] are not available.

9.5 Register description

Due to compatibility requirements with the LPC2300 series ARM7-based products, the LPC176x/5x implements portions of five 32-bit General Purpose I/O ports. Details on a specific GPIO port usage can be found in Section 8.3.

The registers in Table 102 represent the enhanced GPIO features available on all of the GPIO ports. These registers are located on an AHB bus for fast read and write timing. They can all be accessed in byte, half-word, and word sizes. A mask register allows access to a group of bits in a single GPIO port independently from other bits in the same port.

Table 102. GPIO register map (local bus accessible registers - enhanced GPIO features)

Generic Name	Description	Access	Reset value[1]	PORTn Register Name & Address
FIODIR	Fast GPIO Port Direction control register. This register individually controls the direction of each port pin.	R/W	0	FIO0DIR - 0x2009 C000 FIO1DIR - 0x2009 C020 FIO2DIR - 0x2009 C040 FIO3DIR - 0x2009 C060 FIO4DIR - 0x2009 C080
FIOMASK	Fast Mask register for port. Writes, sets, clears, and reads to port (done via writes to FIOPIN, FIOSET, and FIOCLR, and reads of FIOPIN) alter or return only the bits enabled by zeros in this register.	R/W	0	FIO0MASK - 0x2009 C010 FIO1MASK - 0x2009 C030 FIO2MASK - 0x2009 C050 FIO3MASK - 0x2009 C070 FIO4MASK - 0x2009 C090
FIOPIN	Fast Port Pin value register using FIOMASK. The current state of digital port pins can be read from this register, regardless of pin direction or alternate function selection (as long as pins are not configured as an input to ADC). The value read is masked by ANDing with inverted FIOMASK. Writing to this register places corresponding values in all bits enabled by zeros in FIOMASK. **Important:** if an FIOPIN register is read, its bit(s) masked with 1 in the FIOMASK register will be read as 0 regardless of the physical pin state.	R/W	0	FIO0PIN - 0x2009 C014 FIO1PIN - 0x2009 C034 FIO2PIN - 0x2009 C054 FIO3PIN - 0x2009 C074 FIO4PIN - 0x2009 C094
FIOSET	Fast Port Output Set register using FIOMASK. This register controls the state of output pins. Writing 1s produces highs at the corresponding port pins. Writing 0s has no effect. Reading this register returns the current contents of the port output register. Only bits enabled by 0 in FIOMASK can be altered.	R/W	0	FIO0SET - 0x2009 C018 FIO1SET - 0x2009 C038 FIO2SET - 0x2009 C058 FIO3SET - 0x2009 C078 FIO4SET - 0x2009 C098
FIOCLR	Fast Port Output Clear register using FIOMASK. This register controls the state of output pins. Writing 1s produces lows at the corresponding port pins. Writing 0s has no effect. Only bits enabled by 0 in FIOMASK can be altered.	WO	0	FIO0CLR - 0x2009 C01C FIO1CLR - 0x2009 C03C FIO2CLR - 0x2009 C05C FIO3CLR - 0x2009 C07C FIO4CLR - 0x2009 C09C

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

Table 103. GPIO interrupt register map

Generic Name	Description	Access	Reset value[1]	PORTn Register Name & Address
IntEnR	GPIO Interrupt Enable for Rising edge.	R/W	0	IO0IntEnR - 0x4002 8090 IO2IntEnR - 0x4002 80B0
IntEnF	GPIO Interrupt Enable for Falling edge.	R/W	0	IO0IntEnR - 0x4002 8094 IO2IntEnR - 0x4002 80B4
IntStatR	GPIO Interrupt Status for Rising edge.	RO	0	IO0IntStatR - 0x4002 8084 IO2IntStatR - 0x4002 80A4
IntStatF	GPIO Interrupt Status for Falling edge.	RO	0	IO0IntStatF - 0x4002 8088 IO2IntStatF - 0x4002 80A8
IntClr	GPIO Interrupt Clear.	WO	0	IO0IntClr - 0x4002 808C IO2IntClr - 0x4002 80AC
IntStatus	GPIO overall Interrupt Status.	RO	0	IOIntStatus - 0x4002 8080

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

9.5.1 GPIO port Direction register FIOxDIR (FIO0DIR to FIO4DIR- 0x2009 C000 to 0x2009 C080)

This word accessible register is used to control the direction of the pins when they are configured as GPIO port pins. Direction bit for any pin must be set according to the pin functionality.

Note that GPIO pins P0.29 and P0.30 are shared with the USB_D+ and USB_D- pins and must have the same direction. If either FIO0DIR bit 29 or 30 are configured as zero, both P0.29 and P0.30 will be inputs. If both FIO0DIR bits 29 and 30 are ones, both P0.29 and P0.30 will be outputs.

Table 104. Fast GPIO port Direction register FIO0DIR to FIO4DIR - addresses 0x2009 C000 to 0x2009 C080) bit description

Bit	Symbol	Value	Description	Reset value
31:0	FIO0DIR FIO1DIR FIO2DIR FIO3DIR FIO4DIR		Fast GPIO Direction PORTx control bits. Bit 0 in FIOxDIR controls pin Px.0, bit 31 in FIOxDIR controls pin Px.31.	0x0
		0	Controlled pin is input.	
		1	Controlled pin is output.	

Aside from the 32-bit long and word only accessible FIODIR register, every fast GPIO port can also be controlled via several byte and half-word accessible registers listed in Table 105, too. Next to providing the same functions as the FIODIR register, these additional registers allow easier and faster access to the physical port pins.

Table 105. Fast GPIO port Direction control byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxDIR0	Fast GPIO Port x Direction control register 0. Bit 0 in FIOxDIR0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0DIR0 - 0x2009 C000 FIO1DIR0 - 0x2009 C020 FIO2DIR0 - 0x2009 C040 FIO3DIR0 - 0x2009 C060 FIO4DIR0 - 0x2009 C080
FIOxDIR1	Fast GPIO Port x Direction control register 1. Bit 0 in FIOxDIR1 register corresponds to pin Px.8 … bit 7 to pin Px.15.	8 (byte) R/W	0x00	FIO0DIR1 - 0x2009 C001 FIO1DIR1 - 0x2009 C021 FIO2DIR1 - 0x2009 C041 FIO3DIR1 - 0x2009 C061 FIO4DIR1 - 0x2009 C081
FIOxDIR2	Fast GPIO Port x Direction control register 2. Bit 0 in FIOxDIR2 register corresponds to pin Px.16 … bit 7 to pin Px.23.	8 (byte) R/W	0x00	FIO0DIR2 - 0x2009 C002 FIO1DIR2 - 0x2009 C022 FIO2DIR2 - 0x2009 C042 FIO3DIR2 - 0x2009 C062 FIO4DIR2 - 0x2009 C082
FIOxDIR3	Fast GPIO Port x Direction control register 3. Bit 0 in FIOxDIR3 register corresponds to pin Px.24 … bit 7 to pin Px.31.	8 (byte) R/W	0x00	FIO0DIR3 - 0x2009 C003 FIO1DIR3 - 0x2009 C023 FIO2DIR3 - 0x2009 C043 FIO3DIR3 - 0x2009 C063 FIO4DIR3 - 0x2009 C083
FIOxDIRL	Fast GPIO Port x Direction control Lower half-word register. Bit 0 in FIOxDIRL register corresponds to pin Px.0 … bit 15 to pin Px.15.	16 (half-word) R/W	0x0000	FIO0DIRL - 0x2009 C000 FIO1DIRL - 0x2009 C020 FIO2DIRL - 0x2009 C040 FIO3DIRL - 0x2009 C060 FIO4DIRL - 0x2009 C080
FIOxDIRU	Fast GPIO Port x Direction control Upper half-word register. Bit 0 in FIOxDIRU register corresponds to Px.16 … bit 15 to Px.31.	16 (half-word) R/W	0x0000	FIO0DIRU - 0x2009 C002 FIO1DIRU - 0x2009 C022 FIO2DIRU - 0x2009 C042 FIO3DIRU - 0x2009 C062 FIO4DIRU - 0x2009 C082

9.5.2 GPIO port output Set register FIOxSET (FIO0SET to FIO4SET - 0x2009 C018 to 0x2009 C098)

This register is used to produce a HIGH level output at the port pins configured as GPIO in an OUTPUT mode. Writing 1 produces a HIGH level at the corresponding port pins. Writing 0 has no effect. If any pin is configured as an input or a secondary function, writing 1 to the corresponding bit in the FIOxSET has no effect.

Reading the FIOxSET register returns the value of this register, as determined by previous writes to FIOxSET and FIOxCLR (or FIOxPIN as noted above). This value does not reflect the effect of any outside world influence on the I/O pins.

Access to a port pin via the FIOxSET register is conditioned by the corresponding bit of the FIOxMASK register (see Section 9.5.5).

Table 106. Fast GPIO port output Set register (FIO0SET to FIO4SET - addresses 0x2009 C018 to 0x2009 C098) bit description

Bit	Symbol	Value	Description	Reset value
31:0	FIO0SET FIO1SET FIO2SET FIO3SET FIO4SET		Fast GPIO output value Set bits. Bit 0 in FIOxSET controls pin Px.0, bit 31 in FIOxSET controls pin Px.31.	0x0
		0	Controlled pin output is unchanged.	
		1	Controlled pin output is set to HIGH.	

Aside from the 32-bit long and word only accessible FIOxSET register, every fast GPIO port can also be controlled via several byte and half-word accessible registers listed in Table 107, too. Next to providing the same functions as the FIOxSET register, these additional registers allow easier and faster access to the physical port pins.

Table 107. Fast GPIO port output Set byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxSET0	Fast GPIO Port x output Set register 0. Bit 0 in FIOxSET0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0SET0 - 0x2009 C018 FIO1SET0 - 0x2009 C038 FIO2SET0 - 0x2009 C058 FIO3SET0 - 0x2009 C078 FIO4SET0 - 0x2009 C098
FIOxSET1	Fast GPIO Port x output Set register 1. Bit 0 in FIOxSET1 register corresponds to pin Px.8 … bit 7 to pin Px.15.	8 (byte) R/W	0x00	FIO0SET1 - 0x2009 C019 FIO1SET1 - 0x2009 C039 FIO2SET1 - 0x2009 C059 FIO3SET1 - 0x2009 C079 FIO4SET1 - 0x2009 C099
FIOxSET2	Fast GPIO Port x output Set register 2. Bit 0 in FIOxSET2 register corresponds to pin Px.16 … bit 7 to pin Px.23.	8 (byte) R/W	0x00	FIO0SET2 - 0x2009 C01A FIO1SET2 - 0x2009 C03A FIO2SET2 - 0x2009 C05A FIO3SET2 - 0x2009 C07A FIO4SET2 - 0x2009 C09A
FIOxSET3	Fast GPIO Port x output Set register 3. Bit 0 in FIOxSET3 register corresponds to pin Px.24 … bit 7 to pin Px.31.	8 (byte) R/W	0x00	FIO0SET3 - 0x2009 C01B FIO1SET3 - 0x2009 C03B FIO2SET3 - 0x2009 C05B FIO3SET3 - 0x2009 C07B FIO4SET3 - 0x2009 C09B
FIOxSETL	Fast GPIO Port x output Set Lower half-word register. Bit 0 in FIOxSETL register corresponds to pin Px.0 … bit 15 to pin Px.15.	16 (half-word) R/W	0x0000	FIO0SETL - 0x2009 C018 FIO1SETL - 0x2009 C038 FIO2SETL - 0x2009 C058 FIO3SETL - 0x2009 C078 FIO4SETL - 0x2009 C098
FIOxSETU	Fast GPIO Port x output Set Upper half-word register. Bit 0 in FIOxSETU register corresponds to Px.16 … bit 15 to Px.31.	16 (half-word) R/W	0x0000	FIO0SETU - 0x2009 C01A FIO1SETU - 0x2009 C03A FIO2SETU - 0x2009 C05A FIO3SETU - 0x2009 C07A FIO4SETU - 0x2009 C09A

9.5.3 GPIO port output Clear register FIOxCLR (FIO0CLR to FIO4CLR- 0x2009 C01C to 0x2009 C09C)

This register is used to produce a LOW level output at port pins configured as GPIO in an OUTPUT mode. Writing 1 produces a LOW level at the corresponding port pin and clears the corresponding bit in the FIOxSET register. Writing 0 has no effect. If any pin is configured as an input or a secondary function, writing to FIOxCLR has no effect.

Access to a port pin via the FIOxCLR register is conditioned by the corresponding bit of the FIOxMASK register (see Section 9.5.5).

Table 108. Fast GPIO port output Clear register (FIO0CLR to FIO4CLR- addresses 0x2009 C01C to 0x2009 C09C) bit description

Bit	Symbol	Value	Description	Reset value
31:0	FIO0CLR FIO1CLR FIO2CLR FIO3CLR FIO4CLR		Fast GPIO output value Clear bits. Bit 0 in FIOxCLR controls pin Px.0, bit 31 controls pin Px.31.	0x0
		0	Controlled pin output is unchanged.	
		1	Controlled pin output is set to LOW.	

Aside from the 32-bit long and word only accessible FIOxCLR register, every fast GPIO port can also be controlled via several byte and half-word accessible registers listed in Table 109, too. Next to providing the same functions as the FIOxCLR register, these additional registers allow easier and faster access to the physical port pins.

Table 109. Fast GPIO port output Clear byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxCLR0	Fast GPIO Port x output Clear register 0. Bit 0 in FIOxCLR0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) WO	0x00	FIO0CLR0 - 0x2009 C01C FIO1CLR0 - 0x2009 C03C FIO2CLR0 - 0x2009 C05C FIO3CLR0 - 0x2009 C07C FIO4CLR0 - 0x2009 C09C
FIOxCLR1	Fast GPIO Port x output Clear register 1. Bit 0 in FIOxCLR1 register corresponds to pin Px.8 … bit 7 to pin Px.15.	8 (byte) WO	0x00	FIO0CLR1 - 0x2009 C01D FIO1CLR1 - 0x2009 C03D FIO2CLR1 - 0x2009 C05D FIO3CLR1 - 0x2009 C07D FIO4CLR1 - 0x2009 C09D
FIOxCLR2	Fast GPIO Port x output Clear register 2. Bit 0 in FIOxCLR2 register corresponds to pin Px.16 … bit 7 to pin Px.23.	8 (byte) WO	0x00	FIO0CLR2 - 0x2009 C01E FIO1CLR2 - 0x2009 C03E FIO2CLR2 - 0x2009 C05E FIO3CLR2 - 0x2009 C07E FIO4CLR2 - 0x2009 C09E

Table 109. Fast GPIO port output Clear byte and half-word accessible register description *…continued*

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxCLR3	Fast GPIO Port x output Clear register 3. Bit 0 in FIOxCLR3 register corresponds to pin Px.24 … bit 7 to pin Px.31.	8 (byte) WO	0x00	FIO0CLR3 - 0x2009 C01F FIO1CLR3 - 0x2009 C03F FIO2CLR3 - 0x2009 C05F FIO3CLR3 - 0x2009 C07F FIO4CLR3 - 0x2009 C09F
FIOxCLRL	Fast GPIO Port x output Clear Lower half-word register. Bit 0 in FIOxCLRL register corresponds to pin Px.0 … bit 15 to pin Px.15.	16 (half-word) WO	0x0000	FIO0CLRL - 0x2009 C01C FIO1CLRL - 0x2009 C03C FIO2CLRL - 0x2009 C05C FIO3CLRL - 0x2009 C07C FIO4CLRL - 0x2009 C09C
FIOxCLRU	Fast GPIO Port x output Clear Upper half-word register. Bit 0 in FIOxCLRU register corresponds to pin Px.16 … bit 15 to Px.31.	16 (half-word) WO	0x0000	FIO0CLRU - 0x2009 C01E FIO1CLRU - 0x2009 C03E FIO2CLRU - 0x2009 C05E FIO3CLRU - 0x2009 C07E FIO4CLRU - 0x2009 C09E

9.5.4 GPIO port Pin value register FIOxPIN (FIO0PIN to FIO4PIN- 0x2009 C014 to 0x2009 C094)

This register provides the value of port pins that are configured to perform only digital functions. The register will give the logic value of the pin regardless of whether the pin is configured for input or output, or as GPIO or an alternate digital function. As an example, a particular port pin may have GPIO input, GPIO output, UART receive, and PWM output as selectable functions. Any configuration of that pin will allow its current logic state to be read from the corresponding FIOxPIN register.

If a pin has an analog function as one of its options, the pin state cannot be read if the analog configuration is selected. Selecting the pin as an A/D input disconnects the digital features of the pin. In that case, the pin value read in the FIOxPIN register is not valid.

Writing to the FIOxPIN register stores the value in the port output register, bypassing the need to use both the FIOxSET and FIOxCLR registers to obtain the entire written value. This feature should be used carefully in an application since it affects the entire port.

Access to a port pin via the FIOxPIN register is conditioned by the corresponding bit of the FIOxMASK register (see Section 9.5.5).

Only pins masked with zeros in the Mask register (see Section 9.5.5) will be correlated to the current content of the Fast GPIO port pin value register.

Table 110. Fast GPIO port Pin value register (FIO0PIN to FIO4PIN- addresses 0x2009 C014 to 0x2009 C094) bit description

Bit	Symbol	Value	Description	Reset value
31:0	FIO0VAL FIO1VAL FIO2VAL FIO3VAL FIO4VAL		Fast GPIO output value bits. Bit 0 corresponds to pin Px.0, bit 31 corresponds to pin Px.31. Only bits also set to 0 in the FIOxMASK register are affected by a write or show the pin's actual logic state.	0x0
		0	Reading a 0 indicates that the port pin's current state is LOW. Writing a 0 sets the output register value to LOW.	
		1	Reading a 1 indicates that the port pin's current state is HIGH. Writing a 1 sets the output register value to HIGH.	

Aside from the 32-bit long and word only accessible FIOxPIN register, every fast GPIO port can also be controlled via several byte and half-word accessible register listed in Table 111, too. Next to providing the same functions as the FIOxPIN register, these additional registers allow easier and faster access to the physical port pins.

Table 111. Fast GPIO port Pin value byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxPIN0	Fast GPIO Port x Pin value register 0. Bit 0 in FIOxPIN0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) R/W	0x00	FIO0PIN0 - 0x2009 C014 FIO1PIN0 - 0x2009 C034 FIO2PIN0 - 0x2009 C054 FIO3PIN0 - 0x2009 C074 FIO4PIN0 - 0x2009 C094
FIOxPIN1	Fast GPIO Port x Pin value register 1. Bit 0 in FIOxPIN1 register corresponds to pin Px.8 … bit 7 to pin Px.15.	8 (byte) R/W	0x00	FIO0PIN1 - 0x2009 C015 FIO1PIN1 - 0x2009 C035 FIO2PIN1 - 0x2009 C055 FIO3PIN1 - 0x2009 C075 FIO4PIN1 - 0x2009 C095
FIOxPIN2	Fast GPIO Port x Pin value register 2. Bit 0 in FIOxPIN2 register corresponds to pin Px.16 … bit 7 to pin Px.23.	8 (byte) R/W	0x00	FIO0PIN2 - 0x2009 C016 FIO1PIN2 - 0x2009 C036 FIO2PIN2 - 0x2009 C056 FIO3PIN2 - 0x2009 C076 FIO4PIN2 - 0x2009 C096

Table 111. Fast GPIO port Pin value byte and half-word accessible register description *…continued*

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxPIN3	Fast GPIO Port x Pin value register 3. Bit 0 in FIOxPIN3 register corresponds to pin Px.24 … bit 7 to pin Px.31.	8 (byte) R/W	0x00	FIO0PIN3 - 0x2009 C017 FIO1PIN3 - 0x2009 C037 FIO2PIN3 - 0x2009 C057 FIO3PIN3 - 0x2009 C077 FIO4PIN3 - 0x2009 C097
FIOxPINL	Fast GPIO Port x Pin value Lower half-word register. Bit 0 in FIOxPINL register corresponds to pin Px.0 … bit 15 to pin Px.15.	16 (half-word) R/W	0x0000	FIO0PINL - 0x2009 C014 FIO1PINL - 0x2009 C034 FIO2PINL - 0x2009 C054 FIO3PINL - 0x2009 C074 FIO4PINL - 0x2009 C094
FIOxPINU	Fast GPIO Port x Pin value Upper half-word register. Bit 0 in FIOxPINU register corresponds to pin Px.16 … bit 15 to Px.31.	16 (half-word) R/W	0x0000	FIO0PINU - 0x2009 C016 FIO1PINU - 0x2009 C036 FIO2PINU - 0x2009 C056 FIO3PINU - 0x2009 C076 FIO4PINU - 0x2009 C096

9.5.5 Fast GPIO port Mask register FIOxMASK (FIO0MASK to FIO4MASK - 0x2009 C010 to 0x2009 C090)

This register is used to select port pins that will and will not be affected by write accesses to the FIOxPIN, FIOxSET or FIOxCLR register. Mask register also filters out port's content when the FIOxPIN register is read.

A zero in this register's bit enables an access to the corresponding physical pin via a read or write access. If a bit in this register is one, corresponding pin will not be changed with write access and if read, will not be reflected in the updated FIOxPIN register. For software examples, see Section 9.6.

Table 112. Fast GPIO port Mask register (FIO0MASK to FIO4MASK - addresses 0x2009 C010 to 0x2009 C090) bit description

Bit	Symbol	Value	Description	Reset value
31:0	FIO0MASK FIO1MASK FIO2MASK FIO3MASK FIO4MASK		Fast GPIO physical pin access control.	0x0
		0	Controlled pin is affected by writes to the port's FIOxSET, FIOxCLR, and FIOxPIN register(s). Current state of the pin can be read from the FIOxPIN register.	
		1	Controlled pin is not affected by writes into the port's FIOxSET, FIOxCLR and FIOxPIN register(s). When the FIOxPIN register is read, this bit will not be updated with the state of the physical pin.	

Aside from the 32-bit long and word only accessible FIOxMASK register, every fast GPIO port can also be controlled via several byte and half-word accessible registers listed in Table 113, too. Next to providing the same functions as the FIOxMASK register, these additional registers allow easier and faster access to the physical port pins.

Table 113. Fast GPIO port Mask byte and half-word accessible register description

Generic Register name	Description	Register length (bits) & access	Reset value	PORTn Register Address & Name
FIOxMASK0	Fast GPIO Port x Mask register 0. Bit 0 in FIOxMASK0 register corresponds to pin Px.0 … bit 7 to pin Px.7.	8 (byte) R/W	0x0	FIO0MASK0 - 0x2009 C010 FIO1MASK0 - 0x2009 C030 FIO2MASK0 - 0x2009 C050 FIO3MASK0 - 0x2009 C070 FIO4MASK0 - 0x2009 C090
FIOxMASK1	Fast GPIO Port x Mask register 1. Bit 0 in FIOxMASK1 register corresponds to pin Px.8 … bit 7 to pin Px.15.	8 (byte) R/W	0x0	FIO0MASK1 - 0x2009 C011 FIO1MASK1 - 0x2009 C031 FIO2MASK1 - 0x2009 C051 FIO3MASK1 - 0x2009 C071 FIO4MASK1 - 0x2009 C091
FIOxMASK2	Fast GPIO Port x Mask register 2. Bit 0 in FIOxMASK2 register corresponds to pin Px.16 … bit 7 to pin Px.23.	8 (byte) R/W	0x0	FIO0MASK2 - 0x2009 C012 FIO1MASK2 - 0x2009 C032 FIO2MASK2 - 0x2009 C052 FIO3MASK2 - 0x2009 C072 FIO4MASK2 - 0x2009 C092
FIOxMASK3	Fast GPIO Port x Mask register 3. Bit 0 in FIOxMASK3 register corresponds to pin Px.24 … bit 7 to pin Px.31.	8 (byte) R/W	0x0	FIO0MASK3 - 0x2009 C013 FIO1MASK3 - 0x2009 C033 FIO2MASK3 - 0x2009 C053 FIO3MASK3 - 0x2009 C073 FIO4MASK3 - 0x2009 C093
FIOxMASKL	Fast GPIO Port x Mask Lower half-word register. Bit 0 in FIOxMASKL register corresponds to pin Px.0 … bit 15 to pin Px.15.	16 (half-word) R/W	0x0	FIO0MASKL - 0x2009 C010 FIO1MASKL - 0x2009 C030 FIO2MASKL - 0x2009 C050 FIO3MASKL - 0x2009 C070 FIO4MASKL - 0x2009 C090
FIOxMASKU	Fast GPIO Port x Mask Upper half-word register. Bit 0 in FIOxMASKU register corresponds to pin Px.16 … bit 15 to Px.31.	16 (half-word) R/W	0x0	FIO0MASKU - 0x2009 C012 FIO1MASKU - 0x2009 C032 FIO2MASKU - 0x2009 C052 FIO3MASKU - 0x2009 C072 FIO4MASKU - 0x2009 C092

9.5.6 GPIO interrupt registers

The following registers configure the pins of Port 0 and Port 2 to generate interrupts.

9.5.6.1 GPIO overall Interrupt Status register (IOIntStatus - 0x4002 8080)

This read-only register indicates the presence of interrupt pending on all of the GPIO ports that support GPIO interrupts. Only status one bit per port is required.

Table 114. GPIO overall Interrupt Status register (IOIntStatus - address 0x4002 8080) bit description

Bit	Symbol	Value	Description	Reset value
0	P0Int		Port 0 GPIO interrupt pending.	0
		0	There are no pending interrupts on Port 0.	
		1	There is at least one pending interrupt on Port 0.	
1	-	-	Reserved. The value read from a reserved bit is not defined.	NA
2	P2Int		Port 2 GPIO interrupt pending.	0
		0	There are no pending interrupts on Port 2.	
		1	There is at least one pending interrupt on Port 2.	
31:2	-	-	Reserved. The value read from a reserved bit is not defined.	NA

9.5.6.2 GPIO Interrupt Enable for port 0 Rising Edge (IO0IntEnR - 0x4002 8090)

Each bit in these read-write registers enables the rising edge interrupt for the corresponding port 0 pin.

Table 115. GPIO Interrupt Enable for port 0 Rising Edge (IO0IntEnR - 0x4002 8090) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0ER		Enable rising edge interrupt for P0.0.	0
		0	Rising edge interrupt is disabled on P0.0.	
		1	Rising edge interrupt is enabled on P0.0.	
1	P0.1ER		Enable rising edge interrupt for P0.1.	0
2	P0.2ER		Enable rising edge interrupt for P0.2.	0
3	P0.3ER		Enable rising edge interrupt for P0.3.	0
4	P0.4ER[1]		Enable rising edge interrupt for P0.4.	0
5	P0.5ER[1]		Enable rising edge interrupt for P0.5.	0
6	P0.6ER		Enable rising edge interrupt for P0.6.	0
7	P0.7ER		Enable rising edge interrupt for P0.7.	0
8	P0.8ER		Enable rising edge interrupt for P0.8.	0
9	P0.9ER		Enable rising edge interrupt for P0.9.	0
10	P0.10ER		Enable rising edge interrupt for P0.10.	0
11	P0.11ER		Enable rising edge interrupt for P0.11.	0
14:12	-		Reserved	NA
15	P0.15ER		Enable rising edge interrupt for P0.15.	0
16	P0.16ER		Enable rising edge interrupt for P0.16.	0

Table 115. GPIO Interrupt Enable for port 0 Rising Edge (IO0IntEnR - 0x4002 8090) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
17	P0.17ER		Enable rising edge interrupt for P0.17.	0
18	P0.18ER		Enable rising edge interrupt for P0.18.	0
19	P0.19ER[1]		Enable rising edge interrupt for P0.19.	0
20	P0.20ER[1]		Enable rising edge interrupt for P0.20.	0
21	P0.21ER[1]		Enable rising edge interrupt for P0.21.	0
22	P0.22ER		Enable rising edge interrupt for P0.22.	0
23	P0.23ER[1]		Enable rising edge interrupt for P0.23.	0
24	P0.24ER[1]		Enable rising edge interrupt for P0.24.	0
25	P0.25ER		Enable rising edge interrupt for P0.25.	0
26	P0.26ER		Enable rising edge interrupt for P0.26.	0
27	P0.27ER[1]		Enable rising edge interrupt for P0.27.	0
28	P0.28ER[1]		Enable rising edge interrupt for P0.28.	0
29	P0.29ER		Enable rising edge interrupt for P0.29.	0
30	P0.30ER		Enable rising edge interrupt for P0.30.	0
31	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.3 GPIO Interrupt Enable for port 2 Rising Edge (IO2IntEnR - 0x4002 80B0)

Each bit in these read-write registers enables the rising edge interrupt for the corresponding port 2 pin.

Table 116. GPIO Interrupt Enable for port 2 Rising Edge (IO2IntEnR - 0x4002 80B0) bit description

Bit	Symbol	Value	Description	Reset value
0	P2.0ER		Enable rising edge interrupt for P2.0.	0
		0	Rising edge interrupt is disabled on P2.0.	
		1	Rising edge interrupt is enabled on P2.0.	
1	P2.1ER		Enable rising edge interrupt for P2.1.	0
2	P2.2ER		Enable rising edge interrupt for P2.2.	0
3	P2.3ER		Enable rising edge interrupt for P2.3.	0
4	P2.4ER		Enable rising edge interrupt for P2.4.	0
5	P2.5ER		Enable rising edge interrupt for P2.5.	0
6	P2.6ER		Enable rising edge interrupt for P2.6.	0
7	P2.7ER		Enable rising edge interrupt for P2.7.	0
8	P2.8ER		Enable rising edge interrupt for P2.8.	0
9	P2.9ER		Enable rising edge interrupt for P2.9.	0
10	P2.10ER		Enable rising edge interrupt for P2.10.	0
11	P2.11ER[1]		Enable rising edge interrupt for P2.11.	0

Table 116. GPIO Interrupt Enable for port 2 Rising Edge (IO2IntEnR - 0x4002 80B0) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
12	P2.12ER[1]		Enable rising edge interrupt for P2.12.	0
13	P2.13ER[1]		Enable rising edge interrupt for P2.13.	0
31:14	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.4 GPIO Interrupt Enable for port 0 Falling Edge (IO0IntEnF - 0x4002 8094)

Each bit in these read-write registers enables the falling edge interrupt for the corresponding GPIO port 0 pin.

Table 117. GPIO Interrupt Enable for port 0 Falling Edge (IO0IntEnF - address 0x4002 8094) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0EF		Enable falling edge interrupt for P0.0	0
		0	Falling edge interrupt is disabled on P0.0.	
		1	Falling edge interrupt is enabled on P0.0.	
1	P0.1EF		Enable falling edge interrupt for P0.1.	0
2	P0.2EF		Enable falling edge interrupt for P0.2.	0
3	P0.3EF		Enable falling edge interrupt for P0.3.	0
4	P0.4EF[1]		Enable falling edge interrupt for P0.4.	0
5	P0.5EF[1]		Enable falling edge interrupt for P0.5.	0
6	P0.6EF		Enable falling edge interrupt for P0.6.	0
7	P0.7EF		Enable falling edge interrupt for P0.7.	0
8	P0.8EF		Enable falling edge interrupt for P0.8.	0
9	P0.9EF		Enable falling edge interrupt for P0.9.	0
10	P0.10EF		Enable falling edge interrupt for P0.10.	0
11	P0.11EF		Enable falling edge interrupt for P0.11.	0
14:12	-		Reserved.	NA
15	P0.15EF		Enable falling edge interrupt for P0.15.	0
16	P0.16EF		Enable falling edge interrupt for P0.16.	0
17	P0.17EF		Enable falling edge interrupt for P0.17.	0
18	P0.18EF		Enable falling edge interrupt for P0.18.	0
19	P0.19EF[1]		Enable falling edge interrupt for P0.19.	0
20	P0.20EF[1]		Enable falling edge interrupt for P0.20.	0
21	P0.21EF[1]		Enable falling edge interrupt for P0.21.	0
22	P0.22EF		Enable falling edge interrupt for P0.22.	0
23	P0.23EF[1]		Enable falling edge interrupt for P0.23.	0
24	P0.24EF[1]		Enable falling edge interrupt for P0.24.	0
25	P0.25EF		Enable falling edge interrupt for P0.25.	0
26	P0.26EF		Enable falling edge interrupt for P0.26.	0
27	P0.27EF[1]		Enable falling edge interrupt for P0.27.	0

Table 117. GPIO Interrupt Enable for port 0 Falling Edge (IO0IntEnF - address 0x4002 8094) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
28	P0.28EF[1]		Enable falling edge interrupt for P0.28.	0
29	P0.29EF		Enable falling edge interrupt for P0.29.	0
30	P0.30EF		Enable falling edge interrupt for P0.30.	0
31	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.5 GPIO Interrupt Enable for port 2 Falling Edge (IO2IntEnF - 0x4002 80B4)

Each bit in these read-write registers enables the falling edge interrupt for the corresponding GPIO port 2 pin.

Table 118. GPIO Interrupt Enable for port 2 Falling Edge (IO2IntEnF - 0x4002 80B4) bit description

Bit	Symbol	Value	Description	Reset value
0	P2.0EF		Enable falling edge interrupt for P2.0	0
		0	Falling edge interrupt is disabled on P2.0.	
		1	Falling edge interrupt is enabled on P2.0.	
1	P2.1EF		Enable falling edge interrupt for P2.1.	0
2	P2.2EF		Enable falling edge interrupt for P2.2.	0
3	P2.3EF		Enable falling edge interrupt for P2.3.	0
4	P2.4EF		Enable falling edge interrupt for P2.4.	0
5	P2.5EF		Enable falling edge interrupt for P2.5.	0
6	P2.6EF		Enable falling edge interrupt for P2.6.	0
7	P2.7EF		Enable falling edge interrupt for P2.7.	0
8	P2.8EF		Enable falling edge interrupt for P2.8.	0
9	P2.9EF		Enable falling edge interrupt for P2.9.	0
10	P2.10EF		Enable falling edge interrupt for P2.10.	0
11	P2.11EF[1]		Enable falling edge interrupt for P2.11.	0
12	P2.12EF[1]		Enable falling edge interrupt for P2.12.	0
13	P2.13EF[1]		Enable falling edge interrupt for P2.13.	0
31:14	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.6 GPIO Interrupt Status for port 0 Rising Edge Interrupt (IO0IntStatR - 0x4002 8084)

Each bit in these read-only registers indicates the rising edge interrupt status for port 0.

Table 119. GPIO Interrupt Status for port 0 Rising Edge Interrupt (IO0IntStatR - 0x4002 8084) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0REI		Status of Rising Edge Interrupt for P0.0	0
		0	A rising edge has not been detected on P0.0.	
		1	Interrupt has been generated due to a rising edge on P0.0.	
1	P0.1REI		Status of Rising Edge Interrupt for P0.1.	0
2	P0.2REI		Status of Rising Edge Interrupt for P0.2.	0
3	P0.3REI		Status of Rising Edge Interrupt for P0.3.	0
4	P0.4REI[1]		Status of Rising Edge Interrupt for P0.4.	0
5	P0.5REI[1]		Status of Rising Edge Interrupt for P0.5.	0
6	P0.6REI		Status of Rising Edge Interrupt for P0.6.	0
7	P0.7REI		Status of Rising Edge Interrupt for P0.7.	0
8	P0.8REI		Status of Rising Edge Interrupt for P0.8.	0
9	P0.9REI		Status of Rising Edge Interrupt for P0.9.	0
10	P0.10REI		Status of Rising Edge Interrupt for P0.10.	0
11	P0.11REI		Status of Rising Edge Interrupt for P0.11.	0
14:12	-		Reserved.	NA
15	P0.15REI		Status of Rising Edge Interrupt for P0.15.	0
16	P0.16REI		Status of Rising Edge Interrupt for P0.16.	0
17	P0.17REI		Status of Rising Edge Interrupt for P0.17.	0
18	P0.18REI		Status of Rising Edge Interrupt for P0.18.	0
19	P0.19REI[1]		Status of Rising Edge Interrupt for P0.19.	0
20	P0.20REI[1]		Status of Rising Edge Interrupt for P0.20.	0
21	P0.21REI[1]		Status of Rising Edge Interrupt for P0.21.	0
22	P0.22REI		Status of Rising Edge Interrupt for P0.22.	0
23	P0.23REI[1]		Status of Rising Edge Interrupt for P0.23.	0
24	P0.24REI[1]		Status of Rising Edge Interrupt for P0.24.	0
25	P0.25REI		Status of Rising Edge Interrupt for P0.25.	0
26	P0.26REI		Status of Rising Edge Interrupt for P0.26.	0
27	P0.27REI[1]		Status of Rising Edge Interrupt for P0.27.	0
28	P0.28REI[1]		Status of Rising Edge Interrupt for P0.28.	0
29	P0.29REI		Status of Rising Edge Interrupt for P0.29.	0
30	P0.30REI		Status of Rising Edge Interrupt for P0.30.	0
31	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.7 GPIO Interrupt Status for port 2 Rising Edge Interrupt (IO2IntStatR - 0x4002 80A4)

Each bit in these read-only registers indicates the rising edge interrupt status for port 2.

Table 120. GPIO Interrupt Status for port 2 Rising Edge Interrupt (IO2IntStatR - 0x4002 80A4) bit description

Bit	Symbol	Value	Description	Reset value
0	P2.0REI		Status of Rising Edge Interrupt for P2.0	0
		0	A rising edge has not been detected on P2.0.	
		1	Interrupt has been generated due to a rising edge on P2.0.	
1	P2.1REI		Status of Rising Edge Interrupt for P2.1.	0
2	P2.2REI		Status of Rising Edge Interrupt for P2.2.	0
3	P2.3REI		Status of Rising Edge Interrupt for P2.3.	0
4	P2.4REI		Status of Rising Edge Interrupt for P2.4.	0
5	P2.5REI		Status of Rising Edge Interrupt for P2.5.	0
6	P2.6REI		Status of Rising Edge Interrupt for P2.6.	0
7	P2.7REI		Status of Rising Edge Interrupt for P2.7.	0
8	P2.8REI		Status of Rising Edge Interrupt for P2.8.	0
9	P2.9REI		Status of Rising Edge Interrupt for P2.9.	0
10	P2.10REI		Status of Rising Edge Interrupt for P2.10.	0
11	P2.11REI[1]		Status of Rising Edge Interrupt for P2.11.	0
12	P2.12REI[1]		Status of Rising Edge Interrupt for P2.12.	0
13	P2.13REI[1]		Status of Rising Edge Interrupt for P2.13.	0
31:14	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.8 GPIO Interrupt Status for port 0 Falling Edge Interrupt (IO0IntStatF - 0x4002 8088)

Each bit in these read-only registers indicates the falling edge interrupt status for port 0.

Table 121. GPIO Interrupt Status for port 0 Falling Edge Interrupt (IO0IntStatF - 0x4002 8088) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0FEI		Status of Falling Edge Interrupt for P0.0	0
		0	A falling edge has not been detected on P0.0.	
		1	Interrupt has been generated due to a falling edge on P0.0.	
1	P0.1FEI		Status of Falling Edge Interrupt for P0.1.	0
2	P0.2FEI		Status of Falling Edge Interrupt for P0.2.	0
3	P0.3FEI		Status of Falling Edge Interrupt for P0.3.	0
4	P0.4FEI[1]		Status of Falling Edge Interrupt for P0.4.	0
5	P0.5FEI[1]		Status of Falling Edge Interrupt for P0.5.	0
6	P0.6FEI		Status of Falling Edge Interrupt for P0.6.	0
7	P0.7FEI		Status of Falling Edge Interrupt for P0.7.	0

Table 121. GPIO Interrupt Status for port 0 Falling Edge Interrupt (IO0IntStatF - 0x4002 8088) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
8	P0.8FEI		Status of Falling Edge Interrupt for P0.8.	0
9	P0.9FEI		Status of Falling Edge Interrupt for P0.9.	0
10	P0.10FEI		Status of Falling Edge Interrupt for P0.10.	0
11	P0.11FEI		Status of Falling Edge Interrupt for P0.11.	0
14:12	-		Reserved.	NA
15	P0.15FEI		Status of Falling Edge Interrupt for P0.15.	0
16	P0.16FEI		Status of Falling Edge Interrupt for P0.16.	0
17	P0.17FEI		Status of Falling Edge Interrupt for P0.17.	0
18	P0.18FEI		Status of Falling Edge Interrupt for P0.18.	0
19	P0.19FEI[1]		Status of Falling Edge Interrupt for P0.19.	0
20	P0.20FEI[1]		Status of Falling Edge Interrupt for P0.20.	0
21	P0.21FEI[1]		Status of Falling Edge Interrupt for P0.21.	0
22	P0.22FEI		Status of Falling Edge Interrupt for P0.22.	0
23	P0.23FEI[1]		Status of Falling Edge Interrupt for P0.23.	0
24	P0.24FEI[1]		Status of Falling Edge Interrupt for P0.24.	0
25	P0.25FEI		Status of Falling Edge Interrupt for P0.25.	0
26	P0.26FEI		Status of Falling Edge Interrupt for P0.26.	0
27	P0.27FEI[1]		Status of Falling Edge Interrupt for P0.27.	0
28	P0.28FEI[1]		Status of Falling Edge Interrupt for P0.28.	0
29	P0.29FEI		Status of Falling Edge Interrupt for P0.29.	0
30	P0.30FEI		Status of Falling Edge Interrupt for P0.30.	0
31	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.9 GPIO Interrupt Status for port 2 Falling Edge Interrupt (IO2IntStatF - 0x4002 80A8)

Each bit in these read-only registers indicates the falling edge interrupt status for port 2.

Table 122. GPIO Interrupt Status for port 2 Falling Edge Interrupt (IO2IntStatF - 0x4002 80A8) bit description

Bit	Symbol	Value	Description	Reset value
0	P2.0FEI		Status of Falling Edge Interrupt for P2.0	0
		0	A falling edge has not been detected on P2.0.	
		1	Interrupt has been generated due to a falling edge on P2.0.	
1	P2.1FEI		Status of Falling Edge Interrupt for P2.1.	0
2	P2.2FEI		Status of Falling Edge Interrupt for P2.2.	0
3	P2.3FEI		Status of Falling Edge Interrupt for P2.3.	0
4	P2.4FEI		Status of Falling Edge Interrupt for P2.4.	0
5	P2.5FEI		Status of Falling Edge Interrupt for P2.5.	0
6	P2.6FEI		Status of Falling Edge Interrupt for P2.6.	0

Table 122. GPIO Interrupt Status for port 2 Falling Edge Interrupt (IO2IntStatF - 0x4002 80A8) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
7	P2.7FEI		Status of Falling Edge Interrupt for P2.7.	0
8	P2.8FEI		Status of Falling Edge Interrupt for P2.8.	0
9	P2.9FEI		Status of Falling Edge Interrupt for P2.9.	0
10	P2.10FEI		Status of Falling Edge Interrupt for P2.10.	0
11	P2.11FEI[1]		Status of Falling Edge Interrupt for P2.11.	0
12	P2.12FEI[1]		Status of Falling Edge Interrupt for P2.12.	0
13	P2.13FEI[1]		Status of Falling Edge Interrupt for P2.13.	0
31:14	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.10 GPIO Interrupt Clear register for port 0 (IO0IntClr - 0x4002 808C)

Writing a 1 into a bit in this write-only register clears any interrupts for the corresponding port 0 pin.

Table 123. GPIO Interrupt Clear register for port 0 (IO0IntClr - 0x4002 808C)) bit description

Bit	Symbol	Value	Description	Reset value
0	P0.0CI		Clear GPIO port Interrupts for P0.0	0
		0	Corresponding bits in IOxIntStatR and IOxIntStatF are unchanged.	
		1	Corresponding bits in IOxIntStatR and IOxStatF are cleared.	
1	P0.1CI		Clear GPIO port Interrupts for P0.1.	0
2	P0.2CI		Clear GPIO port Interrupts for P0.2.	0
3	P0.3CI		Clear GPIO port Interrupts for P0.3.	0
4	P0.4CI[1]		Clear GPIO port Interrupts for P0.4.	0
5	P0.5CI[1]		Clear GPIO port Interrupts for P0.5.	0
6	P0.6CI		Clear GPIO port Interrupts for P0.6.	0
7	P0.7CI		Clear GPIO port Interrupts for P0.7.	0
8	P0.8CI		Clear GPIO port Interrupts for P0.8.	0
9	P0.9CI		Clear GPIO port Interrupts for P0.9.	0
10	P0.10CI		Clear GPIO port Interrupts for P0.10.	0
11	P0.11CI		Clear GPIO port Interrupts for P0.11.	0
14:12	-		Reserved.	NA
15	P0.15CI		Clear GPIO port Interrupts for P0.15.	0
16	P0.16CI		Clear GPIO port Interrupts for P0.16.	0
17	P0.17CI		Clear GPIO port Interrupts for P0.17.	0
18	P0.18CI		Clear GPIO port Interrupts for P0.18.	0
19	P0.19CI[1]		Clear GPIO port Interrupts for P0.19.	0
20	P0.20CI[1]		Clear GPIO port Interrupts for P0.20.	0
21	P0.21CI[1]		Clear GPIO port Interrupts for P0.21.	0

Table 123. GPIO Interrupt Clear register for port 0 (IO0IntClr - 0x4002 808C)) bit description *...continued*

Bit	Symbol	Value	Description	Reset value
22	P0.22CI		Clear GPIO port Interrupts for P0.22.	0
23	P0.23CI[1]		Clear GPIO port Interrupts for P0.23.	0
24	P0.24CI[1]		Clear GPIO port Interrupts for P0.24.	0
25	P0.25CI		Clear GPIO port Interrupts for P0.25.	0
26	P0.26CI		Clear GPIO port Interrupts for P0.26.	0
27	P0.27CI[1]		Clear GPIO port Interrupts for P0.27.	0
28	P0.28CI[1]		Clear GPIO port Interrupts for P0.28.	0
29	P0.29CI		Clear GPIO port Interrupts for P0.29.	0
30	P0.30CI		Clear GPIO port Interrupts for P0.30.	0
31	-		Reserved.	NA

[1] Not available on 80-pin package.

9.5.6.11 GPIO Interrupt Clear register for port 2 (IO2IntClr - 0x4002 80AC)

Writing a 1 into a bit in this write-only register clears any interrupts for the corresponding port 2 pin.

Table 124. GPIO Interrupt Clear register for port 2 (IO2IntClr - 0x4002 80AC) bit description

Bit	Symbol	Value	Description	Reset value
0	P2.0CI		Clear GPIO port Interrupts for P2.0	0
		0	Corresponding bits in IOxIntStatR and IOxIntStatF are unchanged.	
		1	Corresponding bits in IOxIntStatR and IOxStatF are cleared.	
1	P2.1CI		Clear GPIO port Interrupts for P2.1.	0
2	P2.2CI		Clear GPIO port Interrupts for P2.2.	0
3	P2.3CI		Clear GPIO port Interrupts for P2.3.	0
4	P2.4CI		Clear GPIO port Interrupts for P2.4.	0
5	P2.5CI		Clear GPIO port Interrupts for P2.5.	0
6	P2.6CI		Clear GPIO port Interrupts for P2.6.	0
7	P2.7CI		Clear GPIO port Interrupts for P2.7.	0
8	P2.8CI		Clear GPIO port Interrupts for P2.8.	0
9	P2.9CI		Clear GPIO port Interrupts for P2.9.	0
10	P2.10CI		Clear GPIO port Interrupts for P2.10.	0
11	P2.11CI[1]		Clear GPIO port Interrupts for P2.11.	0
12	P2.12CI[1]		Clear GPIO port Interrupts for P2.12.	0
13	P2.13CI[1]		Clear GPIO port Interrupts for P2.13.	0
31:14	-		Reserved.	NA

[1] Not available on 80-pin package.

9.6 GPIO usage notes

9.6.1 Example: An instantaneous output of 0s and 1s on a GPIO port

Solution 1: using 32-bit (word) accessible fast GPIO registers

```
FIO0MASK = 0xFFFF00FF ;
FIO0PIN  = 0x0000A500;
```

Solution 2: using 16-bit (half-word) accessible fast GPIO registers

```
FIO0MASKL = 0x00FF;
FIO0PINL  = 0xA500;
```

Solution 3: using 8-bit (byte) accessible fast GPIO registers

```
FIO0PIN1  = 0xA5;
```

9.6.2 Writing to FIOSET/FIOCLR vs. FIOPIN

Writing to the FIOSET/FIOCLR registers allow a program to easily change a port's output pin(s) to both high and low levels at the same time. When FIOSET or FIOCLR are used, only pin/bit(s) written with 1 will be changed, while those written as 0 will remain unaffected.

Writing to the FIOPIN register enables instantaneous output of a desired value on the parallel GPIO. Data written to the FIOPIN register will affect all pins configured as outputs on that port: zeroes in the value will produce low level pin outputs and ones in the value will produce high level pin outputs.

A subset of a port's pins may be changed by using the FIOMASK register to define which pins are affected. FIOMASK is set up to contain zeroes in bits corresponding to pins that will be changed, and ones for all others. Solution 2 from Section 9.6.1 above illustrates output of 0xA5 on PORT0 pins 15 to 8 while preserving all other PORT0 output pins as they were before.

21.1 Basic configuration

The Timer 0, 1, 2, and 3 peripherals are configured using the following registers:

1. Power: In the PCONP register (Table 46), set bits PCTIM0/1/2/3.

   Remark: On reset, Timer0/1 are enabled (PCTIM0/1 = 1), and Timer2/3 are disabled (PCTIM2/3 = 0).
2. Peripheral clock: In the PCLKSEL0 register (Table 40), select PCLK_TIMER0/1; in the PCLKSEL1 register (Table 41), select PCLK_TIMER2/3.
3. Pins: Select timer pins through the PINSEL registers. Select the pin modes for the port pins with timer functions through the PINMODE registers (Section 8.5).
4. Interrupts: See register T0/1/2/3MCR (Table 430) and T0/1/2/3CCR (Table 431) for match and capture events. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register.
5. DMA: Up to two match conditions can be used to generate timed DMA requests, see Table 544.

21.2 Features

Remark: The four Timer/Counters are identical except for the peripheral base address. A minimum of two Capture inputs and two Match outputs are pinned out for all four timers, with a choice of multiple pins for each. Timer 2 brings out all four Match outputs.

- A 32-bit Timer/Counter with a programmable 32-bit Prescaler.
- Counter or Timer operation
- Up to two 32-bit capture channels per timer, that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- Four 32-bit match registers that allow:
  - Continuous operation with optional interrupt generation on match.
  - Stop timer on match with optional interrupt generation.
  - Reset timer on match with optional interrupt generation.
- Up to four external outputs corresponding to match registers, with the following capabilities:
  - Set low on match.
  - Set high on match.
  - Toggle on match.
  - Do nothing on match.

21.3 Applications

- Interval Timer for counting internal events.
- Pulse Width Demodulator via Capture inputs.
- Free running timer.

21.4 Description

The Timer/Counter is designed to count cycles of the peripheral clock (PCLK) or an externally-supplied clock, and can optionally generate interrupts or perform other actions at specified timer values, based on four match registers. It also includes four capture inputs to trap the timer value when an input signal transitions, optionally generating an interrupt.

21.5 Pin description

Table 425 gives a brief summary of each of the Timer/Counter related pins.

Table 425. Timer/Counter pin description

Pin	Type	Description
CAP0[1:0] CAP1[1:0] CAP2[1:0] CAP3[1:0]	Input	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. When more than one pin is selected for a Capture input on a single TIMER0/1 channel, the pin with the lowest Port number is used Timer/Counter block can select a capture signal as a clock source instead of the PCLK derived clock. For more details see Section 21.6.3.
MAT0[1:0] MAT1[1:0] MAT2[3:0] MAT3[1:0]	Output	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.

21.5.1 Multiple CAP and MAT pins

Software can select from multiple pins for the CAP or MAT functions in the Pin Select registers, which are described in Section 8.5. When more than one pin is selected for a MAT output, all such pins are driven identically. When more than one pin is selected for a CAP input, the pin with the lowest Port number is used. Note that match conditions may be used internally without the use of a device pin.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input can not exceed one quarter of the PCLK clock. Consequently, duration of the high/low levels on the same CAP input in this case cannot be shorter than 1/(2 x PCLK).

21.6 Register description

Each Timer/Counter contains the registers shown in Table 426 ("Reset Value" refers to the data stored in used bits only; it does not include reserved bits content). More detailed descriptions follow.

Table 426. TIMER/COUNTER0-3 register map

Generic Name	Description	Access	Reset Value[1]	TIMERn Register/ Name & Address
IR	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.	R/W	0	T0IR - 0x4000 4000 T1IR - 0x4000 8000 T2IR - 0x4009 0000 T3IR - 0x4009 4000
TCR	Timer Control Register. The TCR is used to control the Timer Counter functions. The Timer Counter can be disabled or reset through the TCR.	R/W	0	T0TCR - 0x4000 4004 T1TCR - 0x4000 8004 T2TCR - 0x4009 0004 T3TCR - 0x4009 4004
TC	Timer Counter. The 32-bit TC is incremented every PR+1 cycles of PCLK. The TC is controlled through the TCR.	R/W	0	T0TC - 0x4000 4008 T1TC - 0x4000 8008 T2TC - 0x4009 0008 T3TC - 0x4009 4008
PR	Prescale Register. When the Prescale Counter (below) is equal to this value, the next clock increments the TC and clears the PC.	R/W	0	T0PR - 0x4000 400C T1PR - 0x4000 800C T2PR - 0x4009 000C T3PR - 0x4009 400C
PC	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.	R/W	0	T0PC - 0x4000 4010 T1PC - 0x4000 8010 T2PC - 0x4009 0010 T3PC - 0x4009 4010
MCR	Match Control Register. The MCR is used to control if an interrupt is generated and if the TC is reset when a Match occurs.	R/W	0	T0MCR - 0x4000 4014 T1MCR - 0x4000 8014 T2MCR - 0x4009 0014 T3MCR - 0x4009 4014
MR0	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.	R/W	0	T0MR0 - 0x4000 4018 T1MR0 - 0x4000 8018 T2MR0 - 0x4009 0018 T3MR0 - 0x4009 4018
MR1	Match Register 1. See MR0 description.	R/W	0	T0MR1 - 0x4000 401C T1MR1 - 0x4000 801C T2MR1 - 0x4009 001C T3MR1 - 0x4009 401C
MR2	Match Register 2. See MR0 description.	R/W	0	T0MR2 - 0x4000 4020 T1MR2 - 0x4000 8020 T2MR2 - 0x4009 0020 T3MR2 - 0x4009 4020
MR3	Match Register 3. See MR0 description.	R/W	0	T0MR3 - 0x4000 4024 T1MR3 - 0x4000 8024 T2MR3 - 0x4009 0024 T3MR3 - 0x4009 4024
CCR	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.	R/W	0	T0CCR - 0x4000 4028 T1CCR - 0x4000 8028 T2CCR - 0x4009 0028 T3CCR - 0x4009 4028

Table 426. TIMER/COUNTER0-3 register map *…continued*

Generic Name	Description	Access	Reset Value[1]	TIMERn Register/ Name & Address
CR0	Capture Register 0. CR0 is loaded with the value of TC when there is an event on the CAPn.0(CAP0.0 or CAP1.0 respectively) input.	RO	0	T0CR0 - 0x4000 402C T1CR0 - 0x4000 802C T2CR0 - 0x4009 002C T3CR0 - 0x4009 402C
CR1	Capture Register 1. See CR0 description.	RO	0	T0CR1 - 0x4000 4030 T1CR1 - 0x4000 8030 T2CR1 - 0x4009 0030 T3CR1 - 0x4009 4030
EMR	External Match Register. The EMR controls the external match pins MATn.0-3 (MAT0.0-3 and MAT1.0-3 respectively).	R/W	0	T0EMR - 0x4000 403C T1EMR - 0x4000 803C T2EMR - 0x4009 003C T3EMR - 0x4009 403C
CTCR	Count Control Register. The CTCR selects between Timer and Counter mode, and in Counter mode selects the signal and edge(s) for counting.	R/W	0	T0CTCR - 0x4000 4070 T1CTCR - 0x4000 8070 T2CTCR - 0x4009 0070 T3CTCR - 0x4009 4070

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

21.6.1 Interrupt Register (T[0/1/2/3]IR - 0x4000 4000, 0x4000 8000, 0x4009 0000, 0x4009 4000)

The Interrupt Register consists of 4 bits for the match interrupts and 4 bits for the capture interrupts. If an interrupt is generated then the corresponding bit in the IR will be high. Otherwise, the bit will be low. Writing a logic one to the corresponding IR bit will reset the interrupt. Writing a zero has no effect. The act of clearing an interrupt for a timer match also clears any corresponding DMA request.

Table 427. Interrupt Register (T[0/1/2/3]IR - addresses 0x4000 4000, 0x4000 8000, 0x4009 0000, 0x4009 4000) bit description

Bit	Symbol	Description	Reset Value
0	MR0 Interrupt	Interrupt flag for match channel 0.	0
1	MR1 Interrupt	Interrupt flag for match channel 1.	0
2	MR2 Interrupt	Interrupt flag for match channel 2.	0
3	MR3 Interrupt	Interrupt flag for match channel 3.	0
4	CR0 Interrupt	Interrupt flag for capture channel 0 event.	0
5	CR1 Interrupt	Interrupt flag for capture channel 1 event.	0
31:6	-	Reserved	-

21.6.2 Timer Control Register (T[0/1/2/3]CR - 0x4000 4004, 0x4000 8004, 0x4009 0004, 0x4009 4004)

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

Table 428. Timer Control Register (TCR, TIMERn: TnTCR - addresses 0x4000 4004, 0x4000 8004, 0x4009 0004, 0x4009 4004) bit description

Bit	Symbol	Description	Reset Value
0	Counter Enable	When one, the Timer Counter and Prescale Counter are enabled for counting. When zero, the counters are disabled.	0
1	Counter Reset	When one, the Timer Counter and the Prescale Counter are synchronously reset on the next positive edge of PCLK. The counters remain reset until TCR[1] is returned to zero.	0
31:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.3 Count Control Register (T[0/1/2/3]CTCR - 0x4000 4070, 0x4000 8070, 0x4009 0070, 0x4009 4070)

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 PCLK 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 PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input can not exceed one quarter of the PCLK clock. Consequently, duration of the high/low levels on the same CAP input in this case can not be shorter than 1/(2 PCLK).

Table 429. Count Control Register (T[0/1/2/3]CTCR - addresses 0x4000 4070, 0x4000 8070, 0x4009 0070, 0x4009 4070) bit description

Bit	Symbol	Value	Description	Reset Value
1:0	Counter/ Timer Mode		This field selects which rising PCLK edges can increment the Timer's Prescale Counter (PC), or clear the PC and increment the Timer Counter (TC).	00
		00	Timer Mode: the TC is incremented when the Prescale Counter matches the Prescale Register. The Prescale Counter is incremented on every rising PCLK edge.	
		01	Counter Mode: TC is incremented on rising edges on the CAP input selected by bits 3:2.	
		10	Counter Mode: TC is incremented on falling edges on the CAP input selected by bits 3:2.	
		11	Counter Mode: TC is incremented on both edges on the CAP input selected by bits 3:2.	

Table 429. Count Control Register (T[0/1/2/3]CTCR - addresses 0x4000 4070, 0x4000 8070, 0x4009 0070, 0x4009 4070) bit description *…continued*

Bit	Symbol	Value	Description	Reset Value
3:2	Count Input Select		When bits 1:0 in this register are not 00, these bits select which CAP pin is sampled for clocking.	00
		00	CAPn.0 for TIMERn	
		01	CAPn.1 for TIMERn	
		10	Reserved	
		11	Reserved	
			Note: If Counter mode is selected for a particular CAPn input in the TnCTCR, the 3 bits for that input in the Capture Control Register (TnCCR) must be programmed as 000. However, capture and/or interrupt can be selected for the other 3 CAPn inputs in the same timer.	
31:4	-	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.4 Timer Counter registers (T0TC - T3TC, 0x4000 4008, 0x4000 8008, 0x4009 0008, 0x4009 4008)

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.

21.6.5 Prescale register (T0PR - T3PR, 0x4000 400C, 0x4000 800C, 0x4009 000C, 0x4009 400C)

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

21.6.6 Prescale Counter register (T0PC - T3PC, 0x4000 4010, 0x4000 8010, 0x4009 0010, 0x4009 4010)

The 32-bit Prescale Counter controls division of PCLK 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 PCLK. When it reaches the value stored in the Prescale register, the Timer Counter is incremented and the Prescale Counter is reset on the next PCLK. This causes the Timer Counter to increment on every PCLK when PR = 0, every 2 pclks when PR = 1, etc.

21.6.7 Match Registers (MR0 - MR3)

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.

21.6.8 Match Control Register (T[0/1/2/3]MCR - 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x4009 4014)

The Match Control Register is used to control what operations are performed when one of the Match Registers matches the Timer Counter. The function of each of the bits is shown in Table 430.

Table 430. Match Control Register (T[0/1/2/3]MCR - addresses 0x4000 4014, 0x4000 8014, 0x4009 0014, 0x4009 4014) bit description

Bit	Symbol	Value	Description	Reset Value
0	MR0I	1	Interrupt on MR0: an interrupt is generated when MR0 matches the value in the TC.	0
		0	This interrupt is disabled	
1	MR0R	1	Reset on MR0: the TC will be reset if MR0 matches it.	0
		0	Feature disabled.	
2	MR0S	1	Stop on MR0: the TC and PC will be stopped and TCR[0] will be set to 0 if MR0 matches the TC.	0
		0	Feature disabled.	
3	MR1I	1	Interrupt on MR1: an interrupt is generated when MR1 matches the value in the TC.	0
		0	This interrupt is disabled	
4	MR1R	1	Reset on MR1: the TC will be reset if MR1 matches it.	0
		0	Feature disabled.	
5	MR1S	1	Stop on MR1: the TC and PC will be stopped and TCR[0] will be set to 0 if MR1 matches the TC.	0
		0	Feature disabled.	
6	MR2I	1	Interrupt on MR2: an interrupt is generated when MR2 matches the value in the TC.	0
		0	This interrupt is disabled	
7	MR2R	1	Reset on MR2: the TC will be reset if MR2 matches it.	0
		0	Feature disabled.	
8	MR2S	1	Stop on MR2: the TC and PC will be stopped and TCR[0] will be set to 0 if MR2 matches the TC.	0
		0	Feature disabled.	
9	MR3I	1	Interrupt on MR3: an interrupt is generated when MR3 matches the value in the TC.	0
		0	This interrupt is disabled	
10	MR3R	1	Reset on MR3: the TC will be reset if MR3 matches it.	0
		0	Feature disabled.	
11	MR3S	1	Stop on MR3: the TC and PC will be stopped and TCR[0] will be set to 0 if MR3 matches the TC.	0
		0	Feature disabled.	
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 Capture Registers (CR0 - CR1)

Each Capture register is associated with a device pin and may be loaded with the Timer Counter value when a specified event occurs on that pin. 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 pin, the falling edge, or on both edges.

21.6.10 Capture Control Register (T[0/1/2/3]CCR - 0x4000 4028, 0x4000 8028, 0x4009 0028, 0x4009 4028)

The Capture Control Register is used to control whether one of the four Capture Registers is loaded with the value in the Timer Counter when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges. In the description below, "n" represents the Timer number, 0 or 1.

Note: If Counter mode is selected for a particular CAP input in the CTCR, the 3 bits for that input in this register should be programmed as 000, but capture and/or interrupt can be selected for the other 3 CAP inputs.

Table 431. Capture Control Register (T[0/1/2/3]CCR - addresses 0x4000 4028, 0x4000 8020, 0x4009 0028, 0x4009 4028) bit description

Bit	Symbol	Value	Description	Reset Value
0	CAP0RE	1	Capture on CAPn.0 rising edge: a sequence of 0 then 1 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
1	CAP0FE	1	Capture on CAPn.0 falling edge: a sequence of 1 then 0 on CAPn.0 will cause CR0 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
2	CAP0I	1	Interrupt on CAPn.0 event: a CR0 load due to a CAPn.0 event will generate an interrupt.	0
		0	This feature is disabled.	
3	CAP1RE	1	Capture on CAPn.1 rising edge: a sequence of 0 then 1 on CAPn.1 will cause CR1 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
4	CAP1FE	1	Capture on CAPn.1 falling edge: a sequence of 1 then 0 on CAPn.1 will cause CR1 to be loaded with the contents of TC.	0
		0	This feature is disabled.	
5	CAP1I	1	Interrupt on CAPn.1 event: a CR1 load due to a CAPn.1 event will generate an interrupt.	0
		0	This feature is disabled.	
31:6	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

21.6.11 External Match Register (T[0/1/2/3]EMR - 0x4000 403C, 0x4000 803C, 0x4009 003C, 0x4009 403C)

The External Match Register provides both control and status of the external match pins. In the descriptions below, “n” represents the Timer number, 0 or 1, and “m” represent a Match number, 0 through 3.

Match events for Match 0 and Match 1 in each timer can cause a DMA request, see Section 21.6.12.

Table 432. External Match Register (T[0/1/2/3]EMR - addresses 0x4000 403C, 0x4000 803C, 0x4009 003C, 0x4009 403C) bit description

Bit	Symbol	Description	Reset Value
0	EM0	External Match 0. When a match occurs between the TC and MR0, this bit can either toggle, go low, go high, or do nothing, depending on bits 5:4 of this register. This bit can be driven onto a MATn.0 pin, in a positive-logic manner (0 = low, 1 = high).	0
1	EM1	External Match 1. When a match occurs between the TC and MR1, this bit can either toggle, go low, go high, or do nothing, depending on bits 7:6 of this register. This bit can be driven onto a MATn.1 pin, in a positive-logic manner (0 = low, 1 = high).	0
2	EM2	External Match 2. When a match occurs between the TC and MR2, this bit can either toggle, go low, go high, or do nothing, depending on bits 9:8 of this register. This bit can be driven onto a MATn.2 pin, in a positive-logic manner (0 = low, 1 = high).	0
3	EM3	External Match 3. When a match occurs between the TC and MR3, this bit can either toggle, go low, go high, or do nothing, depending on bits 11:10 of this register. This bit can be driven onto a MATn.3 pin, in a positive-logic manner (0 = low, 1 = high).	0
5:4	EMC0	External Match Control 0. Determines the functionality of External Match 0. Table 433 shows the encoding of these bits.	00
7:6	EMC1	External Match Control 1. Determines the functionality of External Match 1. Table 433 shows the encoding of these bits.	00
9:8	EMC2	External Match Control 2. Determines the functionality of External Match 2. Table 433 shows the encoding of these bits.	00
11:10	EMC3	External Match Control 3. Determines the functionality of External Match 3. Table 433 shows the encoding of these bits.	00
15:12	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

Table 433. External Match Control

EMR[11:10], EMR[9:8], EMR[7:6], or EMR[5:4]	Function
00	Do Nothing.
01	Clear the corresponding External Match bit/output to 0 (MATn.m pin is LOW if pinned out).
10	Set the corresponding External Match bit/output to 1 (MATn.m pin is HIGH if pinned out).
11	Toggle the corresponding External Match bit/output.

21.6.12 DMA operation

DMA requests are generated by a match of the Timer Counter (TC) register value to either Match Register 0 (MR0) or Match Register 1 (MR1). This is not connected to the operation of the Match outputs controlled by the EMR register. Each match sets a DMA request flag, which is connected to the DMA controller. In order to have an effect, the GPDMA must be configured and the relevant timer DMA request selected as a DMA source via the DMAREQSEL register, see Section 31.5.15.

When a timer is initially set up to generate a DMA request, the request may already be asserted before a match condition occurs. An initial DMA request may be avoided by having software write a one to the interrupt flag location, as if clearing a timer interrupt. See Section 21.6.1. A DMA request will be cleared automatically when it is acted upon by

the GPDMA controller.

Remark: Because timer DMA requests are generated whenever the timer value is equal to the related Match Register value, DMA requests are always generated when the timer is running, unless the Match Register value is higher than the upper count limit of the timer. It is important not to select and enable timer DMA requests in the GPDMA block unless the timer is correctly configured to generate valid DMA requests.

21.7 Example timer operation

Figure 115 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 116 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.

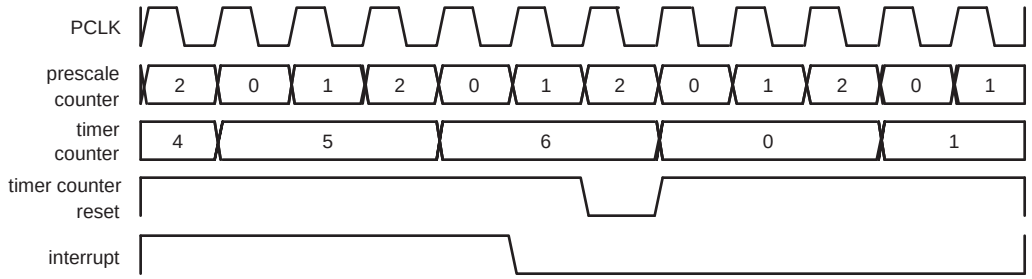

Fig 115. A timer cycle in which PR=2, MRx=6, and both interrupt and reset on match are enabled.

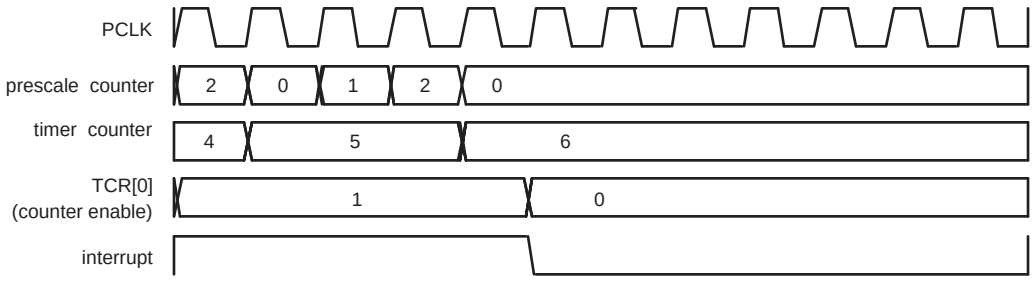

Fig 116. A timer Cycle in Which PR=2, MRx=6, and both interrupt and stop on match are enabled

21.8 Architecture

The block diagram for TIMER/COUNTER0 and TIMER/COUNTER1 is shown in Figure 117.

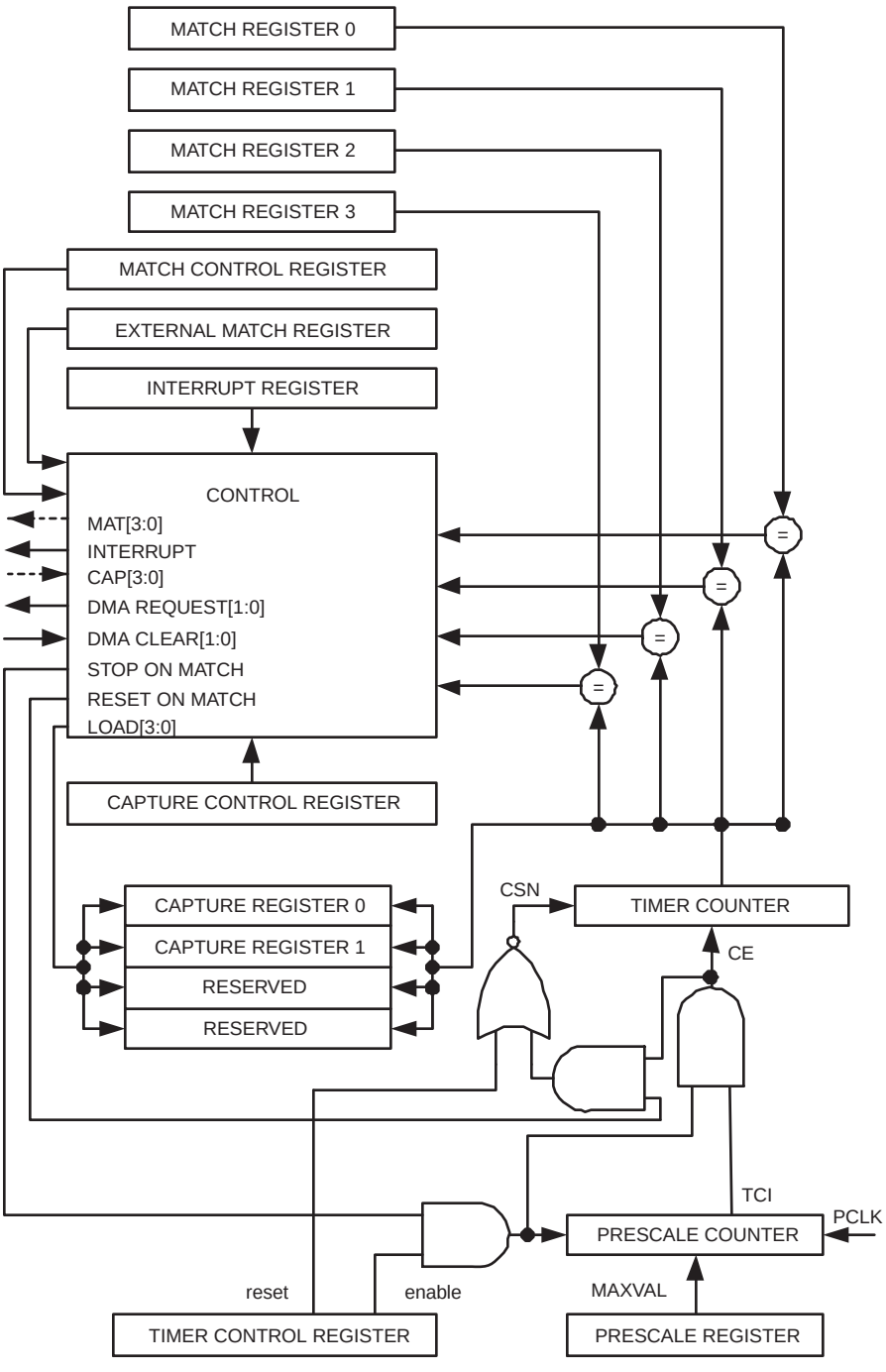

Fig 117. Timer block diagram

23.1 Basic configuration

The System Tick Timer is configured using the following registers:

1. Clock Source: Select either the internal CCLK or external STCLK (P3.26) clock as the source in the STCTRL register.
2. Pins: If STCLK (P3.26) was selected as clock source enable the STCLK pin function in the PINMODE register (Section 8.5).
3. Interrupt: The System Tick Timer Interrupt is enabled in the NVIC using the appropriate Interrupt Set Enable register.

23.2 Features

- Times intervals of 10 milliseconds
- Dedicated exception vector
- Can be clocked internally by the CPU clock or by a clock input from a pin (STCLK)

23.3 Description

The System Tick Timer is an integral part of the Cortex-M3. The System Tick Timer is intended to generate a fixed 10 millisecond interrupt for use by an operating system or other system management software.

Since the System Tick Timer is a part of the Cortex-M3, it facilitates porting of software by providing a standard timer that is available on Cortex-M3 based devices.

Refer to the Cortex-M3 User Guide appended to this manual (Section 34.4.4) for details of System Tick Timer operation.

23.4 Operation

The System Tick 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 System Tick Timer may be clocked either from the CPU clock or from the external pin STCLK. The STCLK function shares pin P3.26 with other functions, and must be selected for use as the System Tick Timer clock. In order to generate recurring interrupts at a specific interval, the STRELOAD register must be initialized with the correct value for the desired interval. A default value is provided in the STCALIB register and may be changed by software. The default value gives a 10 millisecond interrupt rate if the CPU clock is set to 100 MHz.

Remark: The maximum allowable STCLK frequency is 1/4 of CCLK.

The block diagram of the System Tick Timer is shown below in the Figure 119.

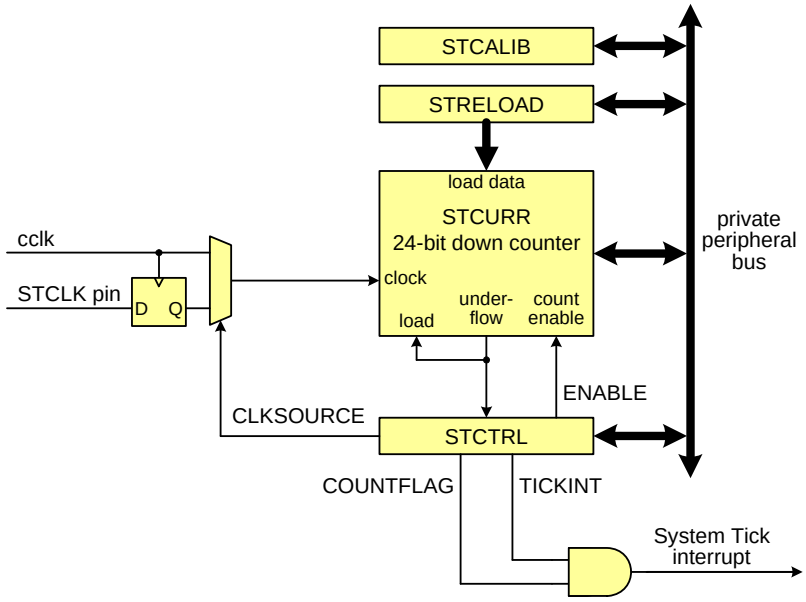

Fig 119. System Tick Timer block diagram

23.5 Register description

Table 439. System Tick Timer register map

Name	Description	Access	Reset value[1]	Address
STCTRL	System Timer Control and status register	R/W	0x4	0xE000 E010
STRELOAD	System Timer Reload value register	R/W	0	0xE000 E014
STCURR	System Timer Current value register	R/W	0	0xE000 E018
STCALIB	System Timer Calibration value register	R/W	0x000F 423F	0xE000 E01C

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

23.5.1 System Timer Control and status register (STCTRL - 0xE000 E010)

The STCTRL register contains control information for the System Tick Timer, and provides a status flag.

Table 440. System Timer Control and status register (STCTRL - 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 CPU clock is selected. When 0, the external clock pin (STCLK) is selected.	1

Table 440. System Timer Control and status register (STCTRL - 0xE000 E010) bit description *...continued*

Bit	Symbol	Description	Reset value
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	System Tick counter flag. This flag is set when the System Tick counter counts down to 0, and is cleared by reading 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

23.5.2 System Timer Reload value register (STRELOAD - 0xE000 E014)

The STRELOAD register is set to the value that will be loaded into the System Tick Timer whenever it counts down to zero. This register is loaded by software as part of timer initialization. The STCALIB register may be read and used as the value for STRELOAD if the CPU or external clock is running at the frequency intended for use with the STCALIB value.

Table 441. System Timer Reload value register (STRELOAD - 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

23.5.3 System Timer Current value register (STCURR - 0xE000 E018)

The STCURR register returns the current count from the System Tick counter when it is read by software.

Table 442. System Timer Current value register (STCURR - 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

23.5.4 System Timer Calibration value register (STCALIB - 0xE000 E01C)

The STCALIB register contains a value that is initialized by the Boot Code to a factory programmed value that is appropriate for generating an interrupt every 10 milliseconds if the System Tick Timer is clocked at a frequency of 100 MHz. This is the intended use of the System Tick Timer by ARM. It can be used to generate interrupts at other frequencies by selecting the correct reload value.

Table 443. System Timer Calibration value register (STCALIB - 0xE000 E01C) bit description

Bit	Symbol	Value	Description	Reset value
23:0	TENMS		Reload value to get a 10 millisecond System Tick underflow rate when running at 100 MHz. This value initialized at reset with a factory supplied value selected for the LPC176x/5x. The provided values of TENMS, SKEW, and NOREF are applicable only when using a CPU clock or external STCLK source of 100 MHz.	0x0F 423F
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		Indicates whether the TENMS value will generate a precise 10 millisecond time, or an approximation. This bit is initialized at reset with a factory supplied value selected for the LPC176x/5x. See the description of TENMS above. When 0, the value of TENMS is considered to be precise. When 1, the value of TENMS is not considered to be precise.	0
31	NOREF		Indicates whether an external reference clock is available. This bit is initialized at reset with a factory supplied value selected for the LPC176x/5x. See the description of TENMS above. When 0, a separate reference clock is available. When 1, a separate reference clock is not available.	0

23.6 Example timer calculations

The following examples illustrate selecting System Tick Timer values for different system configurations. All of the examples calculate an interrupt interval of 10 milliseconds, as the System Tick Timer is intended to be used.

Example 1)

This example is for the System Tick Timer running from the CPU clock (cclk), which is 100 MHz.

STCTRL = 7. This enables the timer and its interrupt, and selects cclk as the clock source.

RELOAD = (cclk / 100) - 1 = 1,000,000 - 1 = 999,999 = 0xF423F

In this case, there is no rounding error, so the result is as accurate as cclk.

Example 2)

This example is for the System Tick Timer running from the CPU clock (cclk), which is 80 MHz.

STCTRL = 7. This enables the timer and its interrupt, and selects cclk as the clock source.

RELOAD = (cclk / 100) - 1 = 800,000 - 1 = 799,999 = 0xC34FF

In this case, there is no rounding error, so the result is as accurate as cclk.

Example 3)

This example is for the CPU clock (cclk) is taken from the Internal RC Oscillator (IRC), factory trimmed to 4 MHz.

STCTRL = 7. This enables the timer and its interrupt, and selects cclk as the clock source.

RELOAD = (F_{IRC} / 100) - 1 = 40,000 - 1 = 39,999 = 0x9C3F

In this case, there is no rounding error, so the result is as accurate as the IRC.

Example 4)

This example is for the System Tick Timer running from an external clock source (the STCLK pin), which in this case happens to be 32.768 kHz.

STCTRL = 3. This enables the timer and its interrupt, and selects the STCLK pin as the clock source. STCLK must be selected as the function of the relevant pin. See Section 8.5.6.

RELOAD = (cclk / 100) - 1 = 327.6 - 1 = 327 (rounded up) = 0x0147

In this case, there is rounding error, so the interrupt rate will drift slightly relative to the input frequency.

29.1 Basic configuration

The ADC is configured using the following registers:

1. Power: In the PCONP register (Table 46), set the PCADC bit.

   Remark: On reset, the ADC is disabled. To enable the ADC, first set the PCADC bit, and then enable the ADC in the AD0CR register (bit PDN Table 532). To disable the ADC, first clear the PDN bit, and then clear the PCADC bit.
2. Clock: In the PCLKSEL0 register (Table 40), select PCLK_ADC. To scale the clock for the ADC, see bits CLKDIV in Table 532.
3. Pins: Enable ADC0 pins through PINSEL registers. Select the pin modes for the port pins with ADC0 functions through the PINMODE registers (Section 8.5).
4. Interrupts: To enable interrupts in the ADC, see Table 536. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register. Disable the ADC interrupt in the NVIC using the appropriate Interrupt Set Enable register.
5. DMA: See Section 29.6.4. For GPDMA system connections, see Table 544.

29.2 Features

- 12-bit successive approximation analog to digital converter.
- Input multiplexing among 8 pins.
- Power-down mode.
- Measurement range V_{REFN} to V_{REFP} (typically 3 V; not to exceed V_{DDA} voltage level).
- 12-bit conversion rate of 200 kHz.
- Burst conversion mode for single or multiple inputs.
- Optional conversion on transition on input pin or Timer Match signal.

29.3 Description

Basic clocking for the A/D converters is provided by the APB clock. A programmable divider is included in each converter to scale this clock to the clock (maximum 13 MHz) needed by the successive approximation process. A non-burst mode conversion requires 65 clocks and a burst mode conversion requires 64 clocks.

29.4 Pin description

Table 530 gives a brief summary of each of ADC related pins.

Table 530. ADC pin description

Pin	Type	Description
AD0.7 to AD0.0	Input	**Analog Inputs.** The ADC cell can measure the voltage on any of these input signals. Digital signals are disconnected from the ADC input pins when the ADC function is selected on that pin in the Pin Select register. **Warning:** if the ADC is used, signal levels on analog input pins must not be above the level of V_{DDA} at any time. Otherwise, A/D converter readings will be invalid. If the A/D converter is not used in an application then the pins associated with A/D inputs can be used as 5 V tolerant digital IO pins.
V_{REFP}, V_{REFN}	Reference	**Voltage References.** These pins provide a voltage reference level for the ADC and DAC. **Note: V_{REFP} should be tied to VDD(3V3) and V_{REFN} should be tied to V_{SS} if the ADC and DAC are not used.**
V_{DDA}, V_{SSA}	Power	**Analog Power and Ground.** These should typically be the same voltages as V_{DD} and V_{SS}, but should be isolated to minimize noise and error. **Note: VDDA should be tied to VDD(3V3) and VSSA should be tied to VSS if the ADC and DAC are not used.**

29.5 Register description

The A/D Converter registers are shown in Table 531.

Table 531. ADC registers

Generic Name	Description	Access	Reset value[1]	AD0 Name & Address
ADCR	A/D Control Register. The ADCR register must be written to select the operating mode before A/D conversion can occur.	R/W	1	AD0CR - 0x4003 4000
ADGDR	A/D Global Data Register. This register contains the ADC's DONE bit and the result of the most recent A/D conversion.	R/W	NA	AD0GDR - 0x4003 4004
ADINTEN	A/D Interrupt Enable Register. This register contains enable bits that allow the DONE flag of each A/D channel to be included or excluded from contributing to the generation of an A/D interrupt.	R/W	0x100	AD0INTEN - 0x4003 400C
ADDR0	A/D Channel 0 Data Register. This register contains the result of the most recent conversion completed on channel 0.	RO	NA	AD0DR0 - 0x4003 4010
ADDR1	A/D Channel 1 Data Register. This register contains the result of the most recent conversion completed on channel 1.	RO	NA	AD0DR1 - 0x4003 4014
ADDR2	A/D Channel 2 Data Register. This register contains the result of the most recent conversion completed on channel 2.	RO	NA	AD0DR2 - 0x4003 4018
ADDR3	A/D Channel 3 Data Register. This register contains the result of the most recent conversion completed on channel 3.	RO	NA	AD0DR3 - 0x4003 401C
ADDR4	A/D Channel 4 Data Register. This register contains the result of the most recent conversion completed on channel 4.	RO	NA	AD0DR4 - 0x4003 4020
ADDR5	A/D Channel 5 Data Register. This register contains the result of the most recent conversion completed on channel 5.	RO	NA	AD0DR5 - 0x4003 4024
ADDR6	A/D Channel 6 Data Register. This register contains the result of the most recent conversion completed on channel 6.	RO	NA	AD0DR6 - 0x4003 4028
ADDR7	A/D Channel 7 Data Register. This register contains the result of the most recent conversion completed on channel 7.	RO	NA	AD0DR7 - 0x4003 402C
ADSTAT	A/D Status Register. This register contains DONE and OVERRUN flags for all of the A/D channels, as well as the A/D interrupt/DMA flag.	RO	0	AD0STAT - 0x4003 4030
ADTRM	ADC trim register.	R/W	0x0000 0F00	AD0TRM - 0x4003 4034

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

29.5.1 A/D Control Register (AD0CR - 0x4003 4000)

Table 532: A/D Control Register (AD0CR - address 0x4003 4000) bit description

Bit	Symbol	Value	Description	Reset value
7:0	SEL		Selects which of the AD0.7:0 pins is (are) to be sampled and converted. For AD0, bit 0 selects Pin AD0.0, and bit 7 selects pin AD0.7. In software-controlled mode, only one of these bits should be 1. In hardware scan mode, any value containing 1 to 8 ones is allowed. All zeroes is equivalent to 0x01.	0x01
15:8	CLKDIV		The APB clock (PCLK_ADC0) is divided by (this value plus one) to produce the clock for the A/D converter, which should be less than or equal to 13 MHz. Typically, software should program the smallest value in this field that yields a clock of 13 MHz or slightly less, but in certain cases (such as a high-impedance analog source) a slower clock may be desirable.	0
16	BURST	1	The AD converter does repeated conversions at up to 200 kHz, scanning (if necessary) through the pins selected by bits set to ones in the SEL field. The first conversion after the start corresponds to the least-significant 1 in the SEL field, then higher numbered 1-bits (pins) if applicable. Repeated conversions can be terminated by clearing this bit, but the conversion that's in progress when this bit is cleared will be completed. **Remark:** START bits must be 000 when BURST = 1 or conversions will not start. If BURST is set to 1, the ADGINTEN bit in the AD0INTEN register (Table 534) must be set to 0.	0
		0	Conversions are software controlled and require 65 clocks.	
20:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
21	PDN	1	The A/D converter is operational.	0
		0	The A/D converter is in power-down mode.	
23:22	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
26:24	START		When the BURST bit is 0, these bits control whether and when an A/D conversion is started:	0
		000	No start (this value should be used when clearing PDN to 0).	
		001	Start conversion now.	
		010	Start conversion when the edge selected by bit 27 occurs on the P2.10 / EINT0 / NMI pin. Set the pin function to EINT0 in PINSEL4 register.	
		011	Start conversion when the edge selected by bit 27 occurs on the P1.27 / CLKOUT / USB_OVRCRn / CAP0.1 pin. Set the pin function to CAP0.1 in PINSEL3 register.	
		100	Start conversion when the edge selected by bit 27 occurs on MAT0.1. Note that this does not require that the MAT0.1 function appear on a device pin.	
		101	Start conversion when the edge selected by bit 27 occurs on MAT0.3. Note that it is not possible to cause the MAT0.3 function to appear on a device pin.	
		110	Start conversion when the edge selected by bit 27 occurs on MAT1.0. Note that this does not require that the MAT1.0 function appear on a device pin.	
		111	Start conversion when the edge selected by bit 27 occurs on MAT1.1. Note that this does not require that the MAT1.1 function appear on a device pin.	
27	EDGE		This bit is significant only when the START field contains 010-111. In these cases:	0
		1	Start conversion on a falling edge on the selected CAP/MAT signal.	
		0	Start conversion on a rising edge on the selected CAP/MAT signal.	
31:28	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

29.5.2 A/D Global Data Register (AD0GDR - 0x4003 4004)

The A/D Global Data Register holds the result of the most recent A/D conversion that has completed, and also includes copies of the status flags that go with that conversion.

Results of ADC conversion can be read in one of two ways. One is to use the A/D Global Data Register to read all data from the ADC. Another is to use the A/D Channel Data Registers. It is important to use one method consistently because the DONE and OVERRUN flags can otherwise get out of synch between the AD0GDR and the A/D Channel Data Registers, potentially causing erroneous interrupts or DMA activity.

Table 533: A/D Global Data Register (AD0GDR - address 0x4003 4004) 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	When DONE is 1, this field contains a binary fraction representing the voltage on the AD0[n] pin selected by the SEL field, 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
23:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
26:24	CHN	These bits contain the channel from which the RESULT bits were converted (e.g. 000 identifies channel 0, 001 channel 1...).	NA
29:27	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits. This bit is cleared by reading this register.	0
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read and when the ADCR is written. If the ADCR is written while a conversion is still in progress, this bit is set and a new conversion is started.	0

29.5.3 A/D Interrupt Enable register (AD0INTEN - 0x4003 400C)

This register allows control over which A/D channels generate an interrupt when a conversion is complete. For example, it may be desirable to use some A/D channels to monitor sensors by continuously performing conversions on them. The most recent results are read by the application program whenever they are needed. In this case, an interrupt is not desirable at the end of each conversion for some A/D channels.

Table 534: A/D Interrupt Enable register (AD0INTEN - address 0x4003 400C) bit description

Bit	Symbol	Value	Description	Reset value
0	ADINTEN0	0	Completion of a conversion on ADC channel 0 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 0 will generate an interrupt.	
1	ADINTEN1	0	Completion of a conversion on ADC channel 1 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 1 will generate an interrupt.	
2	ADINTEN2	0	Completion of a conversion on ADC channel 2 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 2 will generate an interrupt.	

Table 534: A/D Interrupt Enable register (AD0INTEN - address 0x4003 400C) bit description *…continued*

Bit	Symbol	Value	Description	Reset value
3	ADINTEN3	0	Completion of a conversion on ADC channel 3 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 3 will generate an interrupt.	
4	ADINTEN4	0	Completion of a conversion on ADC channel 4 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 4 will generate an interrupt.	
5	ADINTEN5	0	Completion of a conversion on ADC channel 5 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 5 will generate an interrupt.	
6	ADINTEN6	0	Completion of a conversion on ADC channel 6 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 6 will generate an interrupt.	
7	ADINTEN7	0	Completion of a conversion on ADC channel 7 will not generate an interrupt.	0
		1	Completion of a conversion on ADC channel 7 will generate an interrupt.	
8	ADGINTEN	0	Only the individual ADC channels enabled by ADINTEN7:0 will generate interrupts. **Remark:** This bit must be set to 0 in burst mode (BURST = 1 in the AD0CR register).	1
		1	Only the global DONE flag in ADDR is enabled to generate an interrupt.	
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

29.5.4 A/D Data Registers (AD0DR0 to AD0DR7 - 0x4003 4010 to 0x4003 402C)

The A/D Data Registers hold the result of the last conversion for each A/D channel, when an A/D conversion is complete. They also include the flags that indicate when a conversion has been completed and when a conversion overrun has occurred.

Results of ADC conversion can be read in one of two ways. One is to use the A/D Global Data Register to read all data from the ADC. Another is to use the A/D Channel Data Registers. It is important to use one method consistently because the DONE and OVERRUN flags can otherwise get out of synch between the AD0GDR and the A/D Channel Data Registers, potentially causing erroneous interrupts or DMA activity.

Table 535: A/D Data Registers (AD0DR0 to AD0DR7 - 0x4003 4010 to 0x4003 402C) 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	When DONE is 1, this field contains 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
29:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
30	OVERRUN	This bit is 1 in burst mode if the results of one or more conversions was (were) lost and overwritten before the conversion that produced the result in the RESULT bits.This bit is cleared by reading this register.	
31	DONE	This bit is set to 1 when an A/D conversion completes. It is cleared when this register is read.	NA

29.5.5 A/D Status register (ADSTAT - 0x4003 4030)

The A/D Status register allows checking the status of all A/D channels simultaneously. The DONE and OVERRUN flags appearing in the ADDRn register for each A/D channel are mirrored in ADSTAT. The interrupt flag (the logical OR of all DONE flags) is also found in ADSTAT.

Table 536: A/D Status register (AD0STAT - address 0x4003 4030) bit description

Bit	Symbol	Description	Reset value
0	DONE0	This bit mirrors the DONE status flag from the result register for A/D channel 0.	0
1	DONE1	This bit mirrors the DONE status flag from the result register for A/D channel 1.	0
2	DONE2	This bit mirrors the DONE status flag from the result register for A/D channel 2.	0
3	DONE3	This bit mirrors the DONE status flag from the result register for A/D channel 3.	0
4	DONE4	This bit mirrors the DONE status flag from the result register for A/D channel 4.	0
5	DONE5	This bit mirrors the DONE status flag from the result register for A/D channel 5.	0
6	DONE6	This bit mirrors the DONE status flag from the result register for A/D channel 6.	0
7	DONE7	This bit mirrors the DONE status flag from the result register for A/D channel 7.	0
8	OVERRUN0	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 0.	0
9	OVERRUN1	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 1.	0
10	OVERRUN2	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 2.	0
11	OVERRUN3	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 3.	0
12	OVERRUN4	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 4.	0
13	OVERRUN5	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 5.	0
14	OVERRUN6	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 6.	0
15	OVERRUN7	This bit mirrors the OVERRRUN status flag from the result register for A/D channel 7.	0
16	ADINT	This bit is the A/D interrupt flag. It is one when any of the individual A/D channel Done flags is asserted and enabled to contribute to the A/D interrupt via the ADINTEN 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

29.5.6 A/D Trim register (ADTRIM - 0x4003 4034)

This register will be set by the bootcode on start-up. It contains the trim values for the DAC and the ADC. The offset trim values for the ADC can be overwritten by the user. All 12 bits are visible when this register is read.

Table 537: A/D Trim register (ADTRM - address 0x4003 4034) bit description

Bit	Symbol	Description	Reset value
3:0	-	reserved.	NA
7:4	ADCOFFS	Offset trim bits for ADC operation. Initialized by the boot code. Can be overwritten by the user.	0
11:8	TRIM	written-to by boot code. Can **not** be overwritten by the user. These bits are locked after boot code write.	1111
31:12	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

29.6 Operation

Once an ADC conversion is started, it cannot be interrupted. A new software write to launch a new conversion or a new edge-trigger event will be ignored while the previous conversion is in progress.

29.6.1 Hardware-triggered conversion

If the BURST bit in the ADCR is 0 and the START field contains 010-111, the ADC will start a conversion when a transition occurs on a selected pin or Timer Match signal. The choices include conversion on a specified edge of any of 4 Match signals, or conversion on a specified edge of either of 2 Capture/Match pins. The pin state from the selected pad or the selected Match signal, XORed with ADCR bit 27, is used in the edge detection logic.

29.6.2 Interrupts

An interrupt request is asserted to the NVIC when the DONE bit is 1. Software can use the Interrupt Enable bit for the A/D Converter in the NVIC to control whether this assertion results in an interrupt. DONE is negated when the ADDR is read.

29.6.3 Accuracy vs. digital receiver

The ADC function must be selected via the PINSEL registers in order to get accurate voltage readings on the monitored pin. The PINMODE should also be set to the mode for which neither pull-up nor pull-down resistor is enabled. For a pin hosting an ADC input, it is not possible to have a have a digital function selected and yet get valid ADC readings. An inside circuit disconnects ADC hardware from the associated pin whenever a digital function is selected on that pin.

29.6.4 DMA control

A DMA transfer request is generated from the ADC interrupt request line. To generate a DMA transfer the same conditions must be met as the conditions for generating an interrupt (see Section 29.6.2 and Section 29.5.3).

Remark: If the DMA is used, the ADC interrupt must be disabled in the NVIC.

For DMA transfers, only burst requests are supported. The burst size can be set to one in the DMA channel control register (see Section 31.5.20). If the number of ADC channels is not equal to one of the other DMA-supported burst sizes (applicable DMA burst sizes are 1, 4, 8 - see Section 31.5.20), set the burst size to one.

The DMA transfer size determines when a DMA interrupt is generated. The transfer size can be set to the number of ADC channels being converted (see Section 31.5.20). Non-contiguous channels can be transferred by the DMA using the scatter/gather linked lists (see Section 31.5.19).

30.1 Basic configuration

The DAC is configured using the following registers:

1. Power: The DAC is always connected to V_{DDA}. Register access is determined by PINSEL and PINMODE settings (see below).
2. Clock: In the PCLKSEL0 register (Table 40), select PCLK_DAC.
3. Pins: Enable the DAC pin through the PINSEL registers. Select pin mode for port pin with DAC through the PINMODE registers (Section 8.5). This must be done before accessing any DAC registers.
4. DMA: The DAC can be connected to the GPDMA controller (see Section 30.4.2). For GPDMA connections, see Table 544.

30.2 Features

- 10-bit digital to analog converter
- Resistor string architecture
- Buffered output
- Selectable speed vs. power
- Maximum update rate of 1 MHz.

30.3 Pin description

Table 538 gives a brief summary of each of DAC related pins.

Table 538. D/A Pin Description

Pin	Type	Description
AOUT	Output	**Analog Output.** After the selected settling time after the DACR is written with a new value, the voltage on this pin (with respect to V_{SSA}) is VALUE × ((V_{REFP} - V_{REFN})/1024) + V_{REFN}. **Remark:** The DAC output is disabled when the device is in deep-sleep, power-down, or deep power-down mode.
V_{REFP}, V_{REFN}	Reference	**Voltage References.** These pins provide a voltage reference level for the ADC and DAC. **Note: V_{REFP} should be tied to VDD(3V3) and V_{REFN} should be tied to V_{SS} if the ADC and DAC are not used.**
V_{DDA}, V_{SSA}	Power	**Analog Power and Ground.** These should typically be the same voltages as V_{DD} and V_{SS}, but should be isolated to minimize noise and error. **Note: VDDA should be tied to VDD(3V3) and VSSA should be tied to VSS if the ADC and DAC are not used.**

30.4 Register description

The DAC registers are shown in Table 539. Note that the DAC does not have a control bit in the PCONP register. To enable the DAC, its output must be selected to appear on the related pin, P0.26, by configuring the PINSEL1 register. See Section 8.5.2 “Pin Function Select Register 1 (PINSEL1 - 0x4002 C004)”. the DAC must be enabled in this manner prior to accessing any DAC registers.

Table 539. DAC registers

Name	Description	Access	Reset value[1]	Address
DACR	D/A Converter Register. This register contains the digital value to be converted to analog and a power control bit.	R/W	0	0x4008 C000
DACCTRL	DAC Control register. This register controls DMA and timer operation.	R/W	0	0x4008 C004
DACCNTVAL	DAC Counter Value register. This register contains the reload value for the DAC DMA/Interrupt timer.	R/W	0	0x4008 C008

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

30.4.1 D/A Converter Register (DACR - 0x4008 C000)

This read/write register includes the digital value to be converted to analog, and a bit that trades off performance vs. power. Bits 5:0 are reserved for future, higher-resolution D/A converters.

Table 540: D/A Converter Register (DACR - address 0x4008 C000) bit description

Bit	Symbol	Value	Description	Reset Value
5:0	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA
15:6	VALUE		After the selected settling time after this field is written with a new VALUE, the voltage on the AOUT pin (with respect to V_{SSA}) is VALUE × ((V_{REFP} - V_{REFN})/1024) + V_{REFN}.	0
16	BIAS[1]	0	The settling time of the DAC is 1 μs max, and the maximum current is 700 μA. This allows a maximum update rate of 1 MHz.	0
		1	The settling time of the DAC is 2.5 μs and the maximum current is 350 μA. This allows a maximum update rate of 400 kHz.	
31:17	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

[1] The settling times noted in the description of the BIAS bit are valid for a capacitance load on the AOUT pin not exceeding 100 pF. A load impedance value greater than that value will cause settling time longer than the specified time. One or more graph(s) of load impedance vs. settling time will be included in the final data sheet.

30.4.2 D/A Converter Control register (DACCTRL - 0x4008 C004)

This read/write register enables the DMA operation and controls the DMA timer.

Table 541. D/A Control register (DACCTRL - address 0x4008 C004) bit description

Bit	Symbol	Value	Description	Reset Value
0	INT_DMA_REQ	0	This bit is cleared on any write to the DACR register.	0
		1	This bit is set by hardware when the timer times out.	
1	DBLBUF_ENA	0	DACR double-buffering is disabled.	0
		1	When this bit and the CNT_ENA bit are both set, the double-buffering feature in the DACR register will be enabled. Writes to the DACR register are written to a pre-buffer and then transferred to the DACR on the next time-out of the counter.	
2	CNT_ENA	0	Time-out counter operation is disabled.	0
		1	Time-out counter operation is enabled.	
3	DMA_ENA	0	DMA access is disabled.	0
		1	DMA Burst Request Input 7 is enabled for the DAC (see Table 544).	
31:4	-		Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.	NA

30.4.3 D/A Converter Counter Value register (DACCNTVAL - 0x4008 C008)

This read/write register contains the reload value for the Interrupt/DMA counter.

Table 542: D/A Converter register (DACR - address 0x4008 C008) bit description

Bit	Symbol	Description	Reset Value
15:0	VALUE	16-bit reload value for the DAC interrupt/DMA timer.	0

30.5 Operation

30.5.1 DMA counter

When the counter enable bit CNT_ENA in DACCTRL is set, a 16-bit counter will begin counting down, at the rate selected by PCLK_DAC (see Table 40), from the value programmed into the DACCNTVAL register. The counter is decremented Each time the counter reaches zero, the counter will be reloaded by the value of DACCNTVAL and the DMA request bit INT_DMA_REQ will be set in hardware.

Note that the contents of the DACCTRL and DACCNTVAL registers are read and write accessible, but the timer itself is not accessible for either read or write.

If the DMA_ENA bit is set in the DACCTRL register, the DAC DMA request will be routed to the GPDMA. When the DMA_ENA bit is cleared, the default state after a reset, DAC DMA requests are blocked.

30.5.2 Double buffering

Double-buffering is enabled only if both, the CNT_ENA and the DBLBUF_ENA bits are set in DACCTRL. In this case, any write to the DACR register will only load the pre-buffer, which shares its register address with the DACR register. The DACR itself will be loaded from the pre-buffer whenever the counter reaches zero and the DMA request is set. At the same time the counter is reloaded with the COUNTVAL register value.

Reading the DACR register will only return the contents of the DACR register itself, not the contents of the pre-buffer register.

If either the CNT_ENA or the DBLBUF_ENA bits are 0, any writes to the DACR address will go directly to the DACR register.

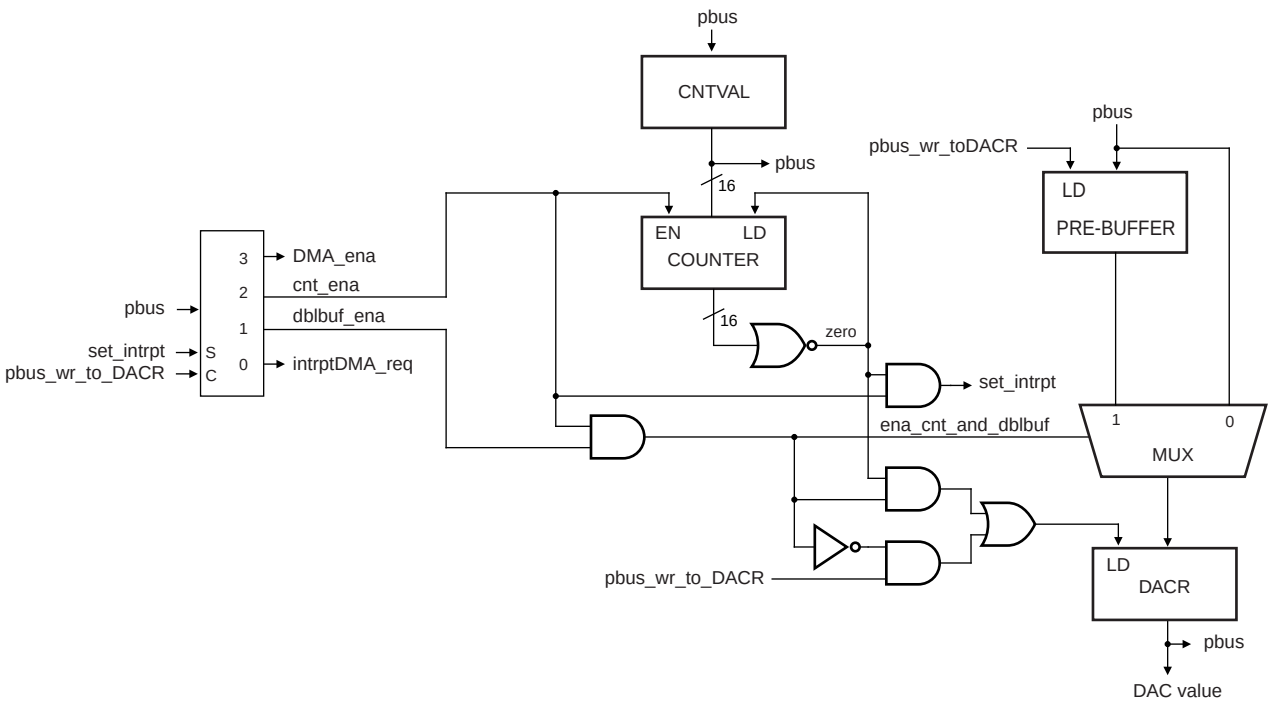

Fig 134. DAC control with DMA interrupt and timer

31.1 Basic configuration

The GPDMA is configured using the following registers:

1. Power: In the PCONP register (Table 46), set bit PCGPDMA.

   Remark: On reset, the GPDMA is disabled (PCGPDMA = 0).
2. Clock: see Table 38.
3. Interrupts: Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register.
4. Programming: see Section 31.6.

31.2 Introduction

The DMA controller allows peripheral-to memory, memory-to-peripheral, and memory-to-memory transactions. Each DMA stream provides unidirectional serial DMA transfers for a single source and destination. For example, a bi-directional port requires one stream for transmit and one for receives. The source and destination areas can each be either a memory region or a peripheral.

31.3 Features

- Eight DMA channels. Each channel can support an unidirectional transfer.
- 16 DMA request lines.
- Memory-to-memory, memory-to-peripheral, and peripheral-to-memory transfers are supported.
- GPDMA supports the SSP, I2S, UART, A/D Converter, and D/A Converter peripherals. DMA can also be triggered by a timer match condition. Memory-to-memory transfers and transfers to or from GPIO are also supported.
- Scatter or gather DMA is supported through the use of linked lists. This means that the source and destination areas do not have to occupy contiguous areas of memory.
- Hardware DMA channel priority.
- AHB slave DMA programming interface. The DMA Controller is programmed by writing to the DMA control registers over the AHB slave interface.
- One AHB bus master for transferring data. The interface transfers data when a DMA request goes active.
- 32-bit AHB master bus width.
- Incrementing or non-incrementing addressing for source and destination.
- Programmable DMA burst size. The DMA burst size can be programmed to more efficiently transfer data.
- Internal four-word FIFO per channel.
- Supports 8-bit, 16-bit, and 32-bit wide transactions.

- Big-endian and little-endian support. The DMA Controller defaults to little-endian mode on reset.
- An interrupt to the processor can be generated on a DMA completion or when a DMA error has occurred.
- Raw interrupt status. The DMA error and DMA count raw interrupt status can be read prior to masking.
- DMA can operate in Sleep mode. (Note that in Sleep mode the GPDMA cannot access the flash memory).

31.4 Functional description

This section describes the major functional blocks of the DMA Controller.

31.4.1 DMA controller functional description

The DMA Controller enables peripheral-to-memory, memory-to-peripheral, peripheral-to-peripheral, and memory-to-memory transactions. Each DMA stream provides unidirectional serial DMA transfers for a single source and destination. For example, a bidirectional port requires one stream for transmit and one for receive. The source and destination areas can each be either a memory region or a peripheral, and can be accessed through the AHB master. Figure 135 shows a block diagram of the DMA Controller.

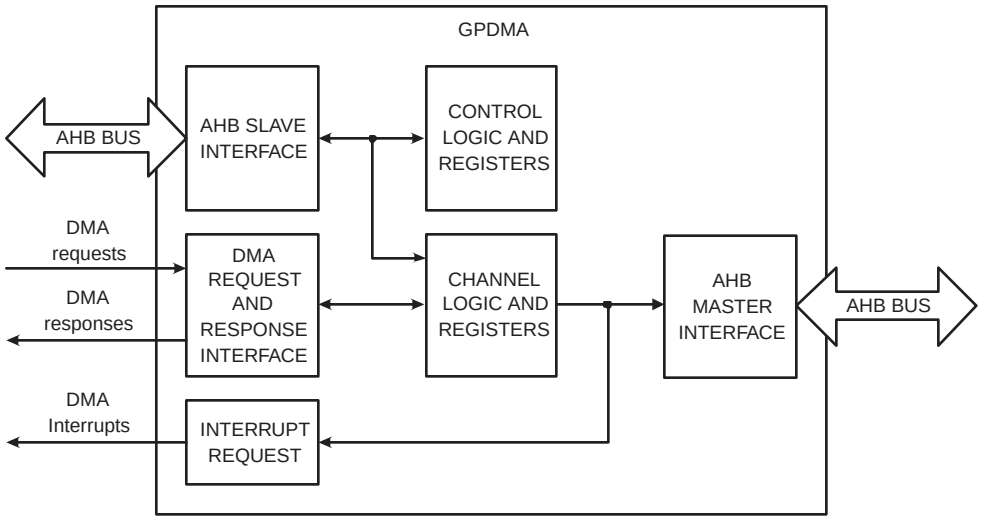

Fig 135. DMA controller block diagram

The functions of the DMA Controller are described in the following sections.

31.4.1.1 AHB slave interface

All transactions to DMA Controller registers on the AHB slave interface are 32 bits wide. 8-bit and 16-bit accesses are not supported and will result in an exception.

31.4.1.2 Control logic and register bank

The register block stores data written or to be read across the AHB interface.

31.4.1.3 DMA request and response interface

See Section 31.4.2 for information on the DMA request and response interface.

31.4.1.4 Channel logic and channel register bank

The channel logic and channel register bank contains registers and logic required for each DMA channel.

31.4.1.5 Interrupt request

The interrupt request generates the interrupt to the ARM processor.

31.4.1.6 AHB master interface

The DMA Controller contains one AHB master interface. The AHB master is capable of dealing with all types of AHB transactions, including:

- Split, retry, and error responses from slaves. If a peripheral performs a split or retry, the DMA Controller stalls and waits until the transaction can complete.
- Locked transfers for source and destination of each stream.
- Setting of protection bits for transfers on each stream.

31.4.1.6.1 Bus and transfer widths

The physical width of the AHB bus is 32 bits. Source and destination transfers can be of differing widths and can be the same width or narrower than the physical bus width. The DMA Controller packs or unpacks data as appropriate.

31.4.1.6.2 Endian behavior

The DMA Controller can cope with both little-endian and big-endian addressing.

Internally the DMA Controller treats all data as a stream of bytes instead of 16-bit or 32-bit quantities. This means that when performing mixed-endian activity, where the endianness of the source and destination are different, byte swapping of the data within the 32-bit data bus is observed.

Note: If byte swapping is not required, then use of different endianness between the source and destination addresses must be avoided. Table 543 shows endian behavior for different source and destination combinations.

Table 543. Endian behavior

Source endian	Destination endian	Source width	Destination width	Source transfer no/byte lane	Source data	Destination transfer no/byte lane	Destination data
Little	Little	8	8	1/[7:0]	21	1/[7:0]	21212121
				2/[15:8]	43	2/[15:8]	43434343
				3/[23:16]	65	3/[23:16]	65656565
				4/[31:24]	87	4/[31:24]	87878787
Little	Little	8	16	1/[7:0]	21	1/[15:0]	43214321
				2/[15:8]	43	2/[31:16]	87658765
				3/[23:16]	65		
				4/[31:24]	87		
Little	Little	8	32	1/[7:0]	21	1/[31:0]	87654321
				2/[15:8]	43		
				3/[23:16]	65		
				4/[31:24]	87		
Little	Little	16	8	1/[7:0]	21	1/[7:0]	21212121
				1/[15:8]	43	2/[15:8]	43434343
				2/[23:16]	65	3/[23:16]	65656565
				2/[31:24]	87	4/[31:24]	87878787
Little	Little	16	16	1/[7:0]	21	1/[15:0]	43214321
				1/[15:8]	43	2/[31:16]	87658765
				2/[23:16]	65		
				2/[31:24]	87		
Little	Little	16	32	1/[7:0]	21	1/[31:0]	87654321
				1/[15:8]	43		
				2/[23:16]	65		
				2/[31:24]	87		
Little	Little	32	8	1/[7:0]	21	1/[7:0]	21212121
				1/[15:8]	43	2/[15:8]	43434343
				1/[23:16]	65	3/[23:16]	65656565
				1/[31:24]	87	4/[31:24]	87878787
Little	Little	32	16	1/[7:0]	21	1/[15:0]	43214321
				1/[15:8]	43	2/[31:16]	87658765
				1/[23:16]	65		
				1/[31:24]	87		
Little	Little	32	32	1/[7:0]	21	1/[31:0]	87654321
				1/[15:8]	43		
				1/[23:16]	65		
				1/[31:24]	87		
Big	Big	8	8	1/[31:24]	12	1/[31:24]	12121212
				2/[23:16]	34	2/[23:16]	34343434
				3/[15:8]	56	3/[15:8]	56565656
				4/[7:0]	78	4/[7:0]	78787878

Table 543. Endian behavior *…continued*

Source endian	Destination endian	Source width	Destination width	Source transfer no/byte lane	Source data	Destination transfer no/byte lane	Destination data
Big	Big	8	16	1/[31:24]	12	1/[15:0]	12341234
				2/[23:16]	34	2/[31:16]	56785678
				3/[15:8]	56		
				4/[7:0]	78		
Big	Big	8	32	1/[31:24]	12	1/[31:0]	12345678
				2/[23:16]	34		
				3/[15:8]	56		
				4/[7:0]	78		
Big	Big	16	8	1/[31:24]	12	1/[31:24]	12121212
				1/[23:16]	34	2/[23:16]	34343434
				2/[15:8]	56	3/[15:8]	56565656
				2/[7:0]	78	4/[7:0]	78787878
Big	Big	16	16	1/[31:24]	12	1/[15:0]	12341234
				1/[23:16]	34	2/[31:16]	56785678
				2/[15:8]	56		
				2/[7:0]	78		
Big	Big	16	32	1/[31:24]	12	1/[31:0]	12345678
				1/[23:16]	34		
				2/[15:8]	56		
				2/[7:0]	78		
Big	Big	32	8	1/[31:24]	12	1/[31:24]	12121212
				1/[23:16]	34	2/[23:16]	34343434
				1/[15:8]	56	3/[15:8]	56565656
				1/[7:0]	78	4/[7:0]	78787878
Big	Big	32	16	1/[31:24]	12	1/[15:0]	12341234
				1/[23:16]	34	2/[31:16]	56785678
				1/[15:8]	56		
				1/[7:0]	78		
Big	Big	32	32	1/[31:24]	12	1/[31:0]	12345678
				1/[23:16]	34		
				1/[15:8]	56		
				1/[7:0]	78		

31.4.1.6.3 Error conditions

An error during a DMA transfer is flagged directly by the peripheral by asserting an Error response on the AHB bus during the transfer. The DMA Controller automatically disables the DMA stream after the current transfer has completed, and can optionally generate an error interrupt to the CPU. This error interrupt can be masked.

31.4.1.7 Channel hardware

Each stream is supported by a dedicated hardware channel, including source and destination controllers, as well as a FIFO. This enables better latency than a DMA controller with only a single hardware channel shared between several DMA streams and simplifies the control logic.

31.4.1.8 DMA request priority

DMA channel priority is fixed. DMA channel 0 has the highest priority and DMA channel 7 has the lowest priority.

If the DMA Controller is transferring data for the lower priority channel and then the higher priority channel goes active, it completes the number of transfers delegated to the master interface by the lower priority channel before switching over to transfer data for the higher priority channel. Transfers delegated to the master interface are staged in the DMA channel FIFO, so the amount of data that needs to transfer could be as large as a 4 words.

It is recommended that memory-to-memory transactions use the lowest priority channel.

31.4.1.9 Interrupt generation

A combined interrupt output is generated as an OR function of the individual interrupt requests of the DMA Controller and is connected to the interrupt controller.

31.4.2 DMA system connections

31.4.2.1 DMA request signals

The DMA request signals are used by peripherals to request a data transfer. The DMA request signals indicate whether a single or burst transfer of data is required. The DMA available request signals are:

DMACBREQ[15:0] — Burst request signals. These cause a programmed burst number of data to be transferred.

DMACSREQ[15:0] — Single transfer request signals. These cause a single data to be transferred. The DMA controller transfers a single transfer to or from the peripheral.

DMACLBREQ[15:0] — Last burst request signals.

DMACLSREQ[15:0] — Last single transfer request signals.

Note that peripherals on this device do not support “last” request types, and many do not support both single and burst request types. See Section 31.4.2.3.

31.4.2.2 DMA response signals

The DMA response signals indicate whether the transfer initiated by the DMA request signal has completed. The response signals can also be used to indicate whether a complete packet has been transferred. The DMA response signals from the DMA controller are:

DMACCLR[15:0] — DMA clear or acknowledge signals. The DMACCLR signal is used by the DMA controller to acknowledge a DMA request from the peripheral.

DMACTC[15:0] — DMA terminal count signals. The DMACTC signal can be used by the DMA controller to indicate to the peripheral that the DMA transfer is complete.

31.4.2.3 DMA request connections

The connection of the GPDMA to the supported peripheral devices depends on the DMA functions implemented in those peripherals. Table 544 shows the DMA Request numbers used by the supported peripherals. UART and timer DMA requests on channels 8 through 15 are chosen via the DMAREQSEL register, see Section 31.5.15.

Table 544. DMA Connections

Peripheral Function	DMA Single Request Input (DMACSREQ)	DMA Burst Request Input (DMACBREQ)	DMA Request Signal
SSP0 Tx	0	0	Dedicated DMA requests
SSP0 Rx	1	1	Dedicated DMA requests
SSP1 Tx	2	2	Dedicated DMA requests
SSP1 Rx	3	3	Dedicated DMA requests
ADC	4	4	ADC interrupt request [1]
I^2S channel 0	-	5	Dedicated DMA request
I^2S channel 1	-	6	Dedicated DMA request
DAC	-	7	Dedicated DMA request
UART0 Tx / MAT0.0	-	8	Dedicated DMA requests
UART0 Rx / MAT0.1	-	9	Dedicated DMA requests
UART1 Tx / MAT1.0	-	10	Dedicated DMA requests
UART1 Rx / MAT1.1	-	11	Dedicated DMA requests
UART2 Tx / MAT2.0	-	12	Dedicated DMA requests
UART2 Rx / MAT2.1	-	13	Dedicated DMA requests
UART3 Tx / MAT3.0	-	14	Dedicated DMA requests
UART3 Rx / MAT3.1	-	15	Dedicated DMA requests

[1] Generates an interrupt and/or DMA request depending on software setup.

31.5 Register description

The DMA Controller supports 8 channels. Each channel has registers specific to the operation of that channel. Other registers controls aspects of how source peripherals relate to the DMA Controller. There are also global DMA control and status registers.

The DMA Controller registers are shown in Table 545.

Table 545. GPDMA register map

Name	Description	Access	Reset state	Address
General registers				
DMACIntStat	DMA Interrupt Status Register	RO	0	0x5000 4000
DMACIntTCStat	DMA Interrupt Terminal Count Request Status Register	RO	0	0x5000 4004
DMACIntTCClear	DMA Interrupt Terminal Count Request Clear Register	WO	-	0x5000 4008
DMACIntErrStat	DMA Interrupt Error Status Register	RO	0	0x5000 400C
DMACIntErrClr	DMA Interrupt Error Clear Register	WO	-	0x5000 4010
DMACRawIntTCStat	DMA Raw Interrupt Terminal Count Status Register	RO	0	0x5000 4014
DMACRawIntErrStat	DMA Raw Error Interrupt Status Register	RO	0	0x5000 4018
DMACEnbldChns	DMA Enabled Channel Register	RO	0	0x5000 401C
DMACSoftBReq	DMA Software Burst Request Register	R/W	0	0x5000 4020
DMACSoftSReq	DMA Software Single Request Register	R/W	0	0x5000 4024
DMACSoftLBReq	DMA Software Last Burst Request Register	R/W	0	0x5000 4028
DMACSoftLSReq	DMA Software Last Single Request Register	R/W	0	0x5000 402C
DMACConfig	DMA Configuration Register	R/W	0	0x5000 4030
DMACSync	DMA Synchronization Register	R/W	0	0x5000 4034
DMAREQSEL	Selects between UART and timer DMA requests on channels 8 through 15	R/W	0	0x400F C1C4
Channel 0 registers				
DMACC0SrcAddr	DMA Channel 0 Source Address Register	R/W	0	0x5000 4100
DMACC0DestAddr	DMA Channel 0 Destination Address Register	R/W	0	0x5000 4104
DMACC0LLI	DMA Channel 0 Linked List Item Register	R/W	0	0x5000 4108
DMACC0Control	DMA Channel 0 Control Register	R/W	0	0x5000 410C
DMACC0Config	DMA Channel 0 Configuration Register	R/W	0 [1]	0x5000 4110
Channel 1 registers				
DMACC1SrcAddr	DMA Channel 1 Source Address Register	R/W	0	0x5000 4120
DMACC1DestAddr	DMA Channel 1 Destination Address Register	R/W	0	0x5000 4124
DMACC1LLI	DMA Channel 1 Linked List Item Register	R/W	0	0x5000 4128
DMACC1Control	DMA Channel 1 Control Register	R/W	0	0x5000 412C
DMACC1Config	DMA Channel 1 Configuration Register	R/W	0 [1]	0x5000 4130
Channel 2 registers				
DMACC2SrcAddr	DMA Channel 2 Source Address Register	R/W	0	0x5000 4140
DMACC2DestAddr	DMA Channel 2 Destination Address Register	R/W	0	0x5000 4144
DMACC2LLI	DMA Channel 2 Linked List Item Register	R/W	0	0x5000 4148
DMACC2Control	DMA Channel 2 Control Register	R/W	0	0x5000 414C

Table 545. GPDMA register map

Name	Description	Access	Reset state	Address
DMACC2Config	DMA Channel 2 Configuration Register	R/W	0 [1]	0x5000 4150
Channel 3 registers				
DMACC3SrcAddr	DMA Channel 3 Source Address Register	R/W	0	0x5000 4160
DMACC3DestAddr	DMA Channel 3 Destination Address Register	R/W	0	0x5000 4164
DMACC3LLI	DMA Channel 3 Linked List Item Register	R/W	0	0x5000 4168
DMACC3Control	DMA Channel 3 Control Register	R/W	0	0x5000 416C
DMACC3Config	DMA Channel 3 Configuration Register	R/W	0 [1]	0x5000 4170
Channel 4 registers				
DMACC4SrcAddr	DMA Channel 4 Source Address Register	R/W	0	0x5000 4180
DMACC4DestAddr	DMA Channel 4 Destination Address Register	R/W	0	0x5000 4184
DMACC4LLI	DMA Channel 4 Linked List Item Register	R/W	0	0x5000 4188
DMACC4Control	DMA Channel 4 Control Register	R/W	0	0x5000 418C
DMACC4Config	DMA Channel 4 Configuration Register	R/W	0 [1]	0x5000 4190
Channel 5 registers				
DMACC5SrcAddr	DMA Channel 5 Source Address Register	R/W	0	0x5000 41A0
DMACC5DestAddr	DMA Channel 5 Destination Address Register	R/W	0	0x5000 41A4
DMACC5LLI	DMA Channel 5 Linked List Item Register	R/W	0	0x5000 41A8
DMACC5Control	DMA Channel 5 Control Register	R/W	0	0x5000 41AC
DMACC5Config	DMA Channel 5 Configuration Register	R/W	0 [1]	0x5000 41B0
Channel 6 registers				
DMACC6SrcAddr	DMA Channel 6 Source Address Register	R/W	0	0x5000 41C0
DMACC6DestAddr	DMA Channel 6 Destination Address Register	R/W	0	0x5000 41C4
DMACC6LLI	DMA Channel 6 Linked List Item Register	R/W	0	0x5000 41C8
DMACC6Control	DMA Channel 6 Control Register	R/W	0	0x5000 41CC
DMACC6Config	DMA Channel 6 Configuration Register	R/W	0 [1]	0x5000 41D0
Channel 7 registers				
DMACC7SrcAddr	DMA Channel 7 Source Address Register	R/W	0	0x5000 41E0
DMACC7DestAddr	DMA Channel 7 Destination Address Register	R/W	0	0x5000 41E4
DMACC7LLI	DMA Channel 7 Linked List Item Register	R/W	0	0x5000 41E8
DMACC7Control	DMA Channel 7 Control Register	R/W	0	0x5000 41EC
DMACC7Config	DMA Channel 7 Configuration Register	R/W	0 [1]	0x5000 41F0

[1] Bit 17 of this register is a read-only status flag.

31.5.1 DMA Interrupt Status register (DMACIntStat - 0x5000 4000)

The DMACIntStat Register is read-only and shows the status of the interrupts after masking. A 1 bit indicates that a specific DMA channel interrupt request is active. The request can be generated from either the error or terminal count interrupt requests. Table 546 shows the bit assignments of the DMACIntStat Register.

Table 546. DMA Interrupt Status register (DMACIntStat - 0x5000 4000)

Bit	Name	Function
7:0	IntStat	Status of DMA channel interrupts after masking. Each bit represents one channel: 0 - the corresponding channel has no active interrupt request. 1 - the corresponding channel does have an active interrupt request.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.2 DMA Interrupt Terminal Count Request Status register (DMACIntTCStat - 0x5000 4004)

The DMACIntTCStat Register is read-only and indicates the status of the terminal count after masking. Table 547 shows the bit assignments of the DMACIntTCStat Register.

Table 547. DMA Interrupt Terminal Count Request Status register (DMACIntTCStat - 0x5000 4004)

Bit	Name	Function
7:0	IntTCStat	Terminal count interrupt request status for DMA channels. Each bit represents one channel: 0 - the corresponding channel has no active terminal count interrupt request. 1 - the corresponding channel does have an active terminal count interrupt request.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.3 DMA Interrupt Terminal Count Request Clear register (DMACIntTCClear - 0x5000 4008)

The DMACIntTCClear Register is write-only and clears one or more terminal count interrupt requests. When writing to this register, each data bit that contains a 1 causes the corresponding bit in the status register (DMACIntTCStat) to be cleared. Data bits that are 0 have no effect. Table 548 shows the bit assignments of the DMACIntTCClear Register.

Table 548. DMA Interrupt Terminal Count Request Clear register (DMACIntTCClear - 0x5000 4008)

Bit	Name	Function
7:0	IntTCClear	Allows clearing the Terminal count interrupt request (IntTCStat) for DMA channels. Each bit represents one channel: 0 - writing 0 has no effect. 1 - clears the corresponding channel terminal count interrupt.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.4 DMA Interrupt Error Status register (DMACIntErrStat - 0x5000 400C)

The DMACIntErrStat Register is read-only and indicates the status of the error request after masking. Table 549 shows the bit assignments of the DMACIntErrStat Register.

Table 549. DMA Interrupt Error Status register (DMACIntErrStat - 0x5000 400C)

Bit	Name	Function
7:0	IntErrStat	Interrupt error status for DMA channels. Each bit represents one channel: 0 - the corresponding channel has no active error interrupt request. 1 - the corresponding channel does have an active error interrupt request.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.5 DMA Interrupt Error Clear register (DMACIntErrClr - 0x5000 4010)

The DMACIntErrClr Register is write-only and clears the error interrupt requests. When writing to this register, each data bit that is 1 causes the corresponding bit in the status register to be cleared. Data bits that are 0 have no effect on the corresponding bit in the register. Table 550 shows the bit assignments of the DMACIntErrClr Register.

Table 550. DMA Interrupt Error Clear register (DMACIntErrClr - 0x5000 4010)

Bit	Name	Function
7:0	IntErrClr	Writing a 1 clears the error interrupt request (IntErrStat) for DMA channels. Each bit represents one channel: 0 - writing 0 has no effect. 1 - clears the corresponding channel error interrupt.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.6 DMA Raw Interrupt Terminal Count Status register (DMACRawIntTCStat - 0x5000 4014)

The DMACRawIntTCStat Register is read-only and indicates which DMA channel is requesting a transfer complete (terminal count interrupt) prior to masking. (Note: the DMACIntTCStat Register contains the same information after masking.) A 1 bit indicates that the terminal count interrupt request is active prior to masking. Table 551 shows the bit assignments of the DMACRawIntTCStat Register.

Table 551. DMA Raw Interrupt Terminal Count Status register (DMACRawIntTCStat - 0x5000 4014)

Bit	Name	Function
7:0	RawIntTCStat	Status of the terminal count interrupt for DMA channels prior to masking. Each bit represents one channel: 0 - the corresponding channel has no active terminal count interrupt request. 1 - the corresponding channel does have an active terminal count interrupt request.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.7 DMA Raw Error Interrupt Status register (DMACRawIntErrStat - 0x5000 4018)

The DMACRawIntErrStat Register is read-only and indicates which DMA channel is requesting an error interrupt prior to masking. (Note: the DMACIntErrStat Register contains the same information after masking.) A 1 bit indicates that the error interrupt request is active prior to masking. Table 552 shows the bit assignments of register of the DMACRawIntErrStat Register.

Table 552. DMA Raw Error Interrupt Status register (DMACRawIntErrStat - 0x5000 4018)

Bit	Name	Function
7:0	RawIntErrStat	Status of the error interrupt for DMA channels prior to masking. Each bit represents one channel: 0 - the corresponding channel has no active error interrupt request. 1 - the corresponding channel does have an active error interrupt request.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.8 DMA Enabled Channel register (DMACEnbldChns - 0x5000 401C)

The DMACEnbldChns Register is read-only and indicates which DMA channels are enabled, as indicated by the Enable bit in the DMACCxConfig Register. A 1 bit indicates that a DMA channel is enabled. A bit is cleared on completion of the DMA transfer. Table 553 shows the bit assignments of the DMACEnbldChns Register.

Table 553. DMA Enabled Channel register (DMACEnbldChns - 0x5000 401C)

Bit	Name	Function
7:0	EnabledChannels	Enable status for DMA channels. Each bit represents one channel: 0 - DMA channel is disabled. 1 - DMA channel is enabled.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.9 DMA Software Burst Request register (DMACSoftBReq - 0x5000 4020)

The DMACSoftBReq Register is read/write and enables DMA burst requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Reading the register indicates which sources are requesting DMA burst transfers. A request can be generated from either a peripheral or the software request register. Each bit is cleared when the related transaction has completed. Table 554 shows the bit assignments of the DMACSoftBReq Register.

Table 554. DMA Software Burst Request register (DMACSoftBReq - 0x5000 4020)

Bit	Name	Function
15:0	SoftBReq	Software burst request flags for each of 16 possible sources. Each bit represents one DMA request line or peripheral function (refer to Table 544 for peripheral hardware connections to the DMA controller): 0 - writing 0 has no effect. 1 - writing 1 generates a DMA burst request for the corresponding request line.
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

Note: It is recommended that software and hardware peripheral requests are not used at the same time.

31.5.10 DMA Software Single Request register (DMACSoftSReq - 0x5000 4024)

The DMACSoftSReq Register is read/write and enables DMA single transfer requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Reading the register indicates which sources are requesting single DMA transfers. A request can be generated from either a peripheral or the software request register. Table 555 shows the bit assignments of the DMACSoftSReq Register.

Table 555. DMA Software Single Request register (DMACSoftSReq - 0x5000 4024)

Bit	Name	Function
15:0	SoftSReq	Software single transfer request flags for each of 16 possible sources. Each bit represents one DMA request line or peripheral function: 0 - writing 0 has no effect. 1 - writing 1 generates a DMA single transfer request for the corresponding request line.
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.11 DMA Software Last Burst Request register (DMACSoftLBReq - 0x5000 4028)

The DMACSoftLBReq Register is read/write and enables DMA last burst requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Reading the register indicates which sources are requesting last burst DMA transfers. A request can be generated from either a peripheral or the software request register. Table 556 shows the bit assignments of the DMACSoftLBReq Register.

Table 556. DMA Software Last Burst Request register (DMACSoftLBReq - 0x5000 4028)

Bit	Name	Function
15:0	SoftLBReq	Software last burst request flags for each of 16 possible sources. Each bit represents one DMA request line or peripheral function: 0 - writing 0 has no effect. 1 - writing 1 generates a DMA last burst request for the corresponding request line.
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.12 DMA Software Last Single Request register (DMACSoftLSReq - 0x5000 402C)

The DMACSoftLSReq Register is read/write and enables DMA last single requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Reading the register indicates which sources are requesting last single DMA transfers. A request can be generated from either a peripheral or the software request register. Table 557 shows the bit assignments of the DMACSoftLSReq Register.

Table 557. DMA Software Last Single Request register (DMACSoftLSReq - 0x5000 402C)

Bit	Name	Function
15:0	SoftLSReq	Software last single transfer request flags for each of 16 possible sources. Each bit represents one DMA request line or peripheral function: 0 - writing 0 has no effect. 1 - writing 1 generates a DMA last single transfer request for the corresponding request line.
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.13 DMA Configuration register (DMACConfig - 0x5000 4030)

The DMACConfig Register is read/write and configures the operation of the DMA Controller. The endianness of the AHB master interface can be altered by writing to the M bit of this register. The AHB master interface is set to little-endian mode on reset. Table 558 shows the bit assignments of the DMACConfig Register.

Table 558. DMA Configuration register (DMACConfig - 0x5000 4030)

Bit	Name	Function
0	E	DMA Controller enable: 0 = disabled (default). Disabling the DMA Controller reduces power consumption. 1 = enabled.
1	M	AHB Master endianness configuration: 0 = little-endian mode (default). 1 = big-endian mode.
31:2	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.14 DMA Synchronization register (DMACSync - 0x5000 4034)

The DMACSync Register is read/write and enables or disables synchronization logic for the DMA request signals. The DMA request signals consist of the DMACBREQ[15:0], DMACSREQ[15:0], DMACLBREQ[15:0], and DMACLSREQ[15:0]. A bit set to 0 enables the synchronization logic for a particular group of DMA requests. A bit set to 1 disables the synchronization logic for a particular group of DMA requests. This register is reset to 0, enabling synchronization logic by default. Table 559 shows the bit assignments of the DMACSync Register.

Table 559. DMA Synchronization register (DMACSync - 0x5000 4034)

Bit	Name	Function
15:0	DMACSync	Controls the synchronization logic for DMA request signals. Each bit represents one set of DMA request lines as described in the preceding text: 0 - synchronization logic for the corresponding DMA request signals are enabled. 1 - synchronization logic for the corresponding request line signals are disabled.
31:16	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.15 DMA Request Select register (DMAReqSel - 0x400F C1C4)

DMAReqSel is a read/write register that allows selecting between UART or Timer DMA requests for DMA inputs 8 through 15. Table 560 shows the bit assignments of the DMAReqSel Register.

Table 560. DMA Request Select register (DMAReqSel - 0x400F C1C4)

Bit	Name	Function
0	DMASEL08	Selects the DMA request for GPDMA input 8: 0 - UART0 TX is selected. 1 - Timer 0 match 0 is selected.
1	DMASEL09	Selects the DMA request for GPDMA input 9: 0 - UART0 RX is selected. 1 - Timer 0 match 1 is selected.
2	DMASEL10	Selects the DMA request for GPDMA input 10: 0 - UART1 TX is selected. 1 - Timer 1match 0 is selected.
3	DMASEL11	Selects the DMA request for GPDMA input 11: 0 - UART1 RX is selected. 1 - Timer 1match 1 is selected.
4	DMASEL12	Selects the DMA request for GPDMA input 12: 0 - UART2 TX is selected. 1 - Timer 2 match 0 is selected.
5	DMASEL13	Selects the DMA request for GPDMA input 13: 0 - UART2 RX is selected. 1 - Timer 2 match 1 is selected.
6	DMASEL14	Selects the DMA request for GPDMA input 14: 0 - UART3 TX is selected. 1 - Timer 3 match 0 is selected.
7	DMASEL15	Selects the DMA request for GPDMA input 15: 0 - UART3 RX is selected. 1 - Timer 3 match 1 is selected.
31:8	-	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.16 DMA Channel registers

The channel registers are used to program the eight DMA channels. These registers consist of:

- Eight DMACCxSrcAddr Registers.
- Eight DMACCxDestAddr Registers.
- Eight DMACCxLLI Registers.
- Eight DMACCxControl Registers.
- Eight DMACCxConfig Registers.

When performing scatter/gather DMA, the first four of these are automatically updated.

31.5.17 DMA Channel Source Address registers (DMACCxSrcAddr - 0x5000 41x0)

The eight read/write DMACCxSrcAddr Registers (DMACC0SrcAddr to DMACC7SrcAddr) contain the current source address (byte-aligned) of the data to be transferred. Each register is programmed directly by software before the appropriate channel is enabled. When the DMA channel is enabled this register is updated:

- As the source address is incremented.
- By following the linked list when a complete packet of data has been transferred.

Reading the register when the channel is active does not provide useful information. This is because by the time software has processed the value read, the address may have progressed. It is intended to be read-only when the channel has stopped, in which case it shows the source address of the last item read.

Note: The source and destination addresses must be aligned to the source and destination widths.

Table 561 shows the bit assignments of the DMACCxSrcAddr Registers.

Table 561. DMA Channel Source Address registers (DMACCxSrcAddr - 0x5000 41x0)

Bit	Name	Function
31:0	SrcAddr	DMA source address. Reading this register will return the current source address.

31.5.18 DMA Channel Destination Address registers (DMACCxDestAddr - 0x5000 41x4)

The eight read/write DMACCxDestAddr Registers (DMACC0DestAddr to DMACC7DestAddr) contain the current destination address (byte-aligned) of the data to be transferred. Each register is programmed directly by software before the channel is enabled. When the DMA channel is enabled the register is updated as the destination address is incremented and by following the linked list when a complete packet of data has been transferred. Reading the register when the channel is active does not provide useful information. This is because by the time that software has processed the value read, the address may have progressed. It is intended to be read-only when a channel has stopped, in which case it shows the destination address of the last item read. Table 562 shows the bit assignments of the DMACCxDestAddr Register.

Table 562. DMA Channel Destination Address registers (DMACCxDestAddr - 0x5000 41x4)

Bit	Name	Function
31:0	DestAddr	DMA Destination address. Reading this register will return the current destination address.

31.5.19 DMA Channel Linked List Item registers (DMACCxLLI - 0x5000 41x8)

The eight read/write DMACCxLLI Registers (DMACC0LLI to DMACC7LLI) contain a word-aligned address of the next Linked List Item (LLI). If the LLI is 0, then the current LLI is the last in the chain, and the DMA channel is disabled when all DMA transfers associated with it are completed. Programming this register when the DMA channel is enabled may have unpredictable side effects. Table 563 shows the bit assignments of the DMACCxLLI Register.

Table 563. DMA Channel Linked List Item registers (DMACCxLLI - 0x5000 41x8)

Bit	Name	Function
1:0	-	Reserved, and must be written as 0.
31:2	LLI	Linked list item. Bits [31:2] of the address for the next LLI. Address bits [1:0] are 0.

31.5.20 DMA channel control registers (DMACCxControl - 0x5000 41xC)

The eight read/write DMACCxControl Registers (DMACC0Control to DMACC7Control) contain DMA channel control information such as the transfer size, burst size, and transfer width. Each register is programmed directly by software before the DMA channel is enabled. When the channel is enabled the register is updated by following the linked list when a complete packet of data has been transferred. Reading the register while the channel is active does not give useful information. This is because by the time software has processed the value read, the channel may have advanced. It is intended to be read-only when a channel has stopped. Table 564 shows the bit assignments of the DMACCxControl Register.

31.5.20.1 Protection and access information

AHB access information is provided to the source and/or destination peripherals when a transfer occurs, although on the LPC176x/5x this has no effect. The transfer information is provided by programming the DMA channel (the Prot bits of the DMACCxControl Register, and the Lock bit of the DMACCxConfig Register). These bits are programmed by software, and can be used by peripherals. Three bits of information are provided, and are used as shown in Table 564.

Table 564. DMA channel control registers (DMACCxControl - 0x5000 41xC)

Bit	Name	Function
11:0	TransferSize	Transfer size. This field sets the size of the transfer. The transfer size value must be set before the channel is enabled. Transfer size is updated as data transfers are completed. A read from this field indicates the number of transfers completed on the destination bus. Reading the register when the channel is active does not give useful information because by the time that the software has processed the value read, the channel might have progressed. It is intended to be used only when a channel is enabled and then disabled.
14:12	SBSize	Source burst size. Indicates the number of transfers that make up a source burst. This value must be set to the burst size of the source peripheral, or if the source is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMACBREQ signal goes active in the source peripheral. 000 - 1 001 - 4 010 - 8 011 - 16 100 - 32 101 - 64 110 - 128 111 - 256
17:15	DBSize	Destination burst size. Indicates the number of transfers that make up a destination burst transfer request. This value must be set to the burst size of the destination peripheral or, if the destination is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMACBREQ signal goes active in the destination peripheral. 000 - 1 001 - 4 010 - 8 011 - 16 100 - 32 101 - 64 110 - 128 111 - 256
20:18	SWidth	Source transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data as required. 000 - Byte (8-bit) 001 - Halfword (16-bit) 010 - Word (32-bit) 011 to 111 - Reserved
23:21	DWidth	Destination transfer width. Transfers wider than the AHB master bus width are not supported. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data as required. 000 - Byte (8-bit) 001 - Halfword (16-bit) 010 - Word (32-bit) 011 to 111 - Reserved
25:24	-	Reserved, and must be written as 0.

Table 564. DMA channel control registers (DMACCxControl - 0x5000 41xC) *…continued*

Bit	Name	Function
26	SI	Source increment: 0 - the source address is not incremented after each transfer. 1 - the source address is incremented after each transfer.
27	DI	Destination increment: 0 - the destination address is not incremented after each transfer. 1 - the destination address is incremented after each transfer.
28	Prot1	This is provided to the peripheral during a DMA bus access and indicates that the access is in user mode or privileged mode. This information is not used in the LPC176x/5x. 0 - access is in user mode. 1 - access is in privileged mode.
29	Prot2	This is provided to the peripheral during a DMA bus access and indicates to the peripheral that the access is bufferable or not bufferable. This information is not used in the LPC176x/5x. 0 - access is not bufferable. 1 - access is bufferable.
30	Prot3	This is provided to the peripheral during a DMA bus access and indicates to the peripheral that the access is cacheable or not cacheable. This information is not used in the LPC176x/5x. 0 - access is not cacheable. 1 - access is cacheable.
31	I	Terminal count interrupt enable bit. 0 - the terminal count interrupt is disabled. 1 - the terminal count interrupt is enabled.

31.5.21 DMA Channel Configuration registers (DMACCxConfig - 0x5000 41x0)

The eight DMACCxConfig Registers (DMACC0Config to DMACC7Config) are read/write with the exception of bit[17] which is read-only. These registers configure each DMA channel. The registers are not updated when a new LLI is requested. Table 565 shows the bit assignments of the DMACCxConfig Register.

Table 565. DMA Channel Configuration registers (DMACCxConfig - 0x5000 41x0)

Bit	Name	Function
0	E	Channel enable. Reading this bit indicates whether a channel is currently enabled or disabled: 0 = channel disabled. 1 = channel enabled. The Channel Enable bit status can also be found by reading the DMACEnbldChns Register. A channel is enabled by setting this bit. A channel can be disabled by clearing the Enable bit. This causes the current AHB transfer (if one is in progress) to complete and the channel is then disabled. Any data in the FIFO of the relevant channel is lost. Restarting the channel by setting the Channel Enable bit has unpredictable effects, the channel must be fully re-initialized. The channel is also disabled, and Channel Enable bit cleared, when the last LLI is reached, the DMA transfer is completed, or if a channel error is encountered. If a channel must be disabled without losing data in the FIFO, the Halt bit must be set so that further DMA requests are ignored. The Active bit must then be polled until it reaches 0, indicating that there is no data left in the FIFO. Finally, the Channel Enable bit can be cleared.
5:1	SrcPeripheral	Source peripheral. This value selects the DMA source request peripheral. This field is ignored if the source of the transfer is from memory. See Table 544 for peripheral identification.
10:6	DestPeripheral	Destination peripheral. This value selects the DMA destination request peripheral. This field is ignored if the destination of the transfer is to memory. See Table 544 for peripheral identification.
13:11	TransferType	This value indicates the type of transfer. The transfer type can be memory-to-memory, memory-to-peripheral, peripheral-to-memory, or peripheral-to-peripheral. Refer to Table 566 for the encoding of this field.
14	IE	Interrupt error mask. When cleared, this bit masks out the error interrupt of the relevant channel.
15	ITC	Terminal count interrupt mask. When cleared, this bit masks out the terminal count interrupt of the relevant channel.
16	L	Lock. When set, this bit enables locked transfers. This information is not used in the LPC176x/5x.
17	A	Active: 0 = there is no data in the FIFO of the channel. 1 = the channel FIFO has data. This value can be used with the Halt and Channel Enable bits to cleanly disable a DMA channel. This is a read-only bit.
18	H	Halt: 0 = enable DMA requests. 1 = ignore further source DMA requests. The contents of the channel FIFO are drained. This value can be used with the Active and Channel Enable bits to cleanly disable a DMA channel.
31:19	Reserved	Reserved, user software should not write ones to reserved bits. The value read from a reserved bit is not defined.

31.5.21.1 Lock control

The lock control may set the lock bit by writing a 1 to bit 16 of the DMACCxConfig Register. When a burst occurs, the AHB arbiter will not de-grant the master during the burst until the lock is de-asserted. The DMA Controller can be locked for a a single burst such as a long source fetch burst or a long destination drain burst. The DMA Controller does not usually assert the lock continuously for a source fetch burst followed by a destination drain burst.

There are situations when the DMA Controller asserts the lock for source transfers followed by destination transfers. This is possible when internal conditions in the DMA Controller permit it to perform a source fetch followed by a destination drain back-to-back.

31.5.21.2 Transfer type

Table 566 lists the bit values of the transfer type bits identified in Table 565.

Table 566. Transfer type bits

Bit value	Transfer type	Controller
000	Memory to memory	DMA
001	Memory to peripheral	DMA
010	Peripheral to memory	DMA
011	Source peripheral to destination peripheral	DMA
100 to 111	Reserved, do not use these combinations	-

31.6 Using the DMA controller

31.6.1 Programming the DMA controller

All accesses to the DMA Controller internal register must be word (32-bit) reads and writes.

31.6.1.1 Enabling the DMA controller

To enable the DMA controller set the Enable bit in the DMACConfig register.

31.6.1.2 Disabling the DMA controller

To disable the DMA controller:

- Read the DMACEnbldChns register and ensure that all the DMA channels have been disabled. If any channels are active, see Disabling a DMA channel.
- Disable the DMA controller by writing 0 to the DMA Enable bit in the DMACConfig register.

31.6.1.3 Enabling a DMA channel

To enable the DMA channel set the channel enable bit in the relevant DMA channel configuration register. Note that the channel must be fully initialized before it is enabled.

31.6.1.4 Disabling a DMA channel

A DMA channel can be disabled in three ways:

- By writing directly to the channel enable bit. Any outstanding data in the FIFO's is lost if this method is used.
- By using the active and halt bits in conjunction with the channel enable bit.
- By waiting until the transfer completes. This automatically clears the channel.

Disabling a DMA channel and losing data in the FIFO

Clear the relevant channel enable bit in the relevant channel configuration register. The current AHB transfer (if one is in progress) completes and the channel is disabled. Any data in the FIFO is lost.

Disabling the DMA channel without losing data in the FIFO

- Set the halt bit in the relevant channel configuration register. This causes any future DMA request to be ignored.
- Poll the active bit in the relevant channel configuration register until it reaches 0. This bit indicates whether there is any data in the channel that has to be transferred.
- Clear the channel enable bit in the relevant channel configuration register

31.6.1.5 Setting up a new DMA transfer

To set up a new DMA transfer:

If the channel is not set aside for the DMA transaction:

1. Read the DMACEnbldChns controller register and find out which channels are inactive.
2. Choose an inactive channel that has the required priority.
3. Program the DMA controller

31.6.1.6 Halting a DMA channel

Set the halt bit in the relevant DMA channel configuration register. The current source request is serviced. Any further source DMA request is ignored until the halt bit is cleared.

31.6.1.7 Programming a DMA channel

1. Choose a free DMA channel with the priority needed. DMA channel 0 has the highest priority and DMA channel 7 the lowest priority.
2. Clear any pending interrupts on the channel to be used by writing to the DMACIntTCClear and DMACIntErrClear register. The previous channel operation might have left interrupt active.
3. Write the source address into the DMACCxSrcAddr register.
4. Write the destination address into the DMACCxDestAddr register.
5. Write the address of the next LLI into the DMACCxLLI register. If the transfer comprises of a single packet of data then 0 must be written into this register.
6. Write the control information into the DMACCxControl register.
7. Write the channel configuration information into the DMACCxConfig register. If the enable bit is set then the DMA channel is automatically enabled.

31.6.2 Flow control

The device that controls the length of the packet is known as the flow controller. On the LPC176x/5x, the flow controller is always the DMA Controller, and the packet length is programmed by software before the DMA channel is enabled.

When the DMA transfer is completed:

1. The DMA Controller issues an acknowledge to the peripheral in order to indicate that the transfer has finished.
2. A TC interrupt is generated, if enabled.
3. The DMA Controller moves on to the next LLI.

The following sections describe the DMA Controller data flow sequences for the four allowed transfer types:

- Memory-to-peripheral.
- Peripheral-to-memory.
- Memory-to-memory.
- Peripheral-to-peripheral.

Table 567 indicates the request signals used for each type of transfer.

Table 567. DMA request signal usage

Transfer direction	Request generator	Flow controller
Memory-to-peripheral	Peripheral	DMA Controller
Peripheral-to-memory	Peripheral	DMA Controller
Memory-to-memory	DMA Controller	DMA Controller
Source peripheral to destination peripheral	Source peripheral and destination peripheral	DMA Controller

31.6.2.1 Peripheral-to-memory or memory-to-peripheral DMA flow

For a peripheral-to-memory or memory-to-peripheral DMA flow, the following sequence occurs:

1. Program and enable the DMA channel.
2. Wait for a DMA request.
3. The DMA Controller starts transferring data when:
   - The DMA request goes active.
   - The DMA stream has the highest pending priority.
   - The DMA Controller is the bus master of the AHB bus.
4. If an error occurs while transferring the data, an error interrupt is generated and disables the DMA stream, and the flow sequence ends.
5. Decrement the transfer count.
6. If the transfer has completed (indicated by the transfer count reaching 0):
   - The DMA Controller responds with a DMA acknowledge.
   - The terminal count interrupt is generated (this interrupt can be masked).
   - If the DMACCxLLI Register is not 0, then reload the DMACCxSrcAddr, DMACCxDestAddr, DMACCxLLI, and DMACCxControl registers and go to back to step 2. However, if DMACCxLLI is 0, the DMA stream is disabled and the flow sequence ends.

31.6.2.2 Peripheral-to-peripheral DMA flow

For a peripheral-to-peripheral DMA flow, the following sequence occurs:

1. Program and enable the DMA channel.
2. Wait for a source DMA request.
3. The DMA Controller starts transferring data when:
   - The DMA request goes active.
   - The DMA stream has the highest pending priority.
   - The DMA Controller is the bus master of the AHB bus.
4. If an error occurs while transferring the data an error interrupt is generated, the DMA stream is disabled, and the flow sequence ends.
5. Decrement the transfer count.
6. If the transfer has completed (indicated by the transfer count reaching 0):
   - The DMA Controller responds with a DMA acknowledge to the source peripheral.
   - Further source DMA requests are ignored.

7. When the destination DMA request goes active and there is data in the DMA Controller FIFO, transfer data into the destination peripheral.
8. If an error occurs while transferring the data, an error interrupt is generated, the DMA stream is disabled, and the flow sequence ends.
9. If the transfer has completed it is indicated by the transfer count reaching 0. The following happens:
   - The DMA Controller responds with a DMA acknowledge to the destination peripheral.
   - The terminal count interrupt is generated (this interrupt can be masked).
   - If the DMACCxLLI Register is not 0, then reload the DMACCxSrcAddr, DMACCxDestAddr, DMACCxLLI, and DMACCxControl Registers and go to back to step 2. However, if DMACCxLLI is 0, the DMA stream is disabled and the flow sequence ends.

31.6.2.3 Memory-to-memory DMA flow

For a memory-to-memory DMA flow the following sequence occurs:

1. Program and enable the DMA channel.
2. Transfer data whenever the DMA channel has the highest pending priority and the DMA Controller gains mastership of the AHB bus.
3. If an error occurs while transferring the data, generate an error interrupt and disable the DMA stream.
4. Decrement the transfer count.
5. If the count has reached zero:
   - Generate a terminal count interrupt (the interrupt can be masked).
   - If the DMACCxLLI Register is not 0, then reload the DMACCxSrcAddr, DMACCxDestAddr, DMACCxLLI, and DMACCxControl Registers and go to back to step 2. However, if DMACCxLLI is 0, the DMA stream is disabled and the flow sequence ends.

Note: Memory-to-memory transfers should be programmed with a low channel priority, otherwise other DMA channels cannot access the bus until the memory-to-memory transfer has finished, or other AHB masters cannot perform any transaction.

31.6.3 Interrupt requests

Interrupt requests can be generated when an AHB error is encountered or at the end of a transfer (terminal count), after all the data corresponding to the current LLI has been transferred to the destination. The interrupts can be masked by programming bits in the relevant DMACCxControl and DMACCxConfig Channel Registers. The interrupt requests from all DMA channels can be found in the DMACRawIntTCStat and DMACRawIntErrStat registers. The masked versions of the DMA interrupt data is contained in the DMACIntTCStat and DMACIntErrStat registers. The DMACIntStat register then combines the DMACIntTCStat and DMACIntErrStat requests into a single register to enable the source of an interrupt to be found quickly. Writing to the DMACIntTCClear or the DMACIntErrClr Registers with a bit set to 1 enables selective clearing of interrupts.

31.6.3.1 Hardware interrupt sequence flow

When a DMA interrupt request occurs, the Interrupt Service Routine needs to:

1. Read the DMACIntTCStat Register to determine whether the interrupt was generated due to the end of the transfer (terminal count). A 1 bit indicates that the transfer completed. If more than one request is active, it is recommended that the highest priority channels be checked first.
2. Read the DMACIntErrStat Register to determine whether the interrupt was generated due to an error occurring. A 1 bit indicates that an error occurred.
3. Service the interrupt request.
4. For a terminal count interrupt, write a 1 to the relevant bit of the DMACIntTCClr Register. For an error interrupt write a 1 to the relevant bit of the DMACIntErrClr Register to clear the interrupt request.

31.6.4 Address generation

Address generation can be either incrementing or non-incrementing (address wrapping is not supported).

Some devices, especially memories, disallow burst accesses across certain address boundaries. The DMA controller assumes that this is the case with any source or destination area, which is configured for incrementing addressing. This boundary is assumed to be aligned with the specified burst size. For example, if the channel is set for 16-transfer burst to a 32-bit wide device then the boundary is 64-bytes aligned (that is address bits [5:0] equal 0). If a DMA burst is to cross one of these boundaries, then, instead of a burst, that transfer is split into separate AHB transactions.

31.6.4.1 Word-aligned transfers across a boundary

The channel is configured for 16-transfer bursts, each transfer 32-bits wide, to a destination for which address incrementing is enabled. The start address for the current burst is 0x0C000024, the next boundary (calculated from the burst size and transfer width) is 0x0C000040.

The transfer will be split into two AHB transactions:

- a 7-transfer burst starting at address 0x0C000024
- a 9-transfer burst starting at address 0x0C000040.

31.6.5 Scatter/gather

Scatter/gather is supported through the use of linked lists. This means that the source and destination areas do not have to occupy contiguous areas in memory. Where scatter/gather is not required, the DMACCxLLI Register must be set to 0.

The source and destination data areas are defined by a series of linked lists. Each Linked List Item (LLI) controls the transfer of one block of data, and then optionally loads another LLI to continue the DMA operation, or stops the DMA stream. The first LLI is programmed into the DMA Controller.

The data to be transferred described by an LLI (referred to as the packet of data) usually requires one or more DMA bursts (to each of the source and destination).

31.6.5.1 Linked list items

A Linked List Item (LLI) consists of four words. These words are organized in the following order:

1. DMACCxSrcAddr.
2. DMACCxDestAddr.
3. DMACCxLLI.
4. DMACCxControl.

Note: The DMACCxConfig DMA channel Configuration Register is not part of the linked list item.

31.6.5.1.1 Programming the DMA controller for scatter/gather DMA

To program the DMA Controller for scatter/gather DMA:

1. Write the LLIs for the complete DMA transfer to memory. Each linked list item contains four words:
   - Source address.
   - Destination address.
   - Pointer to next LLI.
   - Control word.

   The last LLI has its linked list word pointer set to 0.
2. Choose a free DMA channel with the priority required. DMA channel 0 has the highest priority and DMA channel 7 the lowest priority.
3. Write the first linked list item, previously written to memory, to the relevant channel in the DMA Controller.
4. Write the channel configuration information to the channel Configuration Register and set the Channel Enable bit. The DMA Controller then transfers the first and then subsequent packets of data as each linked list item is loaded.
5. An interrupt can be generated at the end of each LLI depending on the Terminal Count bit in the DMACCxControl Register. If this bit is set an interrupt is generated at the end of the relevant LLI. The interrupt request must then be serviced and the relevant bit in the DMACIntTCClear Register must be set to clear the interrupt.

31.6.5.1.2 Example of scatter/gather DMA

See Figure 136 for an example of an LLI. A section of memory is to be transferred to a peripheral. The addresses of each LLI entry are given, in hexadecimal, at the left-hand side of the figure. In this example, the LLIs describing the transfer are to be stored contiguously from address 0x2002 0000, but they could be located anywhere. The right side of the figure shows the memory containing the data to be transferred.

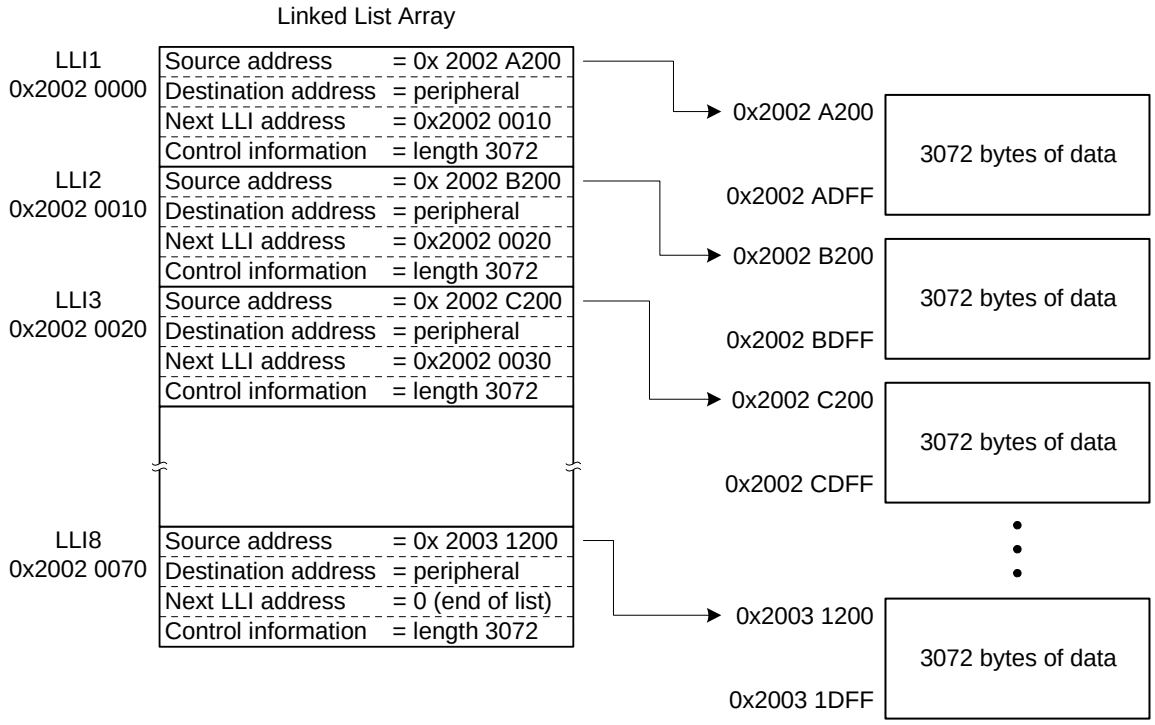

Fig 136. LLI example

The first LLI, stored at 0x2002 0000, defines the first block of data to be transferred, which is the data stored from address 0x2002 A200 to 0x2002 ADFF:

- Source start address 0x2002 A200.
- Destination address set to the destination peripheral address.
- Transfer width, word (32-bit).
- Transfer size, 3072 bytes (0XC00).
- Source and destination burst sizes, 16 transfers.
- Next LLI address, 0x2002 0010.

The second LLI, stored at 0x2002 0010, describes the next block of data to be transferred:

- Source start address 0x2002 B200.
- Destination address set to the destination peripheral address.
- Transfer width, word (32-bit).
- Transfer size, 3072 bytes (0xC00).
- Source and destination burst sizes, 16 transfers.
- Next LLI address, 0x2002 0020.

A chain of descriptors is built up, each one pointing to the next in the series. To initialize the DMA stream, the first LLI, 0x2002 0000, is programmed into the DMA Controller. When the first packet of data has been transferred the next LLI is automatically loaded.

The final LLI is stored at 0x2002 0070 and contains:

- Source start address 0x2003 1200.
- Destination address set to the destination peripheral address.
- Transfer width, word (32-bit).
- Transfer size, 3072 bytes (0xC00).
- Source and destination burst sizes, 16 transfers.
- Next LLI address, 0x0.

Because the next LLI address is set to zero, this is the last descriptor, and the DMA channel is disabled after transferring the last item of data. The channel is probably set to generate an interrupt at this point to indicate to the ARM processor that the channel can be reprogrammed.